

# Fast Estimation of Pedestrian Movement

Ying-Che Kuo,<sup>\*</sup> Cheng-Tao Tsai, and Chih-Hao Chang

Department of Electrical Engineering, National Chin-Yi University of Technology,  
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 41170, Taiwan (ROC)

(Received October 17, 2016; accepted December 26, 2016)

**Keywords:** pedestrian detection, Fisher classifier, Lucas–Kanade optical flow

In this study, a single camera has been used to capture images of the road in front of a moving vehicle. Image processing algorithms identify the location of pedestrians in the images, calculate the direction and rate of movement of each one, and issue safety warnings. The pedestrian motion vector detection and warning system presented here has three components: The first serves to identify, locate, and mark the positions of pedestrians in images using the Fisher classifier. The follow-up image processing is confined to the labeled images, and this significantly reduces the image postprocessing load. The second component involves the calculation of pedestrian motion vectors using the Lucas–Kanade optical flow method. Finally, the vehicle’s future zone of movement is established, and judgment is made as to whether any pedestrians will be in this zone or not. This is determined from the pedestrian movement vectors and a warning can be triggered to alert the driver in time for any necessary avoidance action to be taken. This can improve road safety and reduce the number of accidents involving pedestrians, which result from driver fatigue, negligence or careless driving, and the number of such accidents can be reduced.

## 1. Introduction

According to a World Health Organization (WHO) report in 2013, more than 270000 pedestrians lose their lives on the world’s roads each year, accounting for 22% of a total of 1.24 million road accident deaths. WHO has called on governments and automobile companies to take concrete action to improve pedestrian safety. In recent years, many advanced driver assistance systems (ADAS) have been developed for vehicles to improve both driving and safety. Safety features include collision avoidance technologies that alert the driver to potential problems, or take action to avoid collisions by implementing safeguards or even taking control of the vehicle. Some features of ADAS, including a pedestrian protection system,<sup>(1–3)</sup> a lane departure warning system,<sup>(4,5)</sup> a blind spot monitor,<sup>(6)</sup> automatic parking, and driver drowsiness detection,<sup>(7,8)</sup> have become the focus of accident avoidance. In these studies, the acquisition of information about the traffic environment is obtained by a computerized network of cameras. In some cases, sensing elements such as radar, ultrasound, and infrared have also been used. For example, Mercedes-Benz has developed the “PRE-SAFE with Pedestrian Recognition System”, which uses radar and stereoscopic cameras to determine the direction in which pedestrians are moving to avoid collisions. Volvo has proposed a “Pedestrian Detection with Full Auto Brake (PDFAB)” for 2020, which will enhance driving safety and achieve a “zero accident” vision. The PDFAB system will

---

<sup>\*</sup>Corresponding author: e-mail: kuoyc@ncut.edu.tw  
<http://dx.doi.org/10.18494/SAM.2017.1487>

use computer-vision algorithms to identify pedestrians 80 cm in height or more, who are within 50 m of the front of the vehicle and within a 45° fan-shaped area. Auxiliary radar will measure the distance and detect the direction of the pedestrian's movement. Mazda announced their i-ACTIVSENSE security system in November 2012. This can automatically detect obstacles in front of the vehicle (called Forward Obstruction Warning, FOW), and uses warning lights and audio signals to alert the driver to ensure early reaction. In November 2013, Ford announced a brand new obstacle avoidance system (OAS). Should a vehicle in the front, or to the side, suddenly brake or change direction as a reaction to an emergency, the driver will be warned. If the driver does not take immediate action, the OAS will actively intervene by braking or steering the car to avoid the obstacles and prevent a collision.

Pedestrians and obstacles are the most important factors in traffic accidents. Lin *et al.*<sup>(9)</sup> proposed a monocular-vision-based obstacle detection algorithm for parking assistance applications in an advanced safety vehicle that uses a rear view camera. The corner features of rear obstacles are first estimated using features that presented in accelerated segment test (FAST) corner detection method.<sup>(10)</sup> An inverse perspective mapping (IPM) image is then used to determine whether all the detected features are obstacle candidates. This system is only usable in typical urban parking lots.

Langer and Jochem<sup>(11)</sup> described an integrated millimeter wave radar and vision sensor system for autonomous on-road navigation. It has a range of approximately 200 m and uses a linear array of receivers and wave front reconstruction to compute the range and bearings of objects within the field of view. It is integrated with a vision-based lane-keeping system to accurately detect and classify obstacles with respect to the danger they pose to the vehicle and to execute any required avoidance maneuvers. The combination of radar sensors and visual detection allows road obstacles to be more accurately identified, but also results in higher cost.

Keller and Dang<sup>(12)</sup> presented a novel active pedestrian safety system that combines sensing, situation analysis, decision making, and vehicle control. The sensing component is based on stereo vision, and it fuses two complementary approaches for added robustness: (1) motion-based object detection and (2) pedestrian recognition. The highlight of the system is its ability to decide whether it will perform automatic braking or evasive steering, and reliably execute this maneuver at relatively high vehicle speed (up to 50 km/h). In this study, a stereoscopic camera is used to obtain 6D-Vision information, which is a combination of 3D information with additional information about the movement of the object(s). The 3D position is accompanied by a series of 3D positions of the object, which allows tracking. At the same time, the Kalman Filter, which combines space and time messages, can provide an estimate of the 3D motion path. The disadvantages of this system are the high cost of the stereoscopic visual sensors and the time-consuming calculations.

