

**A Thesis Submitted for the Degree of PhD at the University of Warwick**

**Permanent WRAP URL:**

<http://wrap.warwick.ac.uk/88420>

**Copyright and reuse:**

This thesis is made available online and is protected by original copyright.

Please scroll down to view the document itself.

Please refer to the repository record for this item for information to help you to cite it.

Our policy information is available from the repository home page.

For more information, please contact the WRAP Team at: [wrap@warwick.ac.uk](mailto:wrap@warwick.ac.uk)



Moving Object Detection for Automobiles by the  
Shared Use of H.264/AVC Motion Vectors

Innovation Report

Wong Chup Chung (Elvin)

30<sup>th</sup> November 2015

## **Abstract**

Cost is one of the problems for wider adoption of Advanced Driver Assistance Systems (ADAS) in China. The objective of this research project is to develop a low-cost ADAS by the shared use of motion vectors (MVs) from a H.264/AVC video encoder that was originally designed for video recording only. There were few studies on the use of MVs from video encoders on a moving platform for moving object detection. The main contribution of this research is the novel algorithm proposed to address the problems of moving object detection when MVs from a H.264/AVC encoder are used. It is suitable for mass-produced in-vehicle devices as it combines with MV based moving object detection in order to reduce the cost and complexity of the system, and provides the recording function by default without extra cost. The estimated cost of the proposed system is 50% lower than that making use of the optical flow approach.

To reduce the area of region of interest and to account for the real-time computation requirement, a new block based region growth algorithm is used for the road region detection. To account for the small amplitude and limited precision of H.264/AVC MVs on relatively slow moving objects, the detection task separates the region of interest into relatively fast and relatively slow speed regions by examining the amplitude of MVs, the position of focus of expansion and the result of road region detection.

Relatively slow moving objects are detected and tracked by the use of generic horizontal and vertical contours of rear-view vehicles. This method has addressed the problem of H.264/AVC encoders that possess limited precision and erroneous motion vectors for relatively slow moving objects and regions near the focus of expansion.

Relatively fast moving objects are detected by a two-stage approach. It includes a Hypothesis Generation (HG) and a Hypothesis Verification (HV) stage. This approach addresses the problem that the H.264/AVC MVs are generated for coding efficiency rather than for minimising motion error of objects. The HG stage will report a potential moving object based on clustering the planar parallax residuals satisfying the constraints set out in the algorithm. The HV will verify the existence of the moving object based on the temporal consistency of its displacement in successive frames.

The test results show that the vehicle detection rate higher than 90% which is on a par to methods proposed by other authors, and the computation cost is low enough to achieve the real-time performance requirement.

An invention patent, one international journal paper and two international conference papers have been either published or accepted, showing the originality of the work in this project. One international journal paper is also under preparation.

## **Acknowledgements**

This piece of work cannot be completed without the support and guidance of all the people who have helped me during the past five years.

There are so many people that I would like to say thanks. My wife Athena and my son James are always confident and forgiving to me. I don't know how much time they have sacrificed in allowing me to work on this project. They didn't express their mental pressure when having family time with me, worrying if they would have occupied too much of my time. Thanks to my mother, father, mother-in-law and father-in-law. You always love me and are always adorable.

My academic mentors, Prof. Wan-Chi Siu, Dr. Stuart Barnes, and Prof. Paul Jennings, are all giving valuable advice to my research direction, contents, academic vigour, schedule, as well as English grammar. In addition, Dr. Tina Barnes, Dr. Kevin Neailey, Prof. H. C. Man and Dr. Francis Lau, are very helpful, enabling me to seek advice from them from time to time.

My Industrial mentors, Mr. Frank Leung, Dr. Lawrence Cheung and Mr. Raymond Chiu, are all very supportive to my research. They can always give valuable advice from the industry's and customer's perspective. Although Mr. Frank Leung and Mr. Raymond Chiu had left the organisation that I am working, I would not forget their generosity towards supporting my personal and career development.

My colleagues, Dr. Bernard Fong, Dr. Lawrence Poon, Mr. Jerry Ng, Mr. Shawn Chan, Dr. Kwok-Wai Hung are all giving hands to help solve engineering problems that encountered during the development of the system. Especially Dr. Bernard Fong who helped review and comment comprehensively on my papers submitted for peer review.

Finally, the supporting staff from both WMG and Hong Kong Polytechnic University, and some ex-EngD graduates who attended EngD Workshops in Hong Kong, is all very supportive and truly eager to share their valuable experience.

This project is supported by the Innovation and Technology Commissions of Hong Kong via the R&D project "New Generation Advanced Driver Assistance System (ITP/015/13AI)" and "Development of Advanced Collision Avoidance System (ITT/006/12AP)" of the Hong Kong Automotive Parts and Accessory Systems R&D Centre. Without the funding support, this piece of research cannot be started.

<b>Table of Content</b>		<b>Page</b>
1	Introduction.....	1
1.1	Research Objective .....	1
1.2	Advanced Driver Assistance System .....	1
1.3	ADAS Products in the Market .....	5
1.4	Industrial Requirements and Constraints .....	8
1.5	Statement of Originality .....	9
1.6	Organisation of Chapters .....	10
1.7	Chapter Summary .....	11
2	Literature Review.....	12
2.1	Lane Detection .....	12
2.2	Vehicle Detection.....	15
2.3	Generic Moving Object Detection .....	17
2.3.1	Planar Parallax Method.....	19
2.3.2	Optical Flow versus H.264/AVC Motion Vector .....	20
2.3.3	Ego Motion Estimation .....	23
2.4	H.264/AVC Motion Vector Overview.....	26
2.4.1	Bitrate Comparison .....	27
2.4.2	Evaluation of Motion Vector .....	29
2.4.3	MVs Near FOE and on Relatively Slow Moving Objects.....	35
2.5	Road Region Detection .....	37
2.6	Practical Implications.....	39
2.7	Research Focus .....	41
2.8	Chapter Summary .....	44
3	Algorithm Framework .....	45
3.1	System Preparation .....	47
3.1.1	Configuration for Video Encoder .....	47
3.1.2	Camera Calibration .....	48
3.2	Ego Motion Estimation .....	61
3.2.1	Planar Homography Estimation .....	62
3.2.2	Ego Motion Compensation .....	64
3.2.3	Focus of Expansion Estimation .....	66
3.3	Road Region Detection .....	71
3.3.1	Building Road Colour Model.....	73
3.3.2	Seed Block for Road Region Grow .....	73
3.3.3	Road Region Grow .....	74
3.3.4	Post Road Region Grow Refinement.....	75
3.4	Segmentation of Regions of Interest.....	76

3.4.1	ROI for Slow Relative Speed Objects .....	77
3.4.2	ROI for Fast Relative Speed Objects .....	79
3.5	Slow Relative Speed Moving Object Detection .....	80
3.5.1	Slow Relative Speed Vehicle Detection Method.....	81
3.5.2	Binary Image Creation.....	81
3.5.3	Vehicle Detection.....	82
3.5.4	Vehicle Tracking.....	86
3.5.5	Distance and Speed Estimation.....	89
3.6	Fast Relative Speed Moving Object Detection .....	92
3.6.1	Planar Parallax Residual .....	94
3.6.2	Hypothesis Generation.....	96
3.6.3	Hypothesis Verification .....	103
3.6.4	Tracking .....	104
3.7	Chapter Summary .....	105
4	Test and Evaluation.....	106
4.1	Evaluation of Camera Calibration Results.....	106
4.1.1	Focal Lengths and Principal Point Estimation.....	106
4.1.2	Intrinsic Parameter .....	108
4.1.3	Camera Installation .....	108
4.1.4	Extrinsic Parameter Estimation.....	109
4.1.5	Distance Accuracy Evaluation .....	110
4.1.6	Comparison to Zhang's Method .....	111
4.2	Verification of Ego Motion Compensation.....	114
4.3	Slow Relative Speed Vehicle Detection and Tracking Method.....	117
4.3.1	Region of Interest Formation .....	117
4.3.2	Detection and Tracking.....	120
4.3.3	Detection Rate.....	124
4.3.4	Computation Time Analysis .....	131
4.3.5	Comparison of Results .....	135
4.4	Fast Relative Speed Vehicle Detection and Tracking Method .....	137
4.4.1	Region of Interest Formation .....	137
4.4.2	Setup of Experiments .....	139
4.4.3	Detection Results .....	143
4.4.4	Exception of Detections .....	149
4.5	Computation Time Analysis .....	153
4.6	Cost Analysis .....	155
4.7	Chapter Summary .....	158
5	Commercialisation .....	159
5.1	Chapter Summary .....	163

