A Smart Moving Vehicle Detection System Using Motion Vectors and Generic Line Features

Chup-Chung Wong, *Member*, IEEE, Wan-Chi Siu, *Fellow*, IEEE, Paul Jennings, Stuart Barnes and Bernard Fong, *Senior Member*, IEEE

Abstract — *This paper presents a novel algorithm for the detection and tracking of low relative speed moving vehicles. The proposed algorithm is particularly suitable for mass-produced in-vehicle devices as it combines with motion vector based moving object detection to form a complete solution for an Advanced Driver Assistance System given its reduction in cost and complexity. The algorithm utilizes motion vectors that are readily available from video encoder output. The region of interest for detection is reduced by ignoring the area above the vanishing line of the captured image, evaluation of the amplitude of motion vectors and identification of the road region. During the evaluation process, a binary image is generated by comparing the gray-level of the captured image to the gray-level of the detected road region. Subsequently, the horizontal and vertical contours of specific areas inside the region of interest are evaluated. Test results show the effectiveness of the algorithm with more than 90% detection rate and the suitability for real-time use with cycle time of less than 66ms*[1].

Index Terms — Vehicle detection, ADAS, motion vector, H.264/AVC.

I. Introduction

Advanced Driver Assistance Systems (ADAS) have been employed in many automobiles to assist drivers for improved road safety. For image based ADAS, there have been many proposed methods for rear-view vehicle detection [1]. These methods can be classified as feature based, statistical based and optical-flow based methods.

For the feature based method, vehicles can be detected based on prior knowledge of their characteristics. For instance, some related work makes use of the rectangular shape and strong symmetry of the rear-view vehicles for detection [2]-[4]. Instead of using the symmetrical feature directly from vehicles, a detection method that analyzes the shadow underneath a vehicle has also been proposed [5].

For statistical based approaches, feature extraction based on Haar [6], [7] and Histogram of Oriented Gradients (HOG) [8], [9] has been reported. They combine with the use of statistical training algorithms such as Adaboost [10] or Support Vector Machine (SVM) [11] to generate the classifiers for high true-positive and low false-positive detection rates.

For optical flow based detection, there have been studies on the estimation of ego-motion by analyzing the optical flow vectors between successive frames [12], [13]. These methods depend on the optical flow vectors from static background for reliable estimation of the perspective transformation matrix between successive frames. Few studies on moving vehicle detection exist on a moving platform using motion vectors (MVs) from the video encoder [14].

The motivation of this work is to reduce the cost and complexity of mass-produced vision based driver assistance systems. Since MVs are readily available from H.264/AVC video encoders, the shared use of the MVs for moving object detection does not entail additional computation intensive processes, such as dense optical flow, for evaluating the MVs in the captured image. According to Zhang et al. [15], at least one Digital Signal Processor is required for real-time optical flow evaluation. It is estimated that 30% of the hardware cost can be saved due to the elimination of a Digital Signal Processor for dense optical flow evaluation. Also, feature based object detection requires different algorithms for different types of objects to be detected, demanding for additional computational resources to detect each additional type of object. In contrast, there is only one type of object – the moving object, is to be detected by the shared use of MVs. The required computational cost can therefore be lowered. While video recording function is available by default due to the presence of H.264/AVC encoder, the system cost of the ADAS can also be reduced. Although the proposed technique is targeted to be used in ADAS, it can be extrapolated to many Consumer Electronics applications. For instance, the aftermarket consumer-grade on-board video blackbox is one of the applications that can make use of the proposed technique immediately.

This paper addresses the problem of detecting a rear-view vehicle with slow relative speed near the focus of expansion (FOE) of the moving camera. The proposed method is suitable for use in conjunction with MV based moving object detection. The major contribution of this work is to use the MVs from H.264/AVC encoder, dividing the region of interest for slow relative speed vehicle detection using the generic line features,

[1] This work was supported by the Innovation and Technology Commissions of the Hong Kong S.A.R., R&D Project No. ITP/015/07AP.

Chup-Chung Wong is with WMG, University of Warwick, Coventry, U.K., and Hong Kong Productivity Council, Hong Kong (e-mail: elvin_wong@hotmail.com).

Wan-Chi Siu is with Hong Kong Polytechnic University, Hong Kong. (e-mail: enwcsiu@polyu.edu.hk).

Paul Jennings is with WMG, University of Warwick, Coventry, UK (e-mail: paul.jennings@warwick.ac.uk).

Stuart Barnes is with WMG, University of Warwick, Coventry, UK (e-mail: s.barnes@warwick.ac.uk).

Bernard Fong is with the Hong Kong Automotive Parts and Accessory Systems R&D Centre, Hong Kong (e-mail: bernardfong@hkpc.org).

Contributed Paper
Manuscript received 07/09/15
Current version published 09/29/15
Electronic version published 09/29/15.

0098 3063/15/$20.00 © 2015 IEEE

and for fast relative speed vehicle detection using planar parallax residual. Experimental results show that the proposed method has a performance of over 90% detection rate with a low computational cost, making it particularly suitable for implementation in a low-cost mass produced consumer electronics device.

II. The Architecture and Algorithm Framework

The proposed system architecture of the MV based ADAS is shown in Fig. 1. It consists of a CMOS camera sensor unit, a six degree-of-freedom inertial sensor, a Digital Signal Processor System-on-Chip (DSP SOC), an SD-Card interface and a display and warning device. The six degree-of-freedom inertial sensor is mounted directly to the printed circuit board for the CMOS camera sensor in a position which is very close to the optical centre of the CMOS camera sensor. The inertial sensor is able to measure the 3-dimensional acceleration and angular motion of the camera. Therefore, the instantaneous tilt, roll and yaw angle of the camera relative to the earth plane can be obtained. The signal from the vehicle speed sensor is also fed into the DSP SOC. Therefore, the travelled distance by the vehicle can be calculated from the vehicle speed sensor and the time interval between successive image frames. The SD card interface is for inserting an SD-Card for compressed video recording. The display and warning device is for alerting the driver of a dangerous situation, such as when the time-to-collision is less than 2 seconds. The DSP SOC is the heart of the system. It is responsible for executing all algorithms for lane detection and moving object detection.

The execution sequence of the DSP SOC for detecting slow relative speed vehicle is shown in Fig. 2. Two threads are running in parallel in the DSP as indicated by the two dashed-line blocks. Images and dynamic data are captured simultaneously from the CMOS camera and the inertial sensor. The dynamic data includes the acceleration, angular rate and elevation angle along the 3-dimensional axis on the camera coordinates, and the number of speed pulse returned between two successive frames.