Alonso *et al.*<sup>(13)</sup> have proposed a safety assistant system for dynamically monitoring overtaking from the rear of a vehicle. Rear road status information from side mirror cameras mounted outside the vehicle is acquired, and calculations are made of the dynamic driving status using the Lucas–Kanade optical flow method. In addition, the author has proposed a template filtering and compensation perspective formula method, which can quickly obtain vehicle position. Although it is not as accurate as inverse perspective transformation, it has the advantage of fast execution.

In these previous studies, the use of image processing technology to identify objects (including pedestrians) is consistent, and a combination with radar is often used to accurately calculate object distance. However, the use of multiple cameras to build stereoscopic or panoramic images incurs substantial equipment cost and also increases the amount of visual algorithm computation. In our

study, a single camera has been used to capture road images and there was no need for enormous amounts of stereoscopic image data processing. The system's real-time processing capability was enabled and the processing load was much lower because an oversized  $n \times n$  detection window was not used and only specific image areas, regions of interest (ROI), were processed. The pedestrian motion vector detection and warning system presented here has three components: The first serves to identify, locate and mark the positions of pedestrians in images using the Fisher classifier. The follow-up image processing is confined to the labeled images, and this significantly reduces the image postprocessing load. In the second part, the Lucas–Kanade optical flow method was used to calculate the pedestrian movement vectors so that their forward moving rate and direction could be determined. The last step is the establishment of the area into which the vehicle will move in the future. Judgment is then made as to whether a pedestrian will move into the driving area of the vehicle or not. A signal can then be generated to alert the driver to such an eventuality.

## 2. Pedestrian Detection and Estimation of Pedestrian Movement

### 2.1 Pedestrian detection

In our previous work, the pedestrian detection procedure is accelerated using a two-layer cascade of classifiers.<sup>(14)</sup> At the front end, the Fisher classifier using Haar-like features<sup>(15–17)</sup> can rapidly select candidate regions in the image where pedestrians may be present. At the back end, the Fisher classifier using a covariance matrix descriptor<sup>(18)</sup> can accurately determine if pedestrians are positioned in the candidate regions. If a region is determined as positive by the two-layer cascade classifiers, images that might include pedestrians are captured and delineated by a rectangular frame. Image processing is only carried out on the marked area and the computational load is kept to a minimum.

### 2.2 Estimation of pedestrian movement

#### 2.2.1 Pedestrian corner features

After a pedestrian has been identified by a frame, the Harris corner detection algorithm<sup>(19)</sup> is applied to find the corners of the pedestrian's contour. It tests each pixel in the rectangular frame that contains a pedestrian image to see if a corner is present. Harris corner detection is developed based on the Moravec algorithm. The basic principle is taking a small window of size  $n \times n$  centering on a target pixel, moving the window separately along every 45 degrees (horizontal, vertical, and on the two diagonals), and calculating the gradation change of the window in eight directions. The degree of gradation change between two nearby windows in the image  $I$  that centered on  $(u, v)$  and  $(u + \Delta x, v + \Delta y)$  respectively, is then measured, and denoted in Eq. (1).

$$S(\Delta x, \Delta y)_{u,v} = \sum_u \sum_v w(u, v) (I(u + \Delta x, v + \Delta y) - I(u, v))^2 \quad (1)$$

$w(u, v)$  is a weighted function, and a 2D Gaussian function is generally adopted. Next,  $I(u + \Delta x, v + \Delta y)$  can be approximated by a Taylor expansion, and Eq. (1) can be rearranged as

$$S(\Delta x, \Delta y)_{u,v} \approx [\Delta x \quad \Delta y] A [\Delta x \quad \Delta y]^T, \quad (2)$$

$$A = \sum_u \sum_v w(u, v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}. \quad (3)$$

$I_x$  and  $I_y$  are the partial derivatives of  $I$  in  $x$ - and  $y$ -directions, respectively. A corner is characterized by a large variation of  $S(\Delta x, \Delta y)_{u,v}$  in all directions, which means that the matrix  $A$  will have two large eigenvalues  $\lambda_1$  and  $\lambda_2$ .

Since the calculation of the eigenvalues of matrix  $A$  in Eq. (3) is computationally expensive, Harris and Stephens<sup>(19)</sup> suggest Eq. (4) to verify a corner and make the computation more efficient. The algorithm does not have to actually compute the eigenvalue decomposition of the matrix  $A$ ; it is sufficient to evaluate the determinant and the trace of  $A$  to find a corner.  $M_c$  is positive in the corner region, negative in the edge regions, and small in the flat region.

$$M_c = \lambda_1 \lambda_2 - \kappa (\lambda_1 + \lambda_2)^2 = \det(A) - \kappa \cdot \text{trace}^2(A) \quad (4)$$

Here, the value selection of  $\kappa$  will affect the composition of corners, and the empirical values are generally between 0.04 and 0.06. Corner detection is only carried out for the area within the rectangular frame, in other words, the region with pedestrians. As for the images of the contour area, the horizontal and vertical gradient values  $I_x$  and  $I_y$ , respectively, are higher. To speed up the corner feature calculations, corner detection is concentrated mainly on the image area with higher  $I_x$  and  $I_y$  values.