6	Publications and Patents .....	164
6.1	Chapter Summary .....	166
7	Future Improvements .....	167
7.1	Camera Calibration Method.....	167
7.1.1	Road Gradient Compensation .....	167
7.2	Road Region Detection .....	169
7.2.1	Use of Temporal Information .....	169
7.3	Segmentation Method for Relatively Slow and Fast Objects .....	170
7.4	Slow Relative Speed Moving Object Detection Algorithm.....	171
7.5	Fast Relative Speed Moving Object Detection Algorithm .....	172
7.5.1	Predictive Displacement Estimation .....	172
7.5.2	The Use of Bi-Directional Motion Vector .....	172
7.5.3	The Algorithm for Template Matching.....	173
7.5.4	Reducing False-Positive Detection .....	175
7.6	The Use of Stereo Camera .....	176
7.7	Improving H.264/AVC Motion Estimation .....	177
7.8	Extension to Next Generation Video Encoder.....	178
7.8.1	Comparison to H.264/AVC Encoding Standard .....	179
7.8.2	Shared-Use of Motion Vector from HEVC Encoder.....	181
7.9	Chapter Summary .....	182
8	Future R&D Directions.....	183
9	Further Works on Product Commercialisation .....	184
10	Conclusions.....	186
11	References.....	190

<b>List of Figures</b>	<b>Page</b>
Figure 1-1: Forward Collision Warning System (FCWS). (a) Forward looking camera installed in a vehicle, monitoring the relative location of the front vehicle. (b) Warning signal is issued to alert the driver when the distance between the vehicles is too close. ....	2
Figure 1-2: Illustration of LDWS. (1) A forward looking camera monitoring the driving lanes. (2) Capture image from the forward looking camera. (3) The instance with warning signal output to notify the driver when the vehicle is cutting the driving lane. (4) Vehicle driving back to the designated lanes after the driver has corrected the driving path. (source: <a href="http://fr.wikipedia.org/wiki/Fichier:Lane_Departure_Warnin_g.jpg">http://fr.wikipedia.org/wiki/Fichier:Lane_Departure_Warnin_g.jpg</a> ) .....	3
Figure 1-3: A pedestrian at 60m away from the vehicle marked with a red triangle. ....	4
Figure 1-4: Blind Spot Detection and Warning System (BSDWS). Cameras installed at the wing mirrors looking backward, covering the areas that are not easily seen by the driver. (Source: <a href="http://images.businessweek.com/ss/06/09/cartech/source/2.htm">http://images.businessweek.com/ss/06/09/cartech/source/2.htm</a> ) .....	5
Figure 2-1: (a) Normal captured image. (b) The bird's eye view image transformed from (a) using IPM technique.....	14
Figure 2-2: Planar parallax diagram .....	20
Figure 2-3: Block size for motion estimation. Primary macro-block size is mode 1 at 16x16. Smaller partitions from 16x8 down to 4x4 are possible according to the decision of the motion estimation algorithm.....	26
Figure 2-4: An overview of H.264 encoded snapshot of 2 video sequences. Each video sequence is encoded with intra-mode on and off, using three different motion search algorithms. The red and green boxes shown in each encoded snapshot are macroblocks encoded in intra-mode and inter-mode respectively. The green lines shown at each macroblock indicates the amplitude and direction of the motion vector. ....	31
Figure 2-5: An overview of H.264 encoded snapshot of another 2 video sequences. Each video sequence is encoded with intra-mode on and off, using three different motion search algorithms. The red and green boxes shown in each encoded snapshot are macroblocks encoded in intra-mode and inter-mode respectively. The green lines shown at each macroblock indicates the amplitude and direction of the motion vector.....	32
Figure 2-6: Macroblocks and encoding mode for different Daimler video sequences. Sequence Construction site shows small number of macroblocks to represent the movement of front vehicles. Sequence Crazy Turn Left shows MVs due to the left turn action of the subject vehicle. ....	33
Figure 2-7: Macroblocks and encoding mode for different Daimler video sequences. Dancing Light and Intern On Bike sequences show small	



number of macroblocks to represent the movement of the front vehicles. ....	34
Figure 2-8: Image showing two overlaid consecutive images. Red lines show the optical flow field found by generic KLT feature point tracking algorithm. Green lines show the virtual optical flow field emerging from a point known as the FOE. ....	36
Figure 2-9: Selected MVs near the FOE on the road and on the slow relative speed moving vehicle. The amplitude of MVs is either identical or has small difference, making it difficult to distinguish moving objects and static regions. ....	37
Figure 2-10: Example of Aftermarket Car Cameras. (a) and (b) Built-in Infrared LEDs for night time illumination. (c) Inclined angle to fit more tightly to the windshield. (d) Movable lens for easier camera adjustment (source: <a href="http://www.hktdc.com">http://www.hktdc.com</a> ). ....	40
Figure 2-11: ADAS and H.264/AVC video encoding 2-in-1 system with shared functional blocks. The block diagram shows the motion estimation and motion vector functional blocks are shared for the use of video coding and moving object detection. ....	43
Figure 3-1: Major functional blocks of the proposed algorithm framework .....	46
Figure 3-2: Transforming MV for different block size to represent the same block size of 8x8 .....	47
Figure 3-3: Illustration of World coordinates, camera coordinates, and screen coordinates. There are Rotation $R_w$ and Translation $T_w$ from the World coordinates to the camera coordinates. The screen coordinates start from the top left corner of an image. ....	49
Figure 3-4: Camera with non-zero pitch angle $\theta_x$ . The vanishing point on the screen is not aligned with the optical axis. ....	50
Figure 3-5: Calibration Setup for focal lengths of the intrinsic parameter of the camera .....	54
Figure 3-6: Simple pin-hole camera model. (a) Definition of camera coordinates system and the screen coordinates system. (b) Viewing from X-axis to the origin with the green line representing the image plane .....	55
Figure 3-7: Checker pattern on the up-right board. (a) Physical dimension of the checker pattern. (b) Coordinates and distance between control points in number of pixels. ....	56
Figure 3-8: Illustration of good symmetric positions for symmetric geometrical line construction for camera calibration. The square boxes shown on the left and right side of the vehicle indicate example locations for good symmetrical geometrical line construction. ....	57
Figure 3-9: Illustration of alignment markings for the centre line for camera installation. The banner with checker pattern is not aligned to the centre line until the centre line is found. ....	57
Figure 3-10: The camera installation process with the check-board banner aligned to the centre line LC of the vehicle .....	58