The dynamic data and camera calibration parameters are used for estimating the focus of expansion (FOE) of the camera at the current frame. The resultant optical flow vectors on the captured image are the result of translational and rotational motion [16]. Pure translational motion results in optical flow vectors pointing to the FOE. The primary region of interest (ROI) of the captured image is the area below the "vanishing line". Vanishing line refers to the horizontal line at the *y*-axis location of the FOE. The road region will also be identified by a novel algorithm that tries to find the color similarity of macroblocks starting from the bottom of the image.

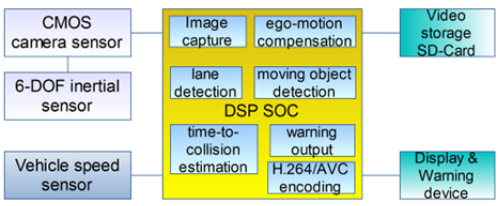


Fig. 1. System architecture of the proposed MV based ADAS

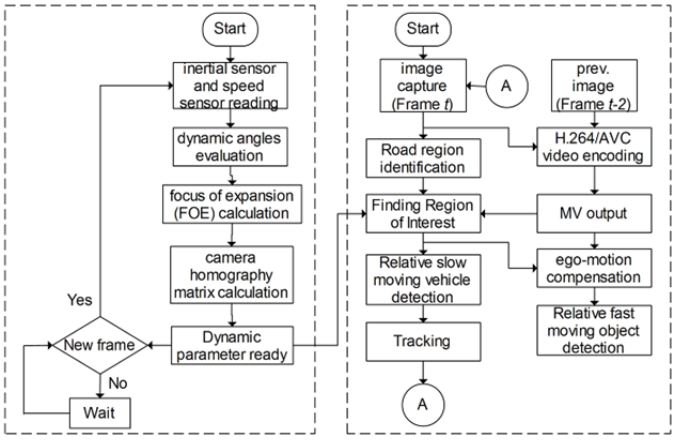


Fig. 2. Functional flowchart for slow relative speed vehicle detection

The ROI is further reduced by the amplitude of MVs. The detection and tracking of slow relative speed moving vehicles are performed inside the ROI. The detection of fast relative speed moving objects is performed after the ego-motion compensation to MVs in the ROI.

III. The Algorithm

The proposed algorithm has functional blocks in common for detecting relatively fast and slow moving objects. These include the acquisition of dynamic data for ego-motion compensation and vanishing point estimation, and the detection of road region to reduce the area of ROI. Relatively fast and slow moving objects are categorized by the amplitude of MVs of macroblocks in the image. Those macroblocks with MVs larger than a threshold q_m will be examined for relatively fast moving objects. Otherwise, relatively slow moving objects will be examined.

For the detection of slow relative speed moving vehicles, methods of getting dynamic parameters of the observing camera, ROI construction, vehicle detection, tracking and distance estimation will be described.

A. Definition of Coordinates

The right-handed coordinate system is employed for both the World coordinates and the camera coordinates. *X*-axis is pointing out of the paper, *Y*-axis is pointing downwards towards the ground, *Z*-axis is pointing forward aligning with the longitudinal moving direction of the vehicle.

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} x_s & y_s & 1 \end{bmatrix}^T = H_{WS} \begin{bmatrix} X_w & Z_w & 1 \end{bmatrix}^T \quad (2)$$

$$\begin{bmatrix} X_w & Z_w & 1 \end{bmatrix}^T = H_{SW} \begin{bmatrix} x_s & y_s & 1 \end{bmatrix}^T \quad (3)$$

For a point $[X_w\ Y_w\ Z_w\ 1]^T$ in the World coordinates, it can be related to its corresponding point $[x_s\ y_s\ 1]^T$ in the screen coordinates by (1), where a_{ij} are the components inside the 3× 4 transformation matrix. Assuming all points are lying on the

ground plane, then Y_w=0. The 3×4 matrix can be reduced to a 3×3 matrix H_{ws}, as shown in (2), which is the matrix for the conversion from world to screen coordinates. Similarly, the matrix H_{sw} shown in (3) is the inverse of H_{ws}. (3) is the conversion from screen to World coordinates.

B. Focus of Expansion Estimation

For a vehicle undergoing linear motion with the camera installed with zero roll, yaw and tilt angles, the FOE (x_v,y_v) can be expressed as in (4) [16], where f is the focal length of the camera, (x_0,y_0) is the vanishing point of the camera on the screen, V_x, V_y and V_z are the velocities along X-, Y- and Z-axis of the World coordinates respectively.

The vanishing point (x_o,y_o) can be expressed as in (5), where K and R are the camera's calibration and rotational matrix respectively.

$$\begin{bmatrix} x_v \\ y_v \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} + \begin{bmatrix} fV_x / V_z \\ fV_y / V_z \end{bmatrix} = \begin{bmatrix} x_0 + fV_x / V_z \\ y_0 + fV_y / V_z \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} x_0 & y_0 & 1 \end{bmatrix}^T = KR \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{5}$$

Since the camera in the ego vehicle is carefully installed so that the roll angle and yaw angle can be set to zero. The tilt angle varies due to the uneven road surface and acceleration of the vehicle. The tilt angle of the camera will affect the vanishing point. With roll and yaw angle equal to zero, (5) can be rewritten to (6), where (c_x,c_y) and θ_x are the coordinates of the principal point and the tilt angle of the camera respectively. The motion of the ego vehicle on the World coordinates is estimated by a simple motion model which is illustrated by the bird's eye view of a moving vehicle in Fig. 3.

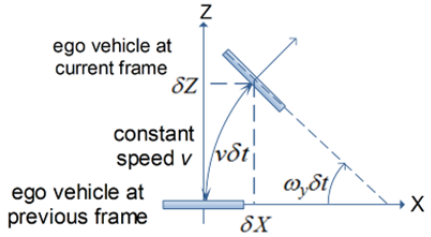


Fig. 3. Simple vehicle motion model in bird's eye-view. The vehicle travels at speed v, and turning at angular rate ω_y. Time interval between successive frames is δ_t.

$$\begin{aligned} \begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} &= \begin{bmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & \sin\theta_x \\ 0 & -\sin\theta_x & \cos\theta_x \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} c_x & c_y + f\tan\theta_x & 1 \end{bmatrix}^T \end{aligned} \tag{6}$$

Assuming the vehicle is moving at constant speed v with constant angular yaw rate ω_y, the time difference between successive frames is δt, and the distance travelled by the vehicle along the X- and Z-axis is δX and δZ respectively, where

$$\begin{cases} \delta X = v/\omega_y \left(1 - \cos\omega_y \delta t\right) \\ \delta Z = v/\omega_y \sin\omega_y \delta t \end{cases} \tag{7}$$

For small time difference δt between successive frames, the speed V_x and V_z in (4) can be approximated by $\delta X/\delta t$ *and* $\delta Z/\delta t$ respectively. The FOE at (x_v,y_v) can therefore be obtained since all parameters in (4) are known.