The results of the calculations are corner characteristics that are distributed over the head, hands, body, and feet, as shown in Fig. 1(a). Since the hands and feet move back and forth rather much as a person walks or runs and the magnitude and direction of their movement are not consistent, they are not representative of a pedestrian moving at a steady speed. The head and body swing less and their movement vectors are a better choice for calculation. Again, significant computation time can be saved when calculation is confined to the characteristic data of the head alone. Therefore, in the actual tests, we only calculated the motion vectors using corner characteristic data from a quarter of the upper part of the rectangular frame, as shown in Figs. 1(b) and 1(c).



Fig. 1. (Color online) Results of corner feature detection.

### 2.2.2 Optical flow algorithm

After selecting the corner characteristics of the pedestrian's head contour to represent their movement characteristics, the Lucas–Kanade optical flow algorithm<sup>(20,21)</sup> was used to calculate the movement vectors of the corner features. The Lucas–Kanade method can be used with two consecutive image frames,  $I(t)$  and  $I(t + 1)$ , to calculate the movement vectors of an object based on the following three assumptions. Assumption 1: There is no marked difference in the brightness of the two consecutive images. For example, if one image was acquired in bright sunlight and the other in the shade, the variation of brightness may be too large for a vector calculation. Assumption 2: The distance moved between successive images cannot be too large. Where a fixed  $n \times n$  detection window is used, the object must remain within the window to be tracked. If motion displacement takes the object outside the window area, tracking will fail. Assumption 3: The detected pixels and the adjacent pixels move in the same direction.

Thus, the Lucas–Kanade optical flow method is assumed to hold for all pixels within an  $n \times n$  window centered at  $p$  in image  $I(t)$  for calculating pedestrian movement. The local image flow (velocity) vector  $\mathbf{d} = [v_x \ v_y]^T$  must satisfy

$$\mathbf{d} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \mathbf{B}^{-1} \mathbf{C}, \quad (5)$$

where

$$\mathbf{B} = \begin{bmatrix} \sum_i w_i I_x(q_i)^2 & \sum_i w_i I_x(q_i) I_y(q_i) \\ \sum_i w_i I_y(q_i) I_x(q_i) & \sum_i w_i I_y(q_i)^2 \end{bmatrix}^{-1},$$

$$\mathbf{C} = \begin{bmatrix} - \sum_i w_i I_x(q_i) I_t(q_i) \\ - \sum_i w_i I_y(q_i) I_t(q_i) \end{bmatrix}.$$

$q_1, q_2, \dots, q_{n^2}$  are the pixels inside the  $n \times n$  window.  $I_x(q_i)$ ,  $I_y(q_i)$ , and  $I_t(q_i)$  are the partial derivatives of the image  $I$  in the  $x$ -direction,  $y$ -direction, and time  $t$ , evaluated at the point  $q_i$  and at the current time. In practice, the weight  $w_i$  is usually set to a Gaussian function of the distance between  $q_i$  and  $p$ .

However, if the pedestrian moves too fast, they might be outside the  $n \times n$  window by the time the next image is captured. A larger window is one solution but that would increase computation by  $O(n^2)$ . The pyramid Lucas–Kanade optical flow method<sup>(22)</sup> shrinks the image into several layers where the width and height of each layer is half that of the preceding one. The same  $n \times n$  window can detect the characteristic displacement values in the shrunken images. Assuming that the shrunken width/height of each image is half that of the original, and the original image is layer 0, the displacement velocity of each layer is as shown in Eq. (6), where  $d$  is the displacement value,  $L$  is the layer number, and  $L_{max}$  is the maximum scalable layer.

Define the pyramid representation of the original image as  $I$  of size  $N \times M$ . Let  $I^0 = I$  be the image at level 0. The width and height of the image  $I^0$  at that level are defined as  $N^0 = N$  and  $M_0 = M$ , respectively. Then,  $N^L$  and  $M^L$ , the width and height of the image  $I^L$  at level  $L$ , respectively, can be computed as  $N^L = \frac{N^0}{2^L}$  and  $M^L = \frac{M^0}{2^L}$ . The image  $I^L$  can be computed from  $I^{L-1}$  in a recursive

fashion, where  $L = 1, 2, \dots, L_{max}$ . Recall the goal of affine feature tracking, for a given point  $p$  in image  $I^L(t)$  at level  $L$ , and find its corresponding location  $q = p + d$  in image  $I^L(t + 1)$ . Image  $I^L(t + 1)$  is the subsequent frame of image  $I^L(t)$  in time.  $d^L = [v_x^L \ v_y^L]^T$  is the optical flow vector of point  $p$  in image  $I^L$ . The optical flow vector  $d^L = [v_x^L \ v_y^L]^T$  is found in Eq. (5). Then, the optical flow  $d^0$  in image  $I^0$  is estimated as