Figure 3-11: An example video display with the scene captured by the camera to be installed. It sees the banner with checker pattern on the level ground. An orange line is overlaid in the centre of the screen to indicate the line with zero yaw angle. A red horizontal line is also overlaid at the bottom of the screen to indicate the zero roll angle. The orange and red line should be aligned vertically and horizontally respectively to a line formed by the checker pattern on the banner.....	59
Figure 3-12: Camera height calibration using a banner with checker pattern placing on the level ground.....	61
Figure 3-13: Display of MVs due to ego motion only. The MVs are displayed at 16x16 pixel interval with vehicle speed and camera parameters from frame number 10 of the “Intern on bike” sequence of the Daimler sequence.....	65
Figure 3-14: Display of MVs due to ego motion only. The MVs are displayed at 16x16 pixel interval with vehicle speed and camera parameters from frame number 202 of the “Crazy Turn” sequence of the Daimler sequence.....	65
Figure 3-15: Illustration of a vehicle on an inclined road, the pitch angle measured by the inertial sensor is the angle between the road plane and the earth plane rather than the angle between the camera optical axis and the road plane.....	68
Figure 3-16: Simple vehicle motion model in bird’s eye-view. The vehicle travels at speed $v$ , and turning at angular rate $\omega$ . The time interval between successive frames is $\delta t$ .....	71
Figure 3-17: Relatively large MVs highlighted in red circles are from static objects near the ego vehicle. ....	72
Figure 3-18: Captured image showing 16x16 grid lines in green colour. The seed block is searched from the bottom left to the bottom right of the image until a block is found with satisfactory mean and standard deviation. An example block highlighted in orange colour is compared to its neighbour blocks marked with number 1 to 8 with purple colour. ....	74
Figure 3-19: Road region blocks are highlighted in white. The minimum and maximum road region block at each row $y_b$ are $X_{min}(y_b)$ and $X_{max}(y_b)$ respectively. The minimum road region block at each column $x_b$ is $Y_{min}(x_b)$ .....	76
Figure 3-20: The road region detection result after the hole-filling refinement process.....	76
Figure 3-21: (a) Typical captured image that has been converted to grayscale image. (b) The FOE and the primary ROI is selected as the area below the FOE. ....	77
Figure 3-22: 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 that combines the image mask (a) and (b). ....	78

Figure 3-23: (a) Regions highlighted in red circles are areas that should be ignored. (b) ROI after refinement. ....	79
Figure 3-24: Illustration of ROI construction for fast relative speed vehicle detection. (a) Image mask by road region identification. (b) Image mask by filtering MVs with amplitude larger than a threshold. (c) Cropped image combining image mask (a) and (b). ....	80
Figure 3-25: Functional block diagram of the slow relative speed vehicle detection algorithm .....	81
Figure 3-26: (a) Binary image with those white areas representing regions that are darker than the minimum graylevel of the road region. (b) Binary image in (a) overlaid to the ROI of the captured image. Those horizontal contours along the rear part of vehicles at the front are identified. ....	82
Figure 3-27: (a) Sobel kernel for finding horizontal gradients. (b) Sobel kernel for finding vertical gradients.....	82
Figure 3-28: Grayscale image with Sobel filtering. (a) Resultant image after applying horizontal gradient Sobel kernel. (b) Resultant image after applying vertical gradient Sobel kernel. ....	83
Figure 3-29: Resultant image of true vertical contour image, after using Equation (3.49). ....	83
Figure 3-30: (a) Combined result of the true detected vertical contours (shown in white colour) and the detected darkest region in the image (shown in red colour). (b) U-shape bracket drawn in green colour, indicating the found U-shape due to the vehicle.....	84
Figure 3-31: Illustration of the area shown in green rectangle identified for examination of whether a vehicle exists. ....	85
Figure 3-32: Illustration of the vertical projection of the horizontal gradient image around the rectangular position where a vehicle potentially exists. (a) Horizontal gradient image and the rectangular area under evaluation. (b) Corresponding vertical projection near the red rectangular area. ....	86
Figure 3-33: Conceptual flow-chart of the tracking algorithm for slow relatively speed moving vehicles .....	87
Figure 3-34: Speed measurement based on binning of the MVs along the bottom line of the rectangle bracketing the detecting vehicle. Red dots in the image indicate the MV samples for true ground speed evaluation.....	91
Figure 3-35: Conceptual algorithm flow chart for fast relative speed moving object detection .....	93
Figure 3-36: Illustration of MV position after time $T_y$ .....	99
Figure 4-1: Sub-pixel coordinates of control point A, B and C on the upright board .....	107
Figure 4-2: Captured image with aligned horizontal and vertical line. The roll and pitch angles are almost zero. ....	108

Figure 4-3: Checker board images used for the estimation of intrinsic parameters of the camera using Zhang’s method. ....	112
Figure 4-4: Simple image sequence containing a static (the house) and a moving object (the square box).....	114
Figure 4-5: The physical dimension of the house in the simple synthesized sequence. The World coordinates of vertices of the house are shown. Values are shown in meter. $Z_w$ is the distance of the house from the ego vehicle. Negative X-axis value means the object is on the left-hand side of the World coordinates. ....	115
Figure 4-6: Two successive frames of the synthesized sequence. The coordinates of the lower right corner of the house is read from the image. The coordinate reading for the previous frame is compared with the calculated result. ....	117
Figure 4-7: Illustration of road region detection and exclusion of areas with large MV amplitude. (a) Original captured image. (b) Result of block based road region detection. White colour represents the detected road region. (c) Blocks with MV amplitude larger than a threshold with white colour. (d) Combined mask from road region and large MV areas. (e) Combined region mask with holes filled. (f) Final ROI overlaid to the original image. ROI is the regions outside the white colour mask. ....	119
Figure 4-8: Detection of different vehicles on the road. ....	123
Figure 4-9: 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.....	125
Figure 4-10: The left side shows MVs of macroblocks of a typical frame with different QP. (a) QP=9, (c) QP=17, (e) QP=28, (g) QP=35, (i) QP=45. The right side shows the corresponding ROI with different QP constructed by using the amplitudes of MVs and the identified road region. It is observed that more coarse macroblocks (16x16) were used with lower QP, but the resulting ROI was essentially the same, preserving the regions with relatively slow speed vehicles at the front. ....	127
Figure 4-11: Illustration of images with un-successful detection. These images show the region of interest in grey which was constructed by combining the detected road region and the region with MV amplitude larger than a threshold. All images shows significant masking of the U-shape feature of these vehicles, leading to unsuccessful detection. ....	128
Figure 4-12: A snapshot of sequence A at frame 84. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the overlaid rectangle representing the identified position of the detected vehicle. ....	129
Figure 4-13: A snapshot of sequence B at frame 4. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front	

vehicle. (c) Original image with the overlaid rectangle representing the identified position of the detected vehicle. ....	129
Figure 4-14: A snapshot of sequence C at frame 92. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle. ....	129
Figure 4-15: A snapshot of sequence D at frame 6. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle. ....	130
Figure 4-16: A snapshot of sequence E at frame 790. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle. ....	130
Figure 4-17: A snapshot of sequence F at frame 228. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle. ....	130
Figure 4-18: A snapshot of sequence G at frame 742. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle. ....	131
Figure 4-19: (a) Binary image with white colour representing the area that is darker than the minimum grey-level of the identified road region. (b) Multiple regions for potential vehicle detection, circled in red colour..	133
Figure 4-20: Processing time for multiple vehicles in an image at different image resolution.....	134
Figure 4-21: Illustration of road region detection and exclusion of areas with small MV amplitude. (a) Original captured image. (b) Result of block based road region detection. White colour represents the detected road region. (c) Blocks with MV amplitude smaller than a threshold filling with white colour. (d) Combined mask from road region and small MV areas. (e) Combined region mask. (f) Final region of interest, with only the areas in the lower half of the image that are not filled with white. The overlaid image shows that most of the MVs on the relatively fast moving object are not masked.....	138
Figure 4-22: The air inflatable dummy car used for testing in this project. Its size is similar to a compact private car as shown in the pictures.....	140
Figure 4-23: Eight video sequences to evaluate the proposed fast relative speed moving object detection.....	142