C. Fast Relative Speed Moving Object Detection

The detection of fast relative speed moving object is based on ego-motion compensation and the evaluation of planar parallax residual. Fig. 4 illustrates the planar parallax concept and the related planar parallax residual.

With reference to Fig. 4, a camera with zero tilt, yaw and roll angle is moving from position C_{t-1} in the previous frame at time *t-1* to C_t in the current frame at time *t.* A static object at P_w of the World coordinates in the current frame is seen by camera C_t at screen coordinate p_2. The same static object P_w is seen by camera C_{t-1} at screen coordinate p_1. The ground plane projections of the static object P_w with respect to the camera at positions C_t and C_{t-1} are P_2^G and P_1^G respectively, the corresponding screen point p_{2G} is the ground plane projection of P_2^G on camera C_{t-1}, Z_1^G and Z_2^G are the ground plane projection distances of the point P_w to the camera at positions C_{t-1} and C_t respectively. μ_w is the difference in ground plane projection distances of point P_w as seen by the camera at C_t and C_{t-1}.

The planar parallax residual μ is defined as the difference in pixel position of the static object P_w and its ground-plane projection in the screen coordinates, as expressed in (8).

For the camera coordinates of a static point on the ground plane at the current frame and the previous frame are p_t^c and p^c_{t-1} respectively, they can be related by (9)[17], where R_c is the rotation matrix between the camera coordinates of the successive frames, R_w is the rotation matrix from the World coordinates to the camera coordinates, I is a 3x3 Identity matrix, and T_c is the translation matrix of the camera between successive frame respectively, n^T and h are the unit normal vector of the camera at the previous frame to the ground plane and the height of the camera to the ground respectively, A is the ground plane homography matrix between the previous frame and the current frame.

$$\mu = p_{2G} - p_1 \tag{8}$$

$$p_t^c = R_c\left(I + \frac{R_w T_c}{h} n^T\right) p_{t-1}^c = A p_{t-1}^c \tag{9}$$

The relationship between the screen coordinates for a static point on the ground plane can be related by (10), where p^s_t and p^s_{t-1} are the corresponding points of the static object on the screen coordinates in the current frame and the previous frame respectively, $\hat{p}^c_{t-1}$ and $\hat{p}^s_{t-1}$ are the normalized coordinates of p^c_{t-1} and p^s_{t-1} respectively, K is the intrinsic matrix of the camera, A is the ground plane homography matrix, and M *is* KAK^{-1}.

$$\begin{aligned} p_t^s &= KA\hat{p}^c_{t-1} = KAK^{-1}\hat{p}^s_{t-1} = Mp^s_{t-1} \\ p^s_{t-1} &= M^{-1}p^s_t \end{aligned} \tag{10}$$

Theoretically, MVs lying on the ground plane exhibit zero planar parallax residual. Also, for stationary objects above the ground plane, the corresponding planar parallax residual vectors should point towards the FOE.

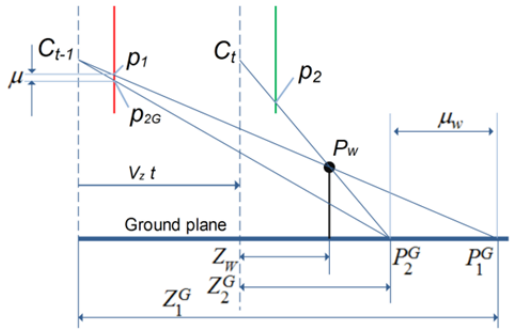


Fig. 4. Illustration of planar parallax residual

To detect a fast moving object, the planar parallax residual of each block of size 8×8 over the ROI is evaluated. If the amplitude of the resulting planar parallax residual is higher than a threshold q_m, and the direction is deviated from the FOE for more than q_r degrees, the block is regarded as a part of a moving object. If the total number of connected blocks satisfying the two constraints mentioned exceeds a certain threshold, a moving object is detected.

D. Road Region Identification

It is desirable to detect the road surface to reduce the region of interest for moving object detection. Since there is only one camera in the system, the detection of the road surface can also help estimate the distance of the moving object more accurately. A simplified flow-chart of the road region identification algorithm is shown in Fig. 5.

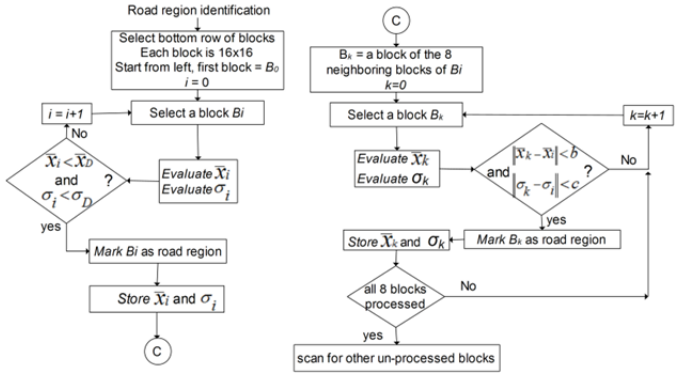


Fig. 5. Simplified flow-chart for road region identification

The captured image will first be converted into grayscale image. The road surface is identified by block based evaluation of the mean gray level and standard deviation of the grayscale image using the region grow algorithm.

Since the road surface usually has uniform color and weak texture, the mean gray level ($\bar{x}$) and standard deviation (σ) of one block of a road region should be close to those of its neighboring blocks. One block of size 16×16 along the row at the bottom of the captured image is used for initial evaluation of $\bar{x}$ and σ. This is because such region is the least likely to have any moving object when the ego vehicle is moving. Some blocks may however be affected by the markings on the road. If either one of $\bar{x}$ and σ exceeds the predefined threshold $\bar{x}_D$ and σ_D respectively, another block along the row will be chosen for the evaluation of $\bar{x}$ and σ again until the new $\bar{x}$ and σ of the block are within the predefined threshold.