$$d^0 = 2^L d^L. \quad (6)$$

Take the coordinates of the characteristic point  $p$  to be  $(x^L, y^L)$  for the  $I^L(t)$  image, and then the optical flow value of  $d^L = [v_x^L \ v_y^L]^T$  can be obtained for the  $n \times n$  window through the pyramid Lucas–Kanade optical flow method. This means that the coordinates of the next image  $I^L(t + 1)$  have reflected a movement to the coordinate of  $(x^L + v_x^L, y^L + v_y^L)$ . Next, use the residual equation as Eq. (7) to determine if the tracking of the characteristic point  $p$  is valid. The residual formula calculates such characteristic point differences (including the neighboring area of  $n \times n$ ) between successive images, performing a similar calculation. A low residual differential value indicates that the characteristic point of the next movement can be tracked in the next image. This means that the calculated optical flow value  $d^L$  is correct. Ideally, the residual values should approach zero. On the other hand, larger residual values are an indication that the characteristic points have not been correctly tracked. In such a case, the calculated optical flow value means nothing. If we cannot correctly track the characteristic points because the pedestrian has moved too fast or has moved out of the  $n \times n$  window, the remedy would be to shrink the images and recalculate the optical flow and residual differential values, using the results to decide if the optical flow values are meaningful. The images should be repeatedly shrunk until the displacements of the characteristic points can be tracked. The optical values  $d^L$  of the characteristic points from  $I^L$  images have been successfully tracked when the calculations show sufficiently small residual differential values. We can then use Eq. (6) to get  $d^0$ , and in turn obtain the movement vectors in image  $I^0$ . Of course, it may be possible that the objects move so fast so that correct optical flow values cannot be obtained from the  $I^{L_{max}}$  images. In our study, we have chosen a setting of  $L_{max} = 2$  and consider this to be a reasonable moving speed suitable for normal pedestrian tracking.

$$\varepsilon(d^L) = \varepsilon(v_x^L \ v_y^L) = \sum_{x=v_x^L - \frac{n}{2}}^{v_x^L + \frac{n}{2}} \sum_{y=v_y^L - \frac{n}{2}}^{v_y^L + \frac{n}{2}} \{I^L(x + v_x^L, y + v_y^L, t + 1) - I^L(x, y, t)\}^2 \quad (7)$$

### 2.2.3 Focus expansion

The cameras used in this study were mounted on the front windshield of the vehicles and commanded an unobstructed view of the road ahead. However, as a vehicle moves forward, focal expansion takes place (as shown in Fig. 2). Objects such as trees, buildings, posts, and other immovable objects expand from the focal point and move outwards. In other words, even pedestrians who are standing still will appear to move in successive frames. This is reflected by changes in the coordinates, which are dependent on the relative speed of the vehicle  $v_c$ . Particularly, the optical flow  $v_f$  acquired from the images does not represent actual pedestrian movement vectors. The real pedestrian movement vectors should be  $v_p = v_f - v_c$ . The pedestrian shown in Fig. 3 is standing still, and when the vehicle travels forward at a speed  $v_c$ , the  $(x_1, y_1)$



Fig. 2. (Color online) Focal point expansion.<sup>(26)</sup>



Fig. 3. (Color online) (a) Coordinate of a standing pedestrian in image  $I(t)$ . (b) Coordinate of a standing pedestrian in image  $I(t+1)$ .

position coordinate of the pedestrian from  $I(t)$  images is as shown in Fig. 3(a), where the position coordinate in the  $I(t + \Delta t)$  image moves to  $(x_2, y_2)$ , as shown in Fig. 3(b). During this time ( $\Delta t$ ), the coordinate changes in the subsequent images are due to the forward movement of the vehicle at speed  $v_c$ .

### 3. Vehicle Driving Areas

The lane model plays a very important role in our pedestrian warning system, as shown in Fig. 4. As already mentioned, a single CCD camera was installed behind the front windshield of the vehicle to capture images of the road ahead. In the image, the horizontal axis ( $u$ ) represents the direction of deviation (left or right) and the vertical axis ( $v$ ) represents the driving direction of the car.

We can use road configurations and settings as well as driving habits (for example, some drivers tend to favor the right or the left side of the lane) to analyze the statistical results and construct a lane model. The lane model describes possible paths that might be taken by a vehicle. This lane model is constructed using the location of the lane markings on both the left and right sides, namely,  $u_{li}$  and  $u_{ri}$ . A covariance matrix that describes the potential range of fluctuation is also used in the lane model.