Figure 4-24: Detection Result for Seq. P.....	145
Figure 4-25: Detection Result for Seq. Q.....	146
Figure 4-26: Detection Result for Seq. R.....	146
Figure 4-27: Detection Result for Seq. S.....	147
Figure 4-28: Detection Result for Seq. T.....	147
Figure 4-29: Detection Result for Seq. U.....	148
Figure 4-30: Detection Result for Seq. V.....	149
Figure 4-31: Detection Result for Seq. W. The tracking after HV shown in (c) is not successful.....	149
Figure 4-32: Loss-of-track after entering tracking mode.....	150
Figure 4-33: False-positive detection of the wall on the road.....	152
Figure 4-34: False-positive detection of the tree shadow on the road.....	152
Figure 4-35: False-positive detection of the fence on the side of the road.....	153
Figure 4-36: System architecture of the proposed MV based ADAS.....	156
Figure 5-1: Government departments and NGOs that had participated in the Trial Project. (a) Hong Kong Police Force. (b) Water Supplies Department. (c) Hong Kong Society for Rehabilitation. (d) The Neighbourhood Advice-action Council. (e) Fire Services Department. (f) Government Logistics Department.....	161
Figure 5-2: (a) Camera mounted to the windsheild behind the rear-view mirror. (b) Warning device to output audible alerts to the driver. (c) Embedded DSP hardware prototype. (d) Prototype installation location.....	162
Figure 7-1: Illustration of road gradient affecting the measured pitch angle for the camera installation.....	168
Figure 7-2: Road region identified in a captured frame and the corresponding diminished road region for use in the next frame. (a) Captured frame with identified road region. The identified road region is shaded with grey colour blocks. (b) Extracted road region from (a). The boundary is diminished by 1 block along the contour. The yellow contour is the difference between the original and diminished contours.....	170
Figure 7-3: Overlaid road region found from the previous frame. The region growth algorithm will start from 1 block along the yellow contour, reducing the required computation time due to reduced number of blocks to process.....	170
Figure 7-4: Integral image. (a) sum of pixel values above and left of $P(x,y)$ , (b) sum of pixel values above and left of $P_4(x,y) = A+B+C+D$ .....	174
Figure 7-5: Comparison of block structure for H.264/AVC and HEVC.....	179

<b>List of Tables</b>	<b>Page</b>
Table 1-1: ADAS Product Comparison .....	7
Table 1-2: Processor Performance Comparison .....	7
Table 1-3: Quantitative industrial requirements and constraints .....	8
Table 2-1: Average bitrate of video sequences with intra mode enabled and disabled. The video sequences were encoded with H.264 Baseline Profile with different motion search algorithm.....	28
Table 2-2: Average bitrate of video sequences with intra mode enabled. The video sequences were encoded with H.264/AVC Baseline Profile and Main Profile with EPZS search algorithm .....	29
Table 3-1: The three constraints for clustering decision for PPRVs $s_j$ .....	101
Table 4-1: The control points values for calibration.....	107
Table 4-2: Screen coordinates of checker corner points .....	109
Table 4-3: The deviation of calculated World coordinates from the measured World coordinates .....	111
Table 4-4: Deviation of calculated World coordinates from the measured World coordinates using Zhang's method .....	113
Table 4-5: The parameters used in the algorithm for successful detection of vehicles. These parameters are determined by repeated testing to the video sequences taken in Hong Kong for this project. ....	122
Table 4-6: Video sequences with different challenges to the proposed algorithm ....	124
Table 4-7: Detection result of seven image sequences. The last row shows the combined result of sequence A to G. ....	128
Table 4-8: Detection rate of the seven image sequences. The last row shows the detection rate of the sequence combined from A to G.....	128
Table 4-9: Average processing time in ms for low relative speed vehicle detection. The time for finding ROI and tracking is relatively stable. The detection time varies due to the difference in area for potential vehicle detection. ....	132
Table 4-10: Processing time of the algorithm at different resolutions. The processing time for 1280x720 and 1920x1080 was projected from the result at 640x480.....	134
Table 4-11: Comparison on the detection rate and false-positive rate of the proposed algorithm vs. other selected algorithms from renowned journals.....	136
Table 4-12: Video sequences with different challenges to the proposed algorithm ..	140
Table 4-13: Parameters for relative fast speed moving object detection .....	144
Table 4-14: Average processing time in ms for fast relative speed vehicle detection.....	154

Table 4-15: Required processors for different ADAS solutions..... 157  
Table 4-16: Cost comparison for different ADAS solutions ..... 158



## Glossary of Terms

ADAS	Advanced Driver Assistance System
DSP	Digital Signal Processor
BSDW	Blind-Spot Detection and Warning System
FCWS	Forward Collision Warning System
FOE	Focus of Expansion
FPS	Frames Per Second (or fps)
IPM	Inverse Perspective Mapping
LDWS	Lane Departure Warning System
ME	Motion Estimation
MV	Motion vector
PDWS	Pedestrian Detection and Warning System
PPRV	Planar parallax residual vector
PRC	People's Republic of China
RDO	Rate Distortion Optimisation
ROI	Region of Interest
SAD	Sum of Absolute Difference
SOC	System on Chip
SPO	State Patent Office
SVM	Support Vector Machine

# **1 Introduction**

## ***1.1 Research Objective***

Cost is one of the problems for wider adoption of Advanced Driver Assistance System (ADAS) in China (Lu and Wevers et al., 2010). The objective of this research project is to develop a low-cost ADAS by the shared use of motion vectors (MVs) from a H.264/AVC video encoder that was originally designed for video recording only. Since MVs from H.264/AVC video encoders are readily available without extra computational cost, successful utilisation of these MVs for moving object detection can simplify the system design, enabling an ADAS with video recording function by default.

The development of this ADAS, targeting low-cost and simplicity, bundling with video recording function by default, aims to address a wider adoption in China where the price of cars is mostly budgetary (CarNewsChina.com, 2015), and the demand for higher level of road safety is growing (Zhang and Zhang, 2010). The system will also be suitable for mass-produced consumer market, and for being used in other countries by optimising the system to meet the regulatory requirements in different countries.

## ***1.2 Advanced Driver Assistance System***

ADAS has been employed in many automobiles to assist drivers for improved road safety. It aims to help drivers recognise potentially hazardous situations. It detects objects surrounding the vehicle and gives warnings to alert the driver of potential hazards. Some systems can take over the vehicle control from the driver, managing the vehicle to decelerate or change direction for collision prevention. In this connection, many algorithms and techniques have been developed to recognise objects such as vehicles, motorcycles, pedestrian and cyclists.

ADAS can be divided into Forward Collision Warning System (FCWS), Lane Departure Warning System (LDWS), Pedestrian Detection and Warning System (PDWS), and Blind-Spot Detection and Warning System (BSDW).

Figure 1-1 shows a typical FCWS where a forward-looking camera is mounted at the windshield inside the car compartment. The FCWS detects the vehicle at the front and estimates the distance to the vehicle. If the distance to the vehicles is so small that the response time of the driver may not be able to prevent collision from happening, as illustrated in Figure 1-1(b), warning is issued by the system to alert the driver for proper actions.

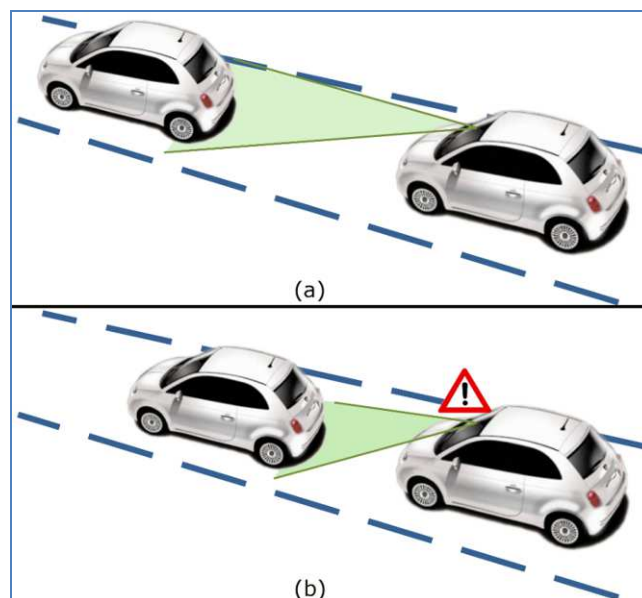


Figure 1-1: Forward Collision Warning System (FCWS). (a) Forward looking camera installed in a vehicle, monitoring the relative location of the front vehicle. (b) Warning signal is issued to alert the driver when the distance between the vehicles is too close.