When a block B_i with mean gray level $\bar{x}_i$ and standard deviation σ_i along the row of blocks near the bottom of the image is identified as the road region, the eight neighboring blocks surrounding block B_i are also evaluated for their respective mean $\bar{x}_k$ and standard deviation σ_k, where k denotes one of the eight blocks. A neighboring block B_k is identified as a road region block if it satisfies (11), where b and c are predefined thresholds..

$$\begin{cases} -b < (\bar{x}_k - \bar{x}_i) < b \\ -c < (\sigma_k - \sigma_i) < c \end{cases} \tag{11}$$

The region grow method is employed so that the road region can "grow" further by comparing the identified road region blocks with their neighboring blocks. Each comparison employs the $\bar{x}$ and σ of the centre block around the eight neighboring blocks until no new road region block can be identified.

E. Region of Interest Construction

The identified road region together with the amplitude of MVs are employed to construct the ROI for slow relative speed moving vehicle detection.

The upper half of the image that is above the *FOE* by L_y pixels is ignored. Fig. 6(a) shows the identified road region using the region growth algorithm mentioned in Section III-D. The road region is masked with white color.

The ROI is further reduced by excluding blocks with MVs of amplitude larger than a threshold q_m. Fig. 6(b) shows the image mask with white color indicating the blocks with MV larger than q_m set to *8*. Fig. 6(c) shows the combined image with white image mask on the cropped input grayscale image. Most of the area is masked, leaving only a small portion of the image for relative slow moving vehicle detection.

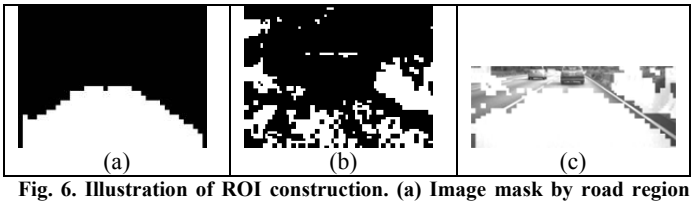


Fig. 6. Illustration of ROI construction. (a) Image mask by road region identification. (b) Image mask by filtering MVs with amplitude larger than a threshold. (c) Cropped image with mask in (a) and (b) combined.

F. U-Shape and Horizontal Contour Detection

After some steps for reducing the ROI, the horizontal contours and the U-shape-like contour are going to be detected. Successful detection indicates the presence of a relatively slow moving vehicle at the front.

During road region identification, the maximum $\bar{x}_{max}$ and minimum $\bar{x}_{min}$ gray levels of the road region are evaluated. Each pixel inside the ROI of the cropped input grayscale image is compared with $\bar{x}_{min}$ to create a binary image. If the pixel in the grayscale image is brighter than $\bar{x}_{min}$, the corresponding pixel in the binary image is set to '0'; otherwise it is set to '1'. The resultant binary image is shown in Fig. 7(a) where white color represents the area that is darker than $\bar{x}_{min}$.

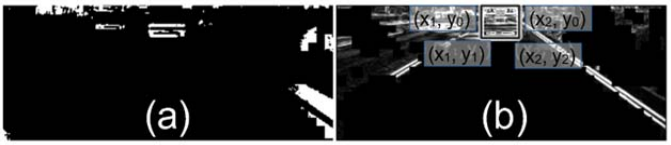


Fig. 7. (a) Binary image with those white areas representing regions that are darker than the minimum graylevel of the road region. (b) Vertical gradient image with a rectangle bracketing the area for vehicle detection.

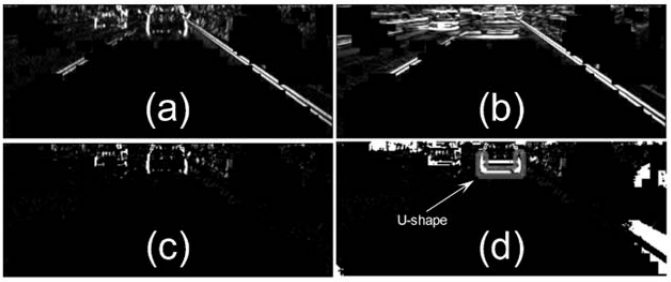


Fig. 8. Grayscale image with Sobel filtering and further processing thereafter. (a) Image after applying horizontal gradient Sobel kernel. (b) Image after applying vertical gradient Sobel kernel. (c) Image after comparing (a) and (b) for true vertical contours. (d) Vehicle U-shape marked for vehicle identification.

Besides the identified darkest area of the image as shown in Fig. 7, Sobel filter is also applied to the cropped grayscale input image to find the horizontal gradient and vertical gradient inside the ROI which are shown in Fig. 8(a) and (b) respectively. The result in Fig. 8(b) shows that there are many horizontal contours for vehicles on the road. Since the vehicle body is similar to a box shape when looking from the rear, there are also vertical contours being found after Sobel filtering using horizontal gradient kernel. Some contours, such as the lane marking, found by the horizontal gradient Sobel kernel are not truly vertical contours. They are further eliminated by comparing the pixel in the vertical gradient image (Fig. 8(b)) with the corresponding pixel in the horizontal gradient image (Fig. 8(a)), expressed as in (12), where P_{vv}, P_h and P_v are the resultant pixel value at screen coordinates *(x,y)*, pixel value in the horizontal gradient image and pixel value in the vertical gradient image respectively. D_{hv} is a predefined threshold for comparison. The resultant image is shown in Fig. 8(c). Fig. 8(d) is the combined result of the detected vertical contours in Fig. 8(c) and the detected darkest region in the image shown in Fig. 7. It shows clearly the position of a vehicle at the front with the U-shape illustrated in Fig. 8(d).

$$P_{vv} = \begin{cases} (P_h - P_v) \ \forall P_h \in [P_v, P_v + D_{hv}] \\ 0 \qquad \text{otherwise} \end{cases} \tag{12}$$

The algorithm detects the U-shape which is the result of horizontal and vertical contours of the vehicle. It starts by searching for the position of horizontal lines from the bottom of the binary image shown in Fig. 7(a). An earlier version of this process used the Hough transform to identify horizontal lines with inclination to the screen *x*-axis by not more than 3.5 degrees [18]. Modification is made so that Hough transform is not required thereby reduces the computation time. If a horizontal line *L* is detected, the width of the horizontal line is evaluated by converting the two endpoints, (x_1,y_1) and (x_2,y_2), of the line in the screen coordinates to the corresponding world coordinates, (X_{w1},Y_{w1}) and (X_{w2},Y_{w2}), using (3). If the resulting width ($W=X_{w2}-X_{w1}$) is longer than W_L or shorter than W_U, where W_L and W_U are predefined lower and upper limits of the width threshold, the line will be discarded.