We referred to the specifications for main roads in Taiwan<sup>(23)</sup> and other statistical data relating to road design and used five parameters to describe the locations of the lane markings in the

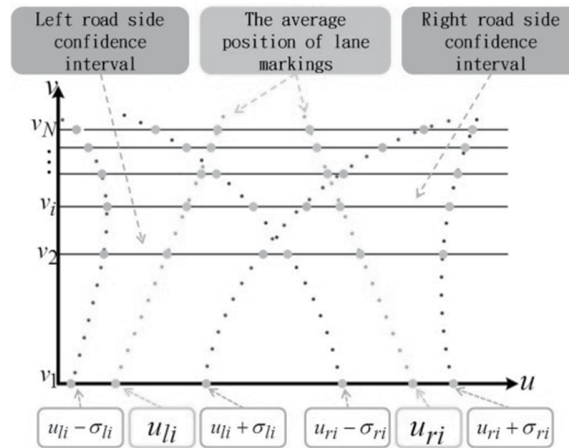


Fig. 4. Lane model construction.

image:<sup>(24)</sup> (1) the camera inclination angle  $\alpha$ , which is the angle of the camera as installed in the car; (2) the vehicle steering angle  $\psi$ ; (3) the lateral curve  $C_l$ ; (4) the road width  $L$ ; and (5) vehicles on the carriageway in the transverse position  $x_0$ , as shown in Fig. 5 and Table 1. An equation was then established corresponding to the locations of the lane markings in the image:

$$u_i = e_u \left[ \frac{-e_v z_0}{2(v_i - e_v \alpha)} C_l - \psi - \frac{d_u}{2} + \frac{v_i - e_v \alpha - \frac{d_v}{2}}{e_v z_0} \left( x_0 + \frac{\gamma L}{2} \right) \right]. \quad (8)$$

When  $\gamma = 1$ ,  $u_{li} = u_i$ .

When  $\gamma = -1$ ,  $u_{ri} = u_i$ .

$e_u = f/d_u$  and  $e_v = f/d_v$ .

$f$  = camera focal length.

$d_u, d_v$  = number of vertical and horizontal pixels of the captured image.

$z_0$  = height of the installed camera.

$C_l$  = lateral curve of the lane.

$v_i$  = vertical location in the image, which presents the distance away from the moving vehicle.

Mean values taken for the five parameters from Table 1 were input into Eq. (8) where  $\gamma = 1$ . We then input  $v_i$  ( $i = 1, \dots, N$ ) into the equation to obtain  $N$  number of  $u_i$  (at this time,  $u_{li} = u_i$ ). These are scalar values, and a collection of  $N$  number of scalar-valued parameters is known as the parameter vector  $\mathbf{X} = [u_1, u_2, \dots, u_N]_{1 \times N}$ . The parameter vector  $\mathbf{X}$  describes the average location of the lane markings in a typical road image, that is, the location where the lane markings most commonly appear. We then find the first-order derivative for these five parameters to obtain the Jacobian matrix, as shown in Eq. (9).



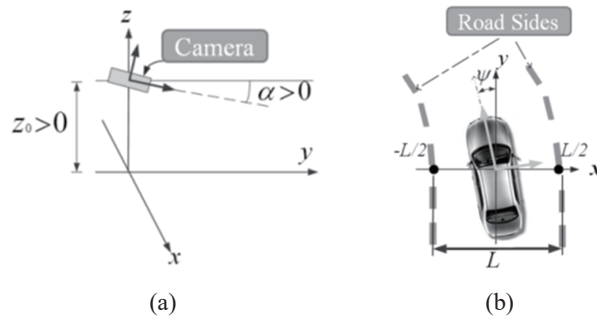


Fig. 5. (a) Location of the camera installed in the car. (b) Relationship between the car and the lane.

Table 1  
Lane location parameters.

	$\alpha$ (°)	$\psi$ (°)	$C_l$ (m <sup>-1</sup> )	$L$ (m)	$x_0$ (m)
Mean value	3	0	0	3.6	0
Standard deviation	0.5	3	0.017	0.3	1.8

$$\mathbf{J}_g = \begin{pmatrix} \frac{\partial u_1}{\partial x_0} & \frac{\partial u_1}{\partial \psi} & \frac{\partial u_1}{\partial C_l} & \frac{\partial u_1}{\partial \alpha} & \frac{\partial u_1}{\partial L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_i}{\partial x_0} & \frac{\partial u_i}{\partial \psi} & \frac{\partial u_i}{\partial C_l} & \frac{\partial u_i}{\partial \alpha} & \frac{\partial u_i}{\partial L} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{\partial u_N}{\partial x_0} & \frac{\partial u_N}{\partial \psi} & \frac{\partial u_N}{\partial C_l} & \frac{\partial u_N}{\partial \alpha} & \frac{\partial u_N}{\partial L} \end{pmatrix}_{N \times 5} \quad i = 1, \dots, N \quad (9)$$

We multiply the Jacobian matrix obtained in Eq. (9) by the  $C_p$  matrix, where  $C_p$  is the diagonal matrix obtained from the standard deviations of the five parameters listed in Table 1. A covariance matrix  $C_x$  (an error covariance matrix) can be obtained, as shown in Eq. (10).