For LDWS, its hardware is similar or even identical to that of FCWS where a forward-looking camera is also mounted at the windshield inside the car compartment. Figure 1-2 illustrates how LDWS works. Sequence (1) inside Figure 1-2 shows the front-side-looking mounting position of the camera at the windshield inside the car compartment. As depicted in sequence (2), driving lanes can be clearly captured with the

front-side-looking camera. The LDWS detects the driving lane marking from the image captured by the forward-looking camera. It monitors the lateral distance of the vehicle from lane markings appearing on both sides of the road. When the vehicle approaches to one side of the lane markings and is going to depart from the current driving lane, warnings with audible or haptic signal as shown in sequence (3) is issued to the driver. The driver then responds to the warning signal and performs corrective actions so that the car stays safely in the current driving lane.

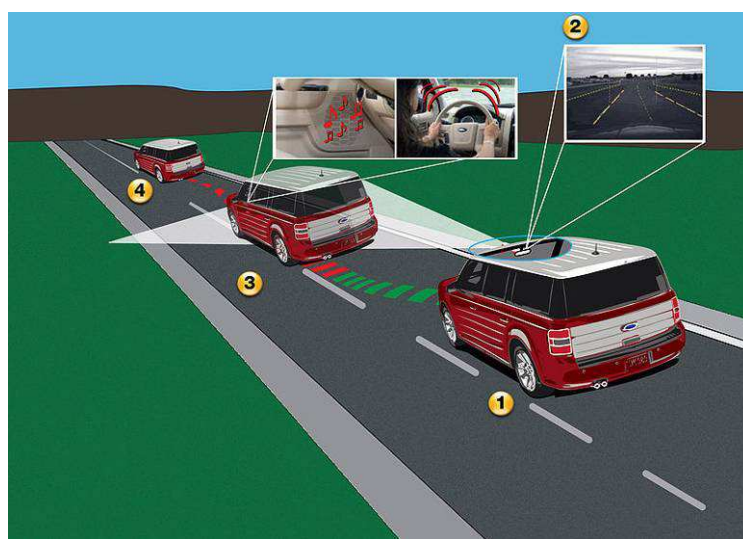


Figure 1-2: Illustration of LDWS. (1) A forward looking camera monitoring the driving lanes. (2) Capture image from the forward looking camera. (3) The instance with warning signal output to notify the driver when the vehicle is cutting the driving lane. (4) Vehicle driving back to the designated lanes after the driver has corrected the driving path.

(source: [http://fr.wikipedia.org/wiki/Fichier:Lane\\_Departure\\_Warning.jpg](http://fr.wikipedia.org/wiki/Fichier:Lane_Departure_Warning.jpg))

PDWS can use the same hardware for LDWS and FCWS. When pedestrians are moving laterally relative to vehicles, the relative moving speed of the pedestrians is significant and is potentially fatal. PDWS aims to inform the driver of potential collision with pedestrians so as to reduce the severity of accidents. Since the size of pedestrians is small comparing to the size of vehicles, pedestrians can hardly be detected when they are far away from the camera. Figure 1-3 shows the size of a pedestrian, marked with a red triangle, appeared in the image at a distance of 60m from the camera. The system detects pedestrians and estimates the distance to the vehicle. It also estimates the moving

path of the pedestrian relative to the vehicle. If the relative moving path will result in a collision within a predefined time, say two seconds, the system will issue a warning to alert the driver for proper actions to prevent the accident.

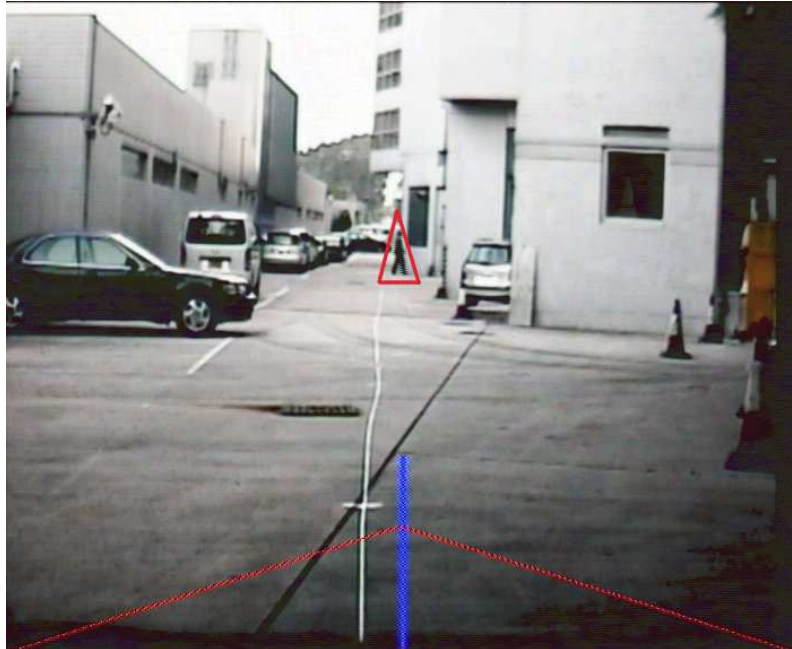


Figure 1-3: A pedestrian at 60m away from the vehicle marked with a red triangle.

BSDWS operates in the same principle as FCWS. The difference is that there are two cameras for BSDWS. They are mounted to both sides of a vehicle, known as the blind-spot zones, to monitor the areas that the driver cannot see. Figure 1-4 shows a BSDWS with cameras mounted to the two wing mirrors. Each camera is responsible for monitoring one blind-spot zone of the driver. The system monitors if objects appear in the blind spot zones shaded in blue and brown in Figure 1-4. When there is an object detected, warnings or indication will be issued to notify the driver. An indicator will turn on in the wing mirror to notify the driver of the presence of an object at the blind spot zone.

There are different sensors used in ADAS, such as Vision, Infrared, Radar, Lidar and Ultrasound. Radar, Lidar and Ultrasound sensors are usually referred to as active sensors. This is because they detect objects by sensing their emitted signals. An infrared sensor is

a special camera sensor which senses the thermal spectrum of the scene rather than the visible spectrum. The research in this project was concentrated on monocular vision based ADAS mainly because of the cost and versatility of the vision sensor. One forward looking camera can be used for multiple functions, such as FCWS, LDWS and PDWS, whereas the use of an active sensor will need to combine with a vision sensor to deliver all these functions.

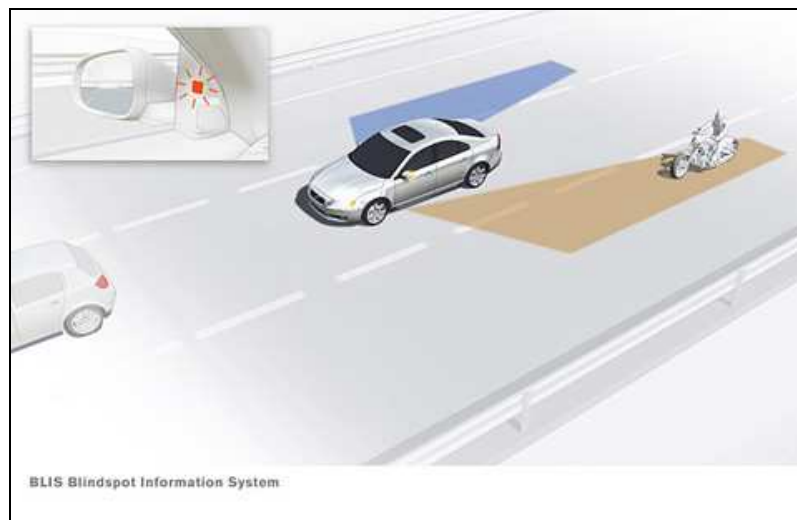


Figure 1-4: Blind Spot Detection and Warning System (BSDWS). Cameras installed at the wing mirrors looking backward, covering the areas that are not easily seen by the driver. (Source: <http://images.businessweek.com/ss/06/09/cartech/source/2.htm>)