If the line *L* falls within W_L and W_U, it is more likely that this line is coming from a vehicle. The line is further evaluated by examining the ratio between the width and the height. The height is obtained by the difference between the line location and the vanishing line location (y_0) in the screen coordinates. If the ratio R_{WH} is within a predefined range R_{WHL} and R_{WHU}, the bounding rectangle shown in Fig. 7(b) that is around the line location, will be further examined to confirm if there is a vehicle.

Further evaluation is performed to examine if there are two distinguished vertical contours near the two endpoints of line *L* , and if the average vertical gradient exceeds a predefined threshold [19].

The sum of gray levels *S(x)* is calculated by (13) at each horizontal position from *x+e to x-e*, where *e* is a predefined constant and *I(x,y)* is the gray level at screen coordinate *(x,y)*. Both the left side and the right side of the line *L* are evaluated.

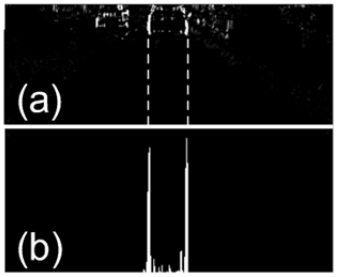


Fig. 9. Example vertical projection of the horizontal gradient image around the position where a vehicle may exist. (a) Horizontal gradient image where the dotted lines indicate the area under evaluation. (b) Corresponding vertical projection near the dotted lines in (a).

$$S(x) = \sum_{y=y_0}^{y_1} I(x,y) \ \forall x \in [x_1 - e, x_1 + e] \tag{13}$$

A plot of *S(x)* against the horizontal position is shown in Fig. 9(b). The maximum *S(x)* for the left and right sides of line *L* is found separately by comparing all *S(x)* in their corresponding side. If the maximum *S(x)* on both sides of the line exceeds a predefined threshold S_h, the existence of the vertical contours is confirmed.

The left and right sides of the bounding rectangle are replaced by the identified positions of the maximum *S(x),* denoted by x_1' and x_2' respectively. The existence of a vehicle is further confirmed by evaluating the average vertical gradient *V(x)* inside the new bounding rectangle using (14).

$$V(x) = \frac{\sum_{x=x_1'}^{x_2'} \sum_{y=y_0}^{y_1} I(x,y)}{\left[\left(x_2' - x_1' + 1\right)\left(y_1 - y_0 + 1\right)\right]} \tag{14}$$

If the value of V(x) exceeds a predefined threshold V_h, a vehicle at the rectangular position is confirmed as detected.

G. Distance and Speed Estimation

After confirming the position of a vehicle, its true moving speed is estimated by the MVs at the bottom line of the

rectangle. This step involves ego-motion compensation to find the true ground motion of the detected vehicle [20].

The bottom position of the vehicle in the screen coordinates of the current frame is $P_{t2}= ((x_1'+x_2')/2, y_1)$, the corresponding World coordinates W_{t2} can be found by using (3). Since the amplitude of MVs of relative slow moving vehicle is small, a small error in the MV will still result in an inaccurate estimation of the true ground motion of the detected vehicle. Therefore, the moving speed of the detected vehicle is calculated by binning multiple MVs along the bottom line of the rectangle bracketing the detected vehicle.

Fig. 10 shows a detected vehicle bracketed in a black rectangle. The image also shows the macroblock boundaries of size 8x8. The MVs of blocks highlighted in white dots in Fig. 10 along the bottom line of the rectangle are used to calculate the position on the world coordinates in the current frame and the previous frame.

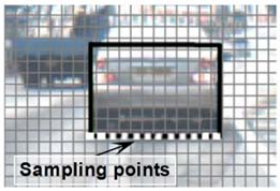


Fig. 10. Distance measurement based on binning of the MVs along the bottom line of the rectangle bracketing the detected vehicle. White dots in the image indicate the MV samples for true ground speed evaluation.

Eq. (3) is used to calculate the World coordinates of the block in the current frame and in the previous frame. If the screen coordinates of the block in the current frame are (x_s,y_s), its coordinates in the previous frame are (x_s+mv_x, y_s+mv_y), where (mv_x, mv_y) are the MV from the H.264/AVC encoder. Hence, the World coordinates in the current and previous frames are W_{t2} and W_{t1} respectively, calculated by (15).

$$W_{t2} = H_{sw}\begin{bmatrix} x_s & y_s & 1 \end{bmatrix}^T$$
$$W_{t1} = H_{sw}\begin{bmatrix} x_s + mv_x & y_s + mv_y & 1 \end{bmatrix}^T \quad (15)$$

Since the video frame rate and the ego vehicle speed are known, the speed u_i of each block i along the bottom white line can be calculated by (16), where v_s is the speed of the ego vehicle, Δt is the time interval between two successive frames, $W_{t1}(z)$ and $W_{t2}(z)$ are the speed components along the Z-axis of block i for the previous frame and current frame respectively.

$$u_i = \frac{W_{t2}(z) + v_s\Delta t - W_{t1}(z)}{\Delta t} \quad (16)$$

The deduced speed of each block along the bottom white line of the rectangle will vary because of the error in MVs. To determine the speed of the vehicle, the first step is to make sure all u_i obtained are of the same direction. Only the majority will be taken if some u_i are positive and some are negative. The average values of those u_i are evaluated to represent the moving speed of the detected vehicle.

H. Tracking of Detected Vehicle

After successful detection of the slow relative speed vehicle, the computation can be simplified by using a suitable tracking algorithm. Since the relative position of the detected vehicle is unlikely to significantly deviate from the last position, the tracking algorithm will first expand the size of the last bounding rectangle of the detected vehicle, then the bottom boundary of the rectangle is refined according to the difference in vertical gradients, the left and right sides of the rectangle are refined by detecting the maximum $S(x)$ on each side.

The tracking algorithm is activated when low relative speed vehicle is detected continuously for d_n consecutive frames.

IV. TEST RESULTS

Video sequences were prepared by capturing image and dynamic data synchronously. The video sequences were chosen to evaluate the algorithm with different interference during day-time on highways. The interferences include shadows, broken roads, road-side fences, symbols and texts on the road. The sequences are prepared from short sequences of tens of frames with simple scenarios to long sequences with hundreds of frames with more complicated scenarios, as listed in TABLE I. The image sequences were then encoded by JM18.4 H.264/AVC encoder. The encoder was configured so that the minimum block size inside each macroblock is 8x8. Since the block size of MVs in a frame will vary from 8x8 to 16x16, those MVs of block size larger 8x8 were regarded as multiple blocks of size 8x8 with the same MV value.