$$\mathbf{C}_x = (\mathbf{J}_g \mathbf{C}_p \mathbf{J}_g^T) = \left( \mathbf{J}_g \begin{pmatrix} \sigma_\alpha^2 & 0 & 0 & 0 & 0 \\ 0 & \sigma_\psi^2 & 0 & 0 & 0 \\ 0 & 0 & \sigma_{C_l}^2 & 0 & 0 \\ 0 & 0 & 0 & \sigma_L^2 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{x_0}^2 \end{pmatrix} \mathbf{J}_g^T \right) = \begin{pmatrix} \sigma_1^2 & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \sigma_i^2 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \sigma_N^2 \end{pmatrix}_{N \times N} \quad i = 1, \dots, N \quad (10)$$

Lastly, we construct our lane model using the parameter vector  $\mathbf{X}$  and the covariance matrix  $C_x$ , as shown in Fig. 4.  $u_{li}$  and  $u_{ri}$  are calculated on the basis of Eq. (10), the parameter vector  $\mathbf{X}$ .  $u_{li} - \sigma_{li}$ ,  $u_{li} + \sigma_{li}$ ,  $u_{ri} - \sigma_{ri}$ , and  $u_{ri} + \sigma_{ri}$  are calculated on the basis of the diagonal elements  $\sigma$  of the covariance matrix  $C_x$  and the parameter vector  $\mathbf{X}$ . We define the range enclosed by  $u_{li} - \sigma_{li}$  and  $u_{ri} + \sigma_{ri}$  as a dangerous zone for pedestrians through which a car might drive.

The pedestrian collision warning system will alert the driver when a pedestrian is within the danger zone, or if a previously obtained pedestrian movement vector shows them to be heading for

the danger zone, as shown in Fig. 6(a). When the vector indicates movement away from the danger zone, the path ahead of the moving vehicle is considered to be safe, as shown in Fig. 6(b).

#### 4. Experimental Results and Discussion

In this article, the tests have been performed mainly on urban roads during daytime and using a single camera mounted in the center of the dash in front of the windshield of an ordinary car to capture images of the road ahead. The specifications of the camera used in this study are listed in Table 2. The experimental results and analysis are presented in three parts: pedestrian detection, pedestrian movement vector, and danger zone warning.

##### 4.1 Pedestrian detection

The pedestrian image database was taken from The National Institute for Research in Computer Science and Control (INRIA),<sup>(25)</sup> where there are 2416 positive and 1218 negative samples in the training set, and 1126 positive and 453 negative samples in the test set. The first step was the use of positive samples from the INRIA person training dataset to train the classifiers. Random samples selected from the negative sample training set were also used. Up to 2000 images were randomly selected from a positive sample training set as well as an equal number of negative samples. With 2000 samples, the true negative of the Haar-like characteristics from the front end of the Fisher classifier was 1.3%, and the false positive was 45.6%. At the back end of the Fisher classifier data, based on covariance matrix descriptors, the true negative was 7% and the false positive was 6.6%.

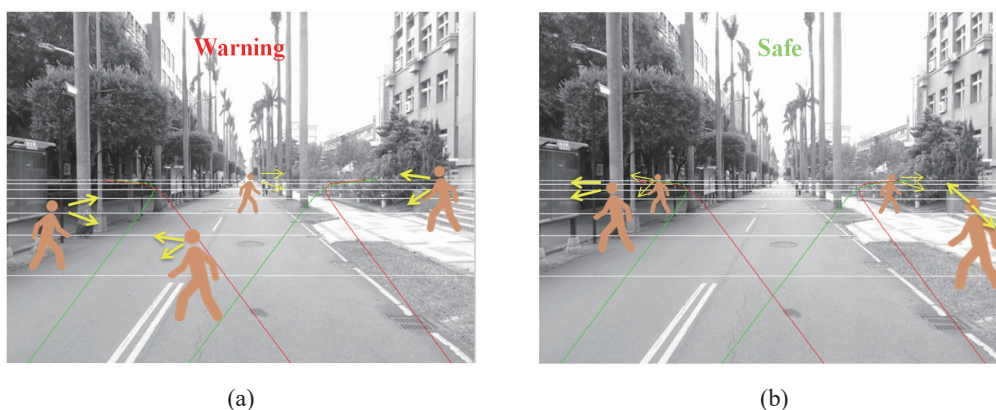


Fig. 6. (Color online) (a) Pedestrian moving into the danger zone. (b) Pedestrian moving out of the danger zone.

Table 2

Camera specifications.

Item	Specifications
Optical sensing element	CMOS
Focal length	29 mm
Resolution	640 × 360
Camera setup height	120 cm
Camera tilt angle	1°

The advantage of using Haar-like characteristics for identification is its fast computation speed. It can very quickly search for areas that may contain pedestrians. The experimental results show that smaller true negative values correspond to lower false pedestrian identification. Relatively false positive corresponds to almost 50% of false judgments. In other words, nonpedestrians could be mistaken as pedestrians. To reduce the number of misidentifications, we used cascaded front-end and back-end classifiers to reidentify the image areas filtered by the front-end classifier. The use of the back-end classifier significantly reduced the false positive ratio. Incidentally, the images used for testing included both stationary and moving pedestrians. Some experimental results obtained with this pedestrian detection system are presented in Fig. 7.