### **1.3 ADAS Products in the Market**

Table 1-1 shows a comparison of typical ADAS products in the retail market. It shows functions equipped with these products, and the embedded processors that were used. The retail price was referenced from popular websites such as Amazon, Ebay and Taobao. Mobileye is one of the earliest suppliers of ADAS. It also supplies ADAS to car makers in addition to that in the retail market. It makes use of its proprietary EyeQ2 processor to perform the processing tasks for Lane Departure Warning (LDW), Forward Collision Warning (FCW) and Pedestrian Detection Warning (PDW). The EyeQ2 processor was designed with highly parallelised architecture for real-time image processing applications, enabling it to process image frames in parallel for different

recognition tasks. AiDriving ADAS A1 is a product from China, it made use of a DSP for handling image processing task, and an ARM based embedded processor with video encoding function for handling the video recording task. Papago P3 is a product from Taiwan, it made use of a System-on-chip for both video recording and image processing. The processor clock speed for ADAS A1 and Papago P3 was around 700MHz. Although the clock speed of the EyeQ2 processor is only 332MHz, its multi-core customised design enables it to perform all the image processing tasks, including the more demanding pedestrian detection and warning (PDW) function.

In order to understand the limitations of embedded processors, the performance benchmark of typical embedded processors used for real-time image processing is shown in Table 1-2. CoreMark is a benchmark for embedded systems developed by an industry alliance EEMBC (<http://www.eembc.org>). The performance of an Intel Pentium Dual-Core E5300 processor which was used in a desktop computer for the development of this project, and an Intel i3-3217UE processor were included in the table to show the performance differences between embedded processors and desktop processors.

As seen from the table, modern embedded processors are still running slower than an Intel E5300 that was launched to the market in 2008. In particular, the benchmark for DM3730 from Texas Instruments (TI), which has an ARM Cortex A8 core and a DSP, was reported with the use of its processor core only.

In addition, the benchmark for AM57xx from TI was reported with the use of its DSP core only. The use of both the ARM and DSP cores in a DM3730 is expected to achieve a benchmark of around 6073, which is still slower than an Intel E5300 processor. However, with code optimisation and fast image processing library for running on the DSP, it is expected that a System-on-Chip (SoC) with performance similar to the DM3730 is able to fulfil the real-time processing requirements for an ADAS.

Table 1-1: ADAS Product Comparison

Product	Market price	Functions	Processor
Mobileye CS-270 	US\$729	<input checked="" type="checkbox"/> LDW <input checked="" type="checkbox"/> FCW <input checked="" type="checkbox"/> PDW	Proprietary EyeQ2 SoC @332MHz Five Vision Computing Engines Three Vector Microcode Processor Two floating point MIPS34K CPU
AiDriving ADAS A1 	US\$1000	<input checked="" type="checkbox"/> LDW <input checked="" type="checkbox"/> FCW <input checked="" type="checkbox"/> Video Recording	TI DSP @650MHz One ARM 32-bit CPU
Papago P3 	US\$335	<input checked="" type="checkbox"/> LDW <input checked="" type="checkbox"/> FCW <input checked="" type="checkbox"/> Video Recording	Ambarella SoC @700MHz One ARM 32-bit CPU One Image DSP One Video DSP

Table 1-2: Processor Performance Comparison

Processor	Clock Speed	CoreMark	CoreMark/MHz
nVidia Tegra 2	1GHz	5866.39	5.87
TI DM3730 Cortex A8 Core	1GHz	2530	2.53
TI AM57xx C66x DSP	750MHz	3543.26	4.72
Intel Pentium Dual-Core E5300	2.6GHz	8885.30	3.42
Intel i3-3217UE	1.6MHz	24231	15.41



## **1.4 Industrial Requirements and Constraints**

Since the performance and memory resources of embedded processors are still lower than that of desktop processors, the major constraints in selecting an embedded processor is to fulfil the real-time computation under tight memory and computational resources, and is able to achieve the cost target and energy efficiency to be competitive in the market. Since the Consumer Electronics market as well as the Automotive market are highly competitive, the embedded processor selected for ADAS applications must be highly cost effective. Achieving all of these constraints is difficult for demanding computer vision applications. Nevertheless, the quantitative requirements were defined at the beginning of this project with reference to available system-on-chips in the market.

Table 1-3: Quantitative industrial requirements and constraints

	<b>Constraints</b>	<b>Description</b>
1	Processor	Use Off-the-Shelf embedded processor / system-on-chip
2	Clock Speed	Should be below 2GHz in view of the availability of embedded system-on-chip in the market
3	Camera resolution	1280 x 720 or better, with 30 frames per second
4	Embedded memory	1GB or below, DDR3 or DDR4 RAM
5	Power consumption	Should be less than 10W
6	Cost	Main component cost should be less than US\$150

## **1.5 Statement of Originality**

There were new ideas generated and new algorithms developed during the research and development of this project. The originality includes the following items:

1. An original integrated approach to the shared-use of motion vectors from the H.264/AVC encoder for moving object detection, capable of running in real-time with performance on a par with other state-of-the-art approaches proposed by other authors, targeting for application to an Advanced Driver Assistance System. This approach comprises of dividing the detection into relatively slow and relatively fast moving objects, block based road region growth, region of interest reduction using amplitudes of motion vectors and road region information, detection of relatively slow moving objects based on generic line features, detection of relatively fast moving objects based on planar parallax residuals, and also tracking of detected objects using image projection information and dynamic template matching for relatively slow and relatively fast objects respectively.
2. A novel approach to address the problems of moving object detection due to the limited precision of motion vectors from a typical H.264/AVC encoder for relatively slow moving objects especially those near the focus of expansion. This approach comprises of dividing the regions of interest into areas for detecting relatively slow and relatively fast moving objects according to the amplitudes of motion vectors.
3. A novel approach to address the problem of moving object detection due to the erroneous motion vectors from a typical H.264/AVC encoder caused by the coding efficiency optimised motion estimation process. This approach comprises of hypothesis generation according to the amplitude, direction and location of ego motion compensated planar parallax residuals, and also hypothesis verification according to the dynamic template matching evaluation.
4. The division of regions of interest into regions for relatively slow and relatively fast moving objects detection is making use of the amplitude of MVs, position of the focus of expansion and the result of road region detection..

5. An original algorithm for relatively fast moving object detection by using planar parallax residual and the associated filtering and clustering techniques using amplitude, position and direction constraints.
6. An original algorithm for relatively slow moving object detection by locating the darkest area in the image making use of a gray level threshold obtained from the road region detection algorithm, and followed by the detection and refinement of generic line features.
7. An original algorithm for the tracking of relatively slow moving objects by the expansion of the bounding rectangle of the found object and refinement of the boundaries by horizontal and vertical projection images.
8. An original algorithm for the tracking of relatively fast moving objects by dynamically updating the template for comparison followed by simple template matching technique.
9. An original algorithm for the road region detection by block based region growth technique, with dynamic update of the intensity of the road for comparison.
10. Incorporation of a six-degree-of-freedom inertial sensor to assist accurate estimation of ego motion and focus of expansion.
11. A novel camera calibration procedure that is designed to simplify the camera installation process, targeting for mass production devices.