Different Quantization Parameters (QP) of values 17, 28 and 35 have been chosen to test the sequences at low, medium and high compression with a target frame rate of 6.5, 3.0 and 1.5kb/s respectively. The output MVs from the encoder with different QP values have similar patterns. The number of macroblocks using 16x8 and smaller partitions is higher for smaller QP, but the resulting ROI for moving object detection is similar, as illustrated in Fig. 11.

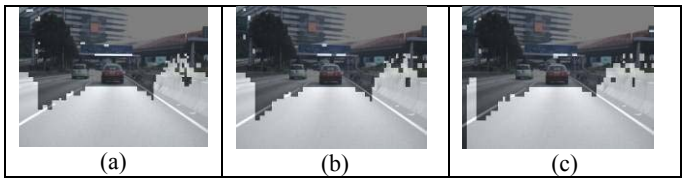


Fig. 11. Resulting ROI of a typical frame with QP set to (a) 17, (b) 28, and (c) 35. The ROI is essentially the same.

During the motion estimation stage, the MVs were exported with block size of 8x8 for each inter-frame encoding process. Sample sequences with preceding vehicles from Vaudrey et al. [21] were used to determine all thresholds except b, c, $\overline{x}_D$ and σ_D for road region detection which were determined by the test sequences presented in this paper. This is because the thresholds for road region are camera dependent, i.e. they are sensitive to the brightness and color response of the camera. For the actual system, these are included into the auto-calibration part of the camera immediately before it is being used in a car. The same set of thresholds is used for all the test video sequences where there are large variations in light intensity. Since these thresholds are mainly for road region detection and making decisions in the Sobel images, where the road region usually has small deviation in terms of color and uniformity, and Sobel images are less sensitive to light

intensity changes. The selected thresholds work well as observed from the test results.

A. Detection Rate

The algorithm was tested using the video sequences listed in TABLE I. The table also shows the challenges contained in these sequences. Fig. 12 illustrates the challenges. They include shadows from the environment, broken road with non-uniform color, text or symbol on the road, fences on road side, increasing or decreasing of distance to the front vehicle, and lane change due to the front vehicle or the ego vehicle.

TABLE I
Video sequences with different challenges to the proposed algorithm

Seq-uence	Shadow	Broken road	Road-side fence	Far to close	Close to far	Lane change	Symbol / Text
A				●			
B		●		●			
C	●					●	
D		●	●				
E		●		●	●		
F	●		●		●	●	
G	●	●		●			●

TABLE II
Detection results

Seq-uence	Total frame	True-positive			False-positive			Unsuccessful			Detection rate %		
		QP 17	QP 28	QP 35	QP 17	QP 28	QP 35	QP 17	QP 28	QP 35	QP 17	QP 28	QP 35
A	42	42	42	42	0	0	0	0	0	0	100	100	100
B	100	97	97	97	0	0	0	3	3	3	97.0	97.0	97.0
C	246	238	240	240	1	1	1	8	6	6	96.8	97.6	97.6
D	275	262	261	262	0	0	0	13	14	13	95.3	94.9	95.3
E	440	397	398	430	1	1	1	40	40	14	90.2	90.5	97.7
F	208	189	190	189	0	0	0	20	19	20	90.4	90.9	90.4
G	342	331	332	331	0	0	0	11	10	11	96.8	97.1	96.8
A to G	1653	1558	1561	1586	2	2	2	95	92	67	94.3	94.4	95.9

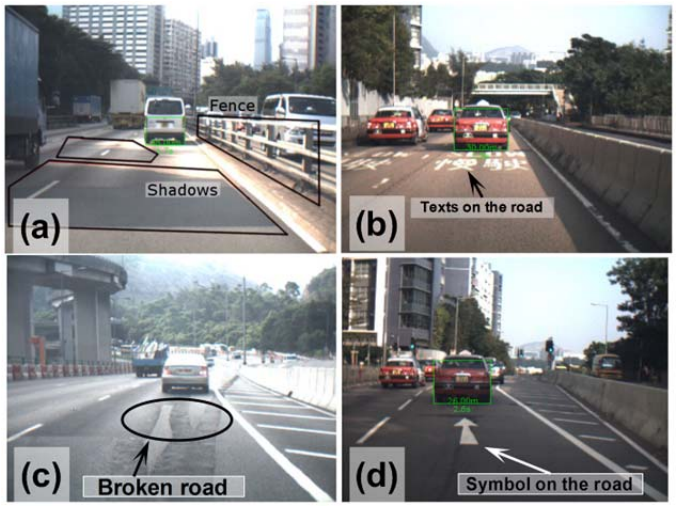


Fig. 12. Challenges appeared in the test video sequences. (a) Road-side fence and shadows. (b) Texts on the road. (c) Broken road. (d) Symbol on the road.

The result is categorized into true-positive, false-positive and un-successful detection. The detection results with different QP values were summarized in TABLE II. The last row shows the combined result of Seq. A to G. The overall result shows that the true-positive detection rate can reach more than 90%, while the false-positive rate remains at very low level of less than 0.2%. Comparing with results from other approaches with different test conditions giving true positive rate ranging from 90 to 95% [6], [7], [9], the performance of the proposed algorithm is on a par with other approaches. The algorithm works well with the same set of fixed parameters to all the test sequences with large difference in luminance conditions and high-contrast scenarios during the day-time.

The result also shows that the algorithm is robust to different compression quality. It is worth to note that both the false-positive and un-successful detection rate were decreased for Seq. E using higher compression QP. This is because the higher compression ratio can lead to a much more texture-less surface on the rear of the vehicle for motion estimation. This results in more MVs with small amplitude on the vehicle. Hence, more useful blocks are retained for use by the algorithm for better detection..

However, there are some cases of false detection, mainly due to the shape of the fence and barrier on the side of the road, as shown in Fig. 13. The wrong detection is because of the similar shape due to the shadow (Fig. 13(a)), horizontal and vertical contours of the detected area (Fig. 13(b)). Also, the MVs at that regions are small, the ROI reduction technique based on the inclusion of MVs with large amplitude does not work effectively, leading to the detection algorithm treating them as valid ROI for vehicle detection.

Fig. 13. False detection of vehicles. Both (a) and (b) detect wrongly because part of the shape of the road-side fence forms the horizontal contours and vertical contours similar to that of a preceding vehicle.

B. Computation Time Analysis

Since the proposed algorithm is targeted to be run in real-time for practical application in consumer products, the computation load of the algorithm is also analyzed.