#### 4.2 Pedestrian movement flow

After the pedestrian positions have been delineated by rectangular frames, the follow-up image processing algorithms need only take care of the upper quarter of these frames, i.e., the ROI, and the amount of computation is relatively small. The sequence of ROI computation is as follows:

To start with, Harris corner detection is used to find the corner feature points of the head outline. Next, the movement vectors  $v_f$  are calculated from the corner point data using the pyramid optical flow method. The vehicle forward moving speed  $v_c$  is then used to correct for focal expansion and to determine the actual movement vectors  $v_p$  of the pedestrian(s). In our experiments, the vehicle forward moving speed  $v_c$  is limited at 40 km/h for the consideration of testing safety. Some experimental results of the pedestrian movement estimation are shown in Fig. 8, and the pedestrian movement vectors are indicated in the figure. In Fig. 9, the danger zone in the vehicle driving forward path is specified on the basis of the lane model. When a pedestrian is within the danger zone, or the pedestrian motion vectors indicate possible movement into the danger zone, the system will alert the driver to this eventuality by generating a visual or audible driver warning.

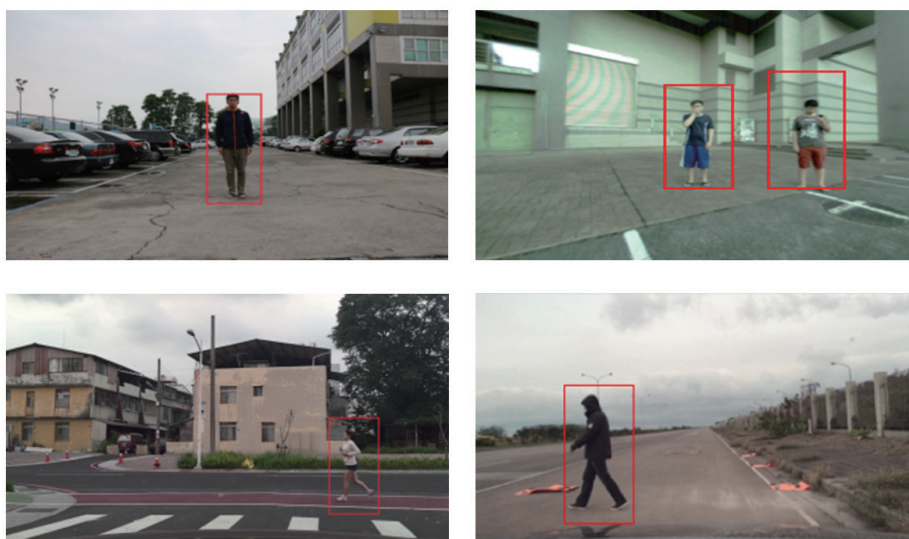


Fig. 7. (Color online) Results of pedestrian detection.



Fig. 8. (Color online) Results of pedestrian movement estimation.



Fig. 9. (Color online) Danger zone warning experimental results.

### 4.3 Discussion

In this study, we have used a number of available technologies to devise a simple ADAS system to improve the safety of pedestrians and the drivers of motor vehicles, more particularly passenger cars. Pedestrians are often very seriously hurt in road accidents, and in this study, we focused on protection and the prevention of such eventuality. Computer visual technology is widely used nowadays for all types of object identification, but the differences between pedestrians and rigid, mostly immovable objects, have been difficult to determine on the basis of singular characteristic data. Most previous studies used multiple characteristics to identify a pedestrian, and this approach makes it necessary to use a huge database to train the learning classifiers. The image algorithms consume a correspondingly large amount of computing power and time. The pedestrian identification algorithms presented in our study consist of two cascaded classifiers. The front end uses singular Haar-like characteristic data for the rapid identification of possible pedestrians. Although the back-end classifier needs to take care of multi-characteristic data to make classification decisions, the computational load is quite light because only image areas that have been prefiltered by the front-end classifier are processed. For the calculation of

Table 3  
Processing time for each function.

Detected item	Time spent
Pedestrian detection	$0.2 \pm 0.05$ s/frame
Pedestrian movement vector calculation	$0.27 \pm 0.05$ s/frame
Danger judgment	$0.02 \pm 0.01$ s/frame
Total	$0.49 \pm 0.11$ s/frame

pedestrian movement vectors, only the corner feature points of the heads of pedestrians are used, which provides overall moving velocities and directions. This is both effective and workable. This method simplifies image processing computation on the premise that the effectiveness of identification shall not be adversely affected in any way. However, each image shown in Table 3 would take about 0.5 s to process using a PC with an Intel Core i5 CPU and 4 GB RAM executing the algorithms coded with Matlab. Even if the requirements of a real-time operation could be satisfactorily carried out using hardware circuitry, the cost will still be relatively high. We will continue our efforts to find even simpler image algorithm alternatives and also investigate the use of parallel processing.