## ***1.6 Organisation of Chapters***

This report is organised as follows: Chapter 2 is the literature review. It reports the key findings of that lead to the research direction of this project. Chapter 3 reports the proposed algorithm framework, which is the major part of the research and development of this project. Chapter 4 reports the test and evaluation results of the proposed algorithm framework. Chapter 5 reports the activities that have been done for commercialising the R&D results. Chapter 6 reports the patents and peer-reviewed papers that have been published or accepted. Chapter 7 elaborates further development

that can be carried out in the future. Chapter 8 is the outline of future research and development direction. Chapter 9 is the outline of further works for commercialisation. Chapter 10 is the conclusion for this research project. Chapter 11 is the references.

## ***1.7 Chapter Summary***

This Chapter introduced the objective of this project which is the development of a low-cost ADAS. It also briefly introduced the four major functions of ADAS which include Forward Collision Warning System (FCWS), Lane Departure Warning System (LDWS), Pedestrian Detection and Warning System (PDWS), and Blind-Spot Detection and Warning System (BSDW). Moreover, the content structure of this report, and the originality of this study were also described for easier reference by the readers.

In the next Chapter, a literature review on vision based ADAS is presented.

## **2 Literature Review**

The literature review has been concentrated on technologies related to vision based ADAS because of the cost and versatility of camera sensors. For instance, the video output of a forward looking camera can be used for both lane detection and moving object detection. Different kinds of detection can also be done with different image processing algorithms. Vision based ADAS works by first capturing the camera image, followed by image processing of regions of interest, extraction and feature recognition. When certain patterns or features of objects in the image are recognised, decision on the level of severity will be performed by the processor. If there is potential danger, a warning signal is output to draw the driver's attention. The patterns and objects that are of interest for the ADAS application are driving lanes, vehicles at the front and at both sides of the car.

### ***2.1 Lane Detection***

Since lane markings are flatly painted on the road, vision based detection of lanes usually relies on the contrast difference between the lane markings and the road surface. There have been some well know projects that developed methods for lane detection algorithms in the past decades, such as AURORA (Chen and Jochem et al., 1995), GOLD (Bertozzi and Broggi, 1998) and TFALDA (Yim and Se-young, 2003). A comprehensive review has been done by McCall et al. (2006) for methods proposed before 2005. Most of the lane detection algorithms involve lane feature extraction, outlier rejection and tracking.

For feature extraction, edges of lane markings are one of the most significant features to extract (Kluge and Lakshmanan, 1995, Li and Zheng et al., 2004, Wang and Teoh et al.,

2004, Chapuis and Aufrere et al., 2002, Wang and Bai et al., 2008). The edge extraction can be noisy if the lane markings are not clearly marked on the road, or the threshold used for extraction is not determined appropriately. Also, if the edge extraction is done on a greyscale image, there is a chance that the edges of coloured road markings, such as those painted in yellow or red, cannot be extracted correctly. This is because the contrast of coloured road markings is relatively low in greyscale images. To overcome this problem, some edge extraction methods made use of some transformations on the colour image to improve the contrast for extraction (Cheng and Jeng et al., 2006, Sun and Tsai et al., 2006). To reduce the false extraction on lane markings, there have been attempts using the intensity change pattern of dark-bright-dark of lane markings to filter falsely detected edges (Bertozzi and Broggi, 1998, Ieng and Tarel et al., 2003).

For lane detection, the algorithm to identify lanes on the road after initial lane feature extraction include the Hough Transform (Li and Zheng et al., 2004), lane curvature modelling (Wang and Teoh et al., 2004, Li and Fang et al., 2015) and probabilistic estimation (Kluge and Lakshmanan, 1995, Liu and Worgotter et al., 2013). Therefore, successful detection of lanes is heavily dependent on the feature extraction algorithm. Some algorithms make use of the Inverse Perspective Mapping (IPM) technique to obtain a bird's eye view image from the original captured image (Muad and Hussain et al., 2004, Sehestedt and Kodagoda et al., 2007). Figure 2-1(a) shows a typical image captured on the road. Figure 2-1(b) shows the corresponding bird's eye view image transformed from Figure 2-1(a) using the IPM technique. As seen from Figure 2-1(b), both the lane markings and texts on the road become clear straight lines. This can facilitate simpler method for recognition by simplifying the edge extraction and detection method since all road markings can be converted to straight line segments with uniform width. However, the conversion to bird's eye view image requires high

computational cost or dedicated hardware to achieve real-time performance. It is not suitable for off-the-shelf microprocessors where hardware for fast perspective transform is not available.

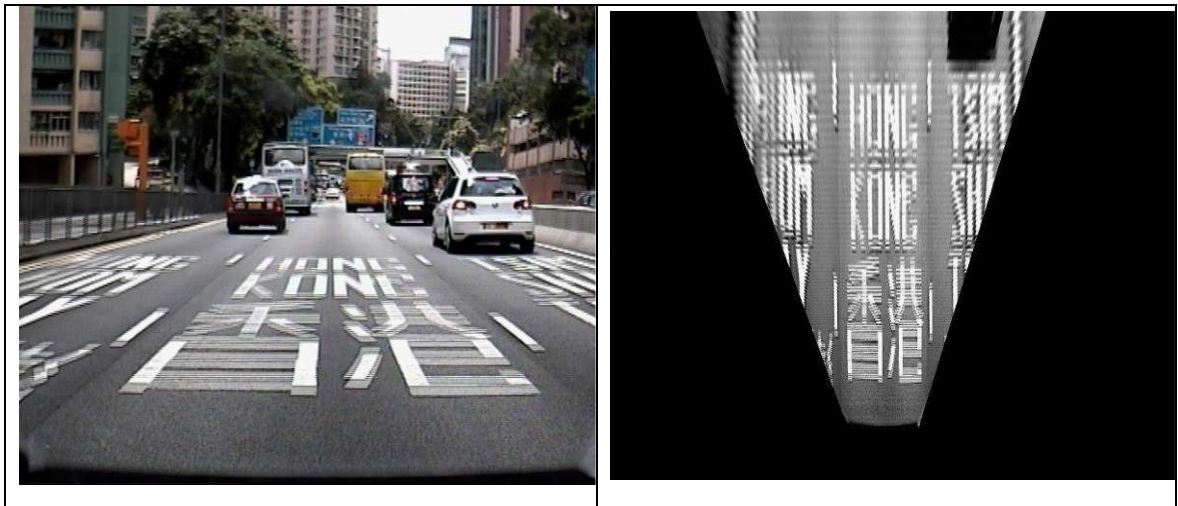


Figure 2-1: (a) Normal captured image. (b) The bird's eye view image transformed from (a) using IPM technique

For lane tracking after detection, the most popular algorithms used are Kalman Filter (Kosecka and Blasi et al., 1998, Lim and Seng et al., 2009) and Particle Filter (Wang and Bai et al., 2008, Apostoloff and Zelinsky, 2003). Both the Kalman Filter and Particle Filter are recursive state estimators making use of Bayes filter. Bayes Filtering is a method for predicting and updating the state of a dynamical system from measurable information. Kalman Filters are designed for estimating linear systems with Gaussian noise. Some modified Kalman Filters, such as Extended Kalman Filter and Unscented Kalman Filter, are derived to estimate non-linear systems. Kalman Filters are parametric, requiring a mathematical model to describe the behaviour of the system.

Particle Filter performs the estimation by a set of random points known as particles. Each particle contains a set of state variables to describe the system. It can represent any systems with complex models. No exact representation of the system model is required.

However, Particle Filter demands higher computational cost because of the larger set of state variables required to describe the system.

Although there are commercially available systems such as AutoVue (Bendix, 2015) and Mobileye LDW (Mobileye, 2015), lane detection is still an active research topic as there are still so many challenges to further enhance reliability. For instance, the effect of shadows during day time and the high dynamic lighting conditions during night time are still challenging to vision based lane detection methods.