The algorithm was developed in C++ language and was tested with a PC with x86 processor running at 2.6GHz clock speed. The captured image sequence of size 640x480 each was stored to the PC for offline processing. A JM18.4 video encoder was used for offline encoding the video to H.264 format. The video encoder was modified to output MV map for each p-frame. The MV map was also stored in the PC. Furthermore, the encoder was set to IBPBP frame structure, with frame rate 30fps, using EPZS motion estimation algorithm, with intra-frame encoding in p-frame and macroblock partition smaller than 8×8 disabled.

The program read the captured image sequence, and the corresponding dynamic data and MV file from the harddisk of the PC. The main functions for relatively low speed vehicle detection include finding the ROI and the detection of the vehicle. After successful detection, the tracking function was used without running the detection algorithm.

The average time for processing the seven sequences for vehicle detection is shown in TABLE III. It was found that the processing time for finding the ROI and for vehicle tracking was relatively low. However, the average processing time for relatively slow vehicle detection varied from 15.7ms to 192.9ms. The large deviation in the required detection time was due to the existence of multiple regions with U-shape features. For instance, Fig. 14(a) shows the binary image of frame 66 of Sequence F with white areas representing regions that are darker than the minimum gray level of the detected road region. Fig. 14(b) shows arrows pointing to the areas that require to run vehicle detection algorithm. Since the area is relatively large when comparing with the case with only one vehicle at the front, extra time is spent on the vehicle detection function.

With the H.264/AVC encoder set to IBPBP frame structure and 30fps, the interval between p-frame for ROI detection is 1/15 second, i.e. 66.7ms. If the cycle time for vehicle detection can be completed within 66.7ms, the detection cycle is fast enough to catch up with the designed frame rate.

Ignoring the file input/output (I/O) time (IV in TABLE III) that can be eliminated in the future real-time system where the image and MV are read directly from the memory, the tracking cycle time (I+III) is less than 24ms. It is fast enough to match with the desired video frame rate.

TABLE III
Average processing time in ms for low relative speed vehicle detection

Seq-uence	I	II	III	IV	Cycle Time		
	Finding ROI	Detection	Tracking	File I/O	I+II+IV	I+II	I+III
A	5.6	15.7	4.3	126	147.3	21.3	9.9
B	10.5	16.5	4.7	76	103	27.0	15.2
C	7.2	51.5	4.5	67	125.7	58.7	11.7
D	11.6	31.8	9.9	89	132.4	43.4	21.5
E	12.7	75.2	7.3	89	176.9	87.9	20
F	11.1	192.9	5.7	83	287	204.0	16.8
G	17.2	52.7	6.3	102	171.9	69.9	23.5

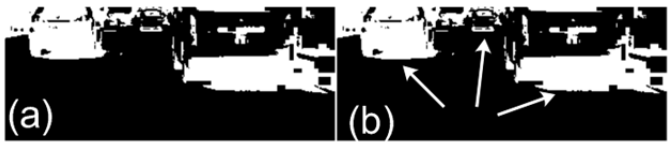


Fig. 14. (a) Binary image with white color representing the area that is darker than the minimum graylevel of the identified road region. (b) Arrows pointing to the multiple regions for potential vehicle detection.

However, the cycle time for vehicle detection (I+II) can vary from tens of milliseconds to hundreds of milliseconds. Even if the detection takes hundreds of milliseconds to complete, the detection algorithm is still effective for real-time application. This is because the expected movement of low relative speed vehicle across several frames is small, the ROI is still valid across several frames for detection even if the detection algorithm is skipped for a few frames.

C. Fast Moving Object Detection

The result of fast moving object detection using planar parallax method is briefly described in this section for the completeness of the paper.

Fig. 15(a) shows one of the captured images in an image sequence from Vaudrey et al. [21]. Fig. 15(b) shows the planar parallax residual vectors (PPRVs) with amplitude larger than eight pixels after ego-motion compensation. The grayscale level of the drawn lines indicates the angle of the vector to the FOE. The FOE in the image is indicated by a cross near the center of the image. The top left of the image shows the grayscale scheme for different angles to the FOE. The vectors in grayscale overlaid to the forthcoming vehicle indicate that fast relative motion between the forthcoming vehicle and the ego vehicle is detected, and that the movement direction is emerging from the FOE. Therefore, the movement of the forthcoming vehicle is in parallel to the ego vehicle. The vectors on the preceding vehicle near the FOE are of small amplitude and with multiple directions. This is because the preceding vehicle has low relative speed to the ego vehicle, resulting in small MVs. These small MVs give erroneous PPRV amplitude and direction after ego-motion compensation. Because of the inaccuracy of MVs near the FOE, the preceding vehicle is detected by the low relative speed vehicle detection algorithm described in the paper. Other outliers can be removed by identifying the road region, and the number of connected macroblocks with identified PPRVs.

Fig. 15. Moving object detection using planar parallax residual. (a) Original image. (b) Vectors with line length indicating the amplitude of the residual vectors, and the grayscale level indicating the angle to the FOE.

V. CONCLUSION

A novel algorithm that uses the MVs from H.264/AVC encoder for identifying the region of interest and the use of generic line features for slow relative speed vehicle detection and tracking is proposed. It is suitable to work with MV based moving object detection for Advanced Driver Assistance System. The detection algorithm relies only on the generic horizontal contours and vertical edges that exist in most small to large size vehicles. The test results show that the true-positive detection rate is high and the computation time is short. This makes the algorithm promising for use in real-time embedded system.

This proposed slow relative speed vehicle detection algorithm is combined with the algorithm for relatively fast moving object detection. Currently a simple criteria that makes use of the MV amplitude threshold q_m is used to switch between the algorithms for relatively fast and slow vehicle detection. The tracking algorithm for relatively fast moving objects is yet to be developed to reduce the computation requirement after successful detection. An effective lane detection algorithm should also be used to form a complete solution for vision based advanced driver assistance system.

The algorithm also forms the basis of assistive safety enhancement in inter-vehicular communication networks. Results from the lane detection algorithm can also be used to further reduce the ROI for relatively low speed vehicle detection so that the required vehicle detection time can further be reduced.

REFERENCES

[1] H.-K. Cheung, W.-C. Siu, S. Lee, L. Poon, and C.-S. Ng, "Accurate distance estimation using camera orientation compensation technique for vehicle driver assistance system," in *IEEE Intl. Conf. on Consumer Electron.*, pp. 227-228, 2012.

[2] Y. C. Kuo, N. S. Pai, and Y. F. Li, "Vision-based vehicle detection for a driver assistance system," *Comput. & Math. with Applications,* vol. 61, pp. 2096-2100, 2011.