## 5. Conclusions

In this study, we utilized a visual image processing method to estimate pedestrian movement vectors and issue anticollision warnings. Two steps are involved, pedestrian detection and pedestrian movement vector estimation, each of which could be established as independent modules. Pedestrian identification is made by two-layered cascaded classifiers. The front-end classifier uses Harr-like characteristic data for rapid classification and the back-end classifier uses covariance descriptor characteristics to reclassify the prefiltered image zone and reach correct identification with 96% certainty. In addition, the postprocessing algorithm of pedestrian movement vectors focuses only on the ROI, and this raises the system processing speed significantly. The pyramid movement vector algorithm is even more efficient for estimating pedestrian movement. Since the speed at which pedestrians move is not very high, a two-layer pyramid optical flow algorithm and  $5 \times 5$  tracking window are sufficient. The pedestrian movement vector estimation and safety warning system designs easily cope with the real-time processing applications. The design of this visual image processing system takes both quantitative calculations and time into consideration.

## References

- 1 P. Dollar, C. Wojek, B. Schiele, and P. Perona: IEEE Trans. Pattern Anal. Mach. Intell. **34** (2012) 743.
- 2 M. Enzweiler and D. M. Gavrilu: IEEE Trans. Pattern Anal. Mach. Intell. **31** (2009) 2179.
- 3 Y. C. Kuo, C. M. Fu, C. T. Tsai, C. C. Lin, and C. H. Chang: Symp. Computer, Consumer and Control (IEEE, Xian, 2016) p. 740.
- 4 J. C. McCall and M. M. Trivedi: IEEE Trans. Intell. Transp. Syst. **7** (2006) 20.
- 5 P. Y. Hsiao, C. W. Yeh, S. S. Huang, and L. C. Fu: IEEE Trans. Veh. Technol. **58** (2009) 2089.
- 6 B. F. Lin, Y. M. Chan, L. C. Fu, P. Y. Hsiao, L. A. Chuang, S. S. Huang, and M. F. Lo: IEEE Trans. Intell. Transp. Syst. **13** (2012) 737.

- 7 C. T. Lin, C. J. Chang, B. S. Lin, S. H. Hung, C. F. Chao, and I. J. Wang: *IEEE Trans. Biomed. Circuits Syst.* **4** (2010) 214.
- 8 F. C. Lin, L. W. Ko, C. H. Chuang, T. P. Su, and C. T. Lin: *IEEE Trans. Circuits Syst. I-Regul. Pap.* **59** (2012) 2044.
- 9 Y. C. Lin, C. T. Lin, W. C. Liu, and L. T. Chen: *Conf. Industrial Electronics and Applications (IEEE, Kota Kinabalu, 2013)* p. 1627.
- 10 E. Rosten, R. Porter, and T. Drummond: *IEEE Trans. Pattern Anal. Mach. Intell.* **32** (2010) 105.
- 11 D. Langer and T. Jochem: *Symp. Intelligent Vehicles (IEEE, Pittsburgh, 1996)* p. 333.
- 12 C. G. Keller and T. Dang: *IEEE Trans. Intell. Transp. Syst.* **12** (2011) 1292.
- 13 J. D. Alonso, E. R. Vidal, A. Rotter, and M. Mühlenberg: *IEEE Trans. Veh. Technol.* **57** (2008) 2736.
- 14 Y. C. Kuo, Z. Y. Yang, and C. H. Yen: *Elsevier J. Comput. Math. Appl.* **64** (2012) 1311.
- 15 P. Viola and M. J. Jones: *Int. J. Comput. Vis.* **57** (2004) 137.
- 16 P. Viola and M. J. Jones: *Proc. Computer Vision and Pattern Recognition (IEEE, Kauai, 2001)* p. 511.
- 17 C. P. Papageorgiou, M. Oren, and T. Poggio: *Conf. Computer Vision (IEEE, Bombay, 1998)* p. 555.
- 18 O. Tuzel, F. Porikli, and P. Meer: *Proc. Eur. Conf. Computer Vision 2 (Springer, Graz, 2006)* p. 589.
- 19 C. Harris and M. Stephens: *Proc. Alvey Vision Conf. (Plessey, Manchester, 1988)* p. 147.
- 20 B. D. Lucas and T. Kanade: *Proc. Artificial Intelligence (Morgan Kaufmann, Vancouver, 1981)* p. 674.
- 21 J. Shi and C. Tomasi: *Conf. Computer Vision and Pattern Recognition (IEEE, Seattle, 1994)* p. 593.
- 22 J. Y. Bouguet: *Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the Algorithm, Intel Corporation Microprocessor Research Labs (2000)* p. 1.
- 23 Q. Y. Chen, Z. P. Sun, T. T. Chen, and L. Cao: *Conf. Vehicular Electronics and Safety (IEEE, Beijing, 2011)* p. 65.
- 24 Q. Y. Chen, Z. P. Sun, X. J. An, D. Liu, J. Song, and B. Zhang: *Conf. Intelligent Transportation Systems (IEEE, Washington, 2011)* p. 1727.
- 25 P. Viola and M. J. Jones: *Int. J. Comput. Vision* **57** (2004) 137.
- 26 B. Goldstein: Thomson Higher Education, [https://classconnection.s3.amazonaws.com/932/flashcards/1253932/png/flow\\_of\\_environment\\_711366759435046.png](https://classconnection.s3.amazonaws.com/932/flashcards/1253932/png/flow_of_environment_711366759435046.png) (accessed April 2016).