[3] A. Broggi, P. Cerri, and P. C. Antonello, "Multi-resolution vehicle detection using artificial vision," in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, pp. 310-314, 2004.

[4] T. Liu, N. Zheng, L. Zhao, and H. Cheng, "Learning based symmetric features selection for vehicle detection," in *Proc. IEEE Intell. Veh. Symp.*, pp. 124-129, 2005.

[5] C. Tzomakas and W. v. Seelen, "Vehicle Detection in Traffic Scenes Using Shadows," *IR-INI, INSTITUT FUR NUEROINFORMATIK, RUHR-UNIVERSITAT.,* 1998.

[6] A. Haselhoff and A. Kummert, "A vehicle detection system based on Haar and Triangle features," in *Proc. IEEE Intell. Veh. Symp.*, pp. 261-266, 2009.

[7] X. Yong, L. Zhang, Z. Song, Y. Hu, L. Zheng, and J. Zhang, "Real-time vehicle detection based on Haar features and Pairwise Geometrical Histograms," in *IEEE Intl. Conf. on Information and Automation*, pp. 390-395, 2011.

[8] L. Mao, M. Xie, Y. Huang, and Y. Zhang, "Preceding vehicle detection using Histograms of Oriented Gradients," in *IEEE Intl. Conf. on Comm., Circuits and Syst.*, pp. 354-358, 2010.

[9] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-Based Vehicle Detection System With Consideration of the Detecting Location," *IEEE Trans. on Intell. Transp. Syst.,* vol. 13, pp. 1243-1252, 2012.

[10] Y. Freund and R. E. Schapire, "A desicion-theoretic generalization of on-line learning and an application to boosting," in *Computational learning theory*, pp. 23-37, 1995.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning,* vol. 20, pp. 273-297, 1995.

[12] A. Giachetti, M. Campani, and V. Torre, "The use of optical flow for road navigation," *IEEE Trans. on Robotics and Automation,* vol. 14, pp. 34-48, 1998.

[13] J. Klappstein, F. Stein, and U. Franke, "Monocular Motion Detection Using Spatial Constraints in a Unified Manner," in *Proc. IEEE Intell. Veh. Symp.*, pp. 261-267, 2006.

[14] T.-J. Kim, J.-E. Hong, and J.-W. Suh, "A fast intra mode skip decision algorithm based on adaptive motion vector map," *IEEE Trans. Consumer Electron.,* vol. 55, pp. 179-184, 2009.

[15] F. Zhang, Y. Gao, and J. D. Bakos, "Lucas-Kanade Optical Flow estimation on the TI C66x digital signal processor," *IEEE High Performance Extreme Computing Conference*, pp. 1-6, 2014.

[16] J.-Y. Chang, W.-F. Hu, M.-H. Cheng, and B.-S. Chang, "Digital image translational and rotational motion stabilization using optical flow technique," *IEEE Trans. Consumer Electron.,* vol. 48, pp. 108-115, 2002.

[17] K.-L. Chung, W.-J. Yang, W.-M. Yan, and C.-S. Fuh, "New joint demosaicing and arbitrary-ratio resizing algorithm for color filter array based on DCT approach," *IEEE Trans. Consumer Electron.,* vol. 56, pp. 783-791, 2010.

[18] C. C. Wong, W. C. Siu, P. Jennings, and S. Barnes, "Low Relative Speed Moving Vehicle Detection Using Motion Vectors and Generic Line Features," in *IEEE Intl. Conf. on Consumer Electron.*, Las Vegas, USA, pp. 208-209, Jan. 2015.

[19] B. C. M. Fong and W. C. Siu, "Detection system for assisting a driver when driving a vehicle using a plurality of image capturing devices," US 8,405,491, Mar. 26, 2013, 2013.

[20] D.-U. Jung, Y.-I. Yun, and J.-S. Choi, "A new interface using image-based foot tracking for motion sensing devices," *IEEE Trans. Consumer Electron.,* vol. 58, pp. 803-809, 2012.

[21] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn, "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," in *IEEE Intl. Conf. on Image and Vis. Comput.*, New Zealand, pp. 1-6, 2008.

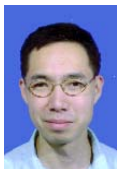
BIOGRAPHIES

Chup-Chung Wong (M'08) received the B.Eng.(Hons) degree in Electrical and Electronic Engineering in 1991 and M.Sc. degree (with distinction) in Automotive Engineering and Design in 2008 from the University of Hong Kong and Hong Kong Polytechnic University respectively. He is currently an Eng.D. student in WMG, University of Warwick. He is a Senior Consultant of the Hong Kong Productivity Council. His research interests include image processing and video coding for real-time applications.

Wan-Chi Siu (S'77-M'77-SM'90-F'12) received the PhD degree from Imperial College, London, in 1984, and is Fellow of the IEEE. He is currently a Chair Professor with the Hong Kong Polytechnic University with main research interest in DSP, fast algorithms, video coding, 3D videos, pattern recognition and surveillance. He has published 480 research papers, and has 8 recent patents. He served as the Vice President of the IEEE Signal Processing Society (2012–2014), and he has recently been elected as the President-Elect (2015-2016) of the APSIPA (Asia-Pacific Signal and Information Processing Association). He has been/was guest editor and associate editor of IEEE Transactions on IP, CAS, CSVT and in additional to other journals.

Paul Jennings received a BA degree in physics from the University of Oxford in 1985 and an Engineering Doctorate from the University of Warwick in 1996.Since 1988 he has worked on industry-focused research for WMG at the University of Warwick. His current interests include: vehicle electrification, in particular energy management and storage; autonomous vehicles; and user engagement in product and environment design, with a particular focus on automotive and healthcare applications.

Stuart Barnes studied in the Department of Metallurgy and Materials, University of Birmingham (UK) and graduated with a PhD in 1988 after spending his earlier career in manufacturing. He is a Chartered Engineer and a Fellow of the IoM3. Dr Barnes then worked for GKN Technology before joining WMG, University of Warwick in 1992 where he has researched various aspects of machining and laser processing and is currently an Associate Professor and Director of Research Degrees.



Bernard Fong (M'93-SM'03) received his BS degree in electronics from UMIST (UK) and PhD degree in medical technology from UNSW (Australia) in 1993 and 2005, respectively. He currently serves as an associate editor for the IEEE Transactions on Consumer Electronics, IEEE Consumer Electronics Magazine, and the Journal of Advances in Information Technology. He has served as a guest editor for the IEEE Communications Magazine and the IEEE Transactions on Affective Computing.