## **2.2 Vehicle Detection**

There have been many methods proposed for vehicle detection in the past decades (Sivaraman and Trivedi, 2013). 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 to their characteristics. For instance, the left-right symmetrical characteristics of vehicles can be used as one of the features for vehicle detection. Because most vehicles look rectangular from the front and rear, the edges and corners of vehicles are also frequently used as the clues for vehicle detection (Broggi and Cerri et al., 2004, Du and Papanikolopoulos, 1997, Kuehnle, 1991, Kuo and Pai et al., 2011, Liu and Zheng et al., 2005, Zielke and Brauckmann et al., 1992). Detection methods that analyse the shadow underneath a vehicle have also been proposed (Tzomakas and Seelen, 1998, Manuel Ibarra Arenado and Juan Maria Perez Oria et al., 2014).

For statistical based approaches, feature extraction based on Haar (Haselhoff and Kummert, 2009, Yong and Zhang et al., 2011, Chang and Cho, 2010) and Histogram of Oriented Gradients (HOG) (Mao and Xie et al., 2010, Sivaraman and Trivedi, 2010) have been reported. They combine with the use of statistical training algorithms such as



Adaboost (Freund and Schapire, 1997) and Support Vector Machine (SVM) (Cortes and Vapnik, 1995) to generate the classifiers for high true-positive and low false-positive detection rates. In the hope of increasing the true-positive detection rate and decreasing false-positive rate, there has been a trend for using online learning, allowing new samples taken to be added to the classifiers (Sivaraman and Trivedi, 2010, Chang and Cho, 2010). But the problem with online learning is the validity of samples collected. It is difficult to determine automatically if the samples collected are true-positive samples or true-negative samples without human interpretation.

Both the feature based and statistical based approaches require prior knowledge on the characteristics of the objects to be detected. The detection involves identifying both rear-view vehicles and vehicles on the road viewing at different perspectives. There are different characteristics appearing on the vehicles to be detected when they are viewed at angles. Therefore, multiple sets of features or statistical models are required for detecting vehicles viewing differently. This implies additional computation for each set of features or models for improved detection.

Other techniques have been reported that can supplement the vehicle detection tasks. For instance, Optical Flow algorithms can be used to differentiate moving objects from the background. The direction of movement of pixels in the consecutively captured image sequence is known as Optical Flow. By identifying the features of clustered moving pixels in the image sequence, moving vehicles can be detected by applying feature based techniques. However, the ego motion of the camera needs to be evaluated for extracting the true ground motion vectors instead of those relative to the moving camera. Some research activities have been reported on the use of optical flow for ego motion estimation and moving object detection (Giachetti and Campani et al., 1998, Klappstein and Stein et al., 2006).

In addition, the disparity of objects from the captured images of stereo cameras is able to provide important depth information for object recognition. Therefore the use of stereo images for object detection is able to combine with other feature extraction techniques such as optical flow, symmetry and / or statistical training for object detection (Bertozzi and Broggi, 1998, El Ansari and Stéphane et al., 2010, Toulminet and Bertozzi et al., 2006).

The problem with stereo vision is the highly demanding calibration process. The two cameras need to be calibrated so that the epipolar line lies on the same horizontal axis in the images captured by both cameras. There are so many uncertainties that may affect the calibration result for Stereo cameras installed in vehicles are affected by uncontrollable parameters such as shocks, vibrations and even collisions may drastically affect calibration. Although a monocular camera is a lot simpler, it also requires a good calibration for accurate estimation of physical parameters of detected objects. The refinement of calibration process opens up new research opportunity to research on how stereo cameras can be setup quickly and reduce the impact of environmental factors.

### ***2.3 Generic Moving Object Detection***

The vehicle detection algorithms mentioned in Section 2.2 are used specifically for detecting vehicles. Another consideration for vehicle detection is that drivers may not be interested to recognise vehicles that are not posing any trouble to them. In contrast, they are required to know if there is any object that may pose danger to them and to be alerted early enough to mitigate the situation. Therefore, the methods on generic (or non-parametric) moving object detection with a moving observer (i.e. the camera) have been reported in this Chapter.

Moving objection detection methods are commonly used in surveillance systems with cameras mounted to fixed positions. When the camera is fixed, the simplest idea for detecting moving objects is simply by frame differencing (Jung and Sukhatme, 2004). However, for cameras mounted to automobiles, they move with vehicles' translational and rotational motions. Many techniques for fixed cameras are not applicable to situations with moving cameras. For the case with a moving camera, the ego-motion of the camera has to be estimated for compensation in order to estimate the ground truth motion of the independently moving object.

For a monocular camera setup, the movements of independently moving objects that appear in the screen are composed of both their ground truth motions and the ego motion of the camera. Also, a Monocular camera lacks the depth information, the distance and moving speed of independently moving objects can only be recovered by the use of algorithms.

To detect an independently moving object, the motion vectors after ego motion compensation in the scene are examined. The two most commonly used methods for monocular vision are planar parallax violation (Giachetti and Campani et al., 1998, Baehring and Simon et al., 2005) and flow field angle criterion (Pauwels and Van Hulle, 2004, Clauss and Bayerl et al., 2006). Klappstein and Stein et al. (2006) have developed a method called Two-View constraint. This method detects the optical flow field irregularity using positive depth constraint and positive height constraint. However, these constraints are referring to the difference in the measured motion field to the expected motion field of a static object when the camera is moving. Therefore, the Two-View constraints can also be regarded as planar parallax violation detection. The flow field angle criterion detects the angle of the flow field vector to the focus of expansion (FOE). FOE is the point where objects in the scene are apparently emerging

from when the camera is moving. If the angle is larger than a threshold to the expected flow field due to ego-motion, a moving object is detected.

### 2.3.1 Planar Parallax Method

The relationship of a point on the World coordinates and its corresponding point on the screen can be represented by the planar parallax diagram shown in Figure 2-2 (Baehring and Simon et al., 2005). The green plane is the camera plane at the current frame at time  $T_t$ , the red plane is the camera plane at the previous frame at time  $T_{t-1}$ ,  $P_w$  is a point above the ground plane,  $p_1$  and  $p_2$  are the projected points of  $P_w$  on the image planes at time  $T_{t-1}$  and  $T_t$  respectively,  $P_1^G$  and  $P_2^G$  are the points on the ground plane due to  $P_w$  when viewed by camera at  $C_{t-1}$  and  $C_t$  respectively,  $p_{2G}$  is the point virtually projected to the image plane at  $T_{t-1}$  due to  $P_2^G$  on the ground plane. Since the image contains no depth information, the point correspondence between points in the screen in successive frames is estimated by assuming that these points are lying on the ground plane. By knowing the camera translation and rotation between successive frames, a ground plane homography matrix can be estimated. This planar homography matrix is only able to identify the point correspondence of the projected point  $P_2^G$  on the ground plane to  $p_{2G}$  on the screen in the previous frame, rather than the true point correspondence at  $p_1$  in the previous frame.



Motion vectors (MVs) in H.264/AVC video encoding are determined by minimising the cost function ( $J$ ) that essentially consists of a distortion term ( $D$ ) and a rate term ( $R$ ), as shown in Equation (2.1).  $J$  is known as the Rate Distortion Optimisation (RDO) cost. The distortion term ( $D$ ) is the matching function that is usually evaluated by the Sum of Absolute Difference (SAD) with formula shown in Equation (2.2), where  $s$  is the signal from the original video,  $c$  is the signal from the coded video,  $B_x \times B_y$  is the block size for the evaluation,  $m = (m_x, m_y)^T$  is the motion vector (MV).

$$J = D + \lambda R \quad (2.1)$$

$$SAD(s, c(\mathbf{m})) = \sum_{x=1}^{B_x} \sum_{y=1}^{B_y} |s(x, y) - c(x - m_x, y - m_y)| \quad (2.2)$$

Each MV in a H.264/AVC video encoder represents an image block of variable size of either 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, or 4x4, depending on the decision of the motion estimation algorithm (Chiu and Siu, 2010). Motion vectors are generated when a motion estimation algorithm is run during video encoding. The motion vectors represent the displacement of blocks between successive frames. The goal of motion estimation in H.264 video compression is to achieve high quality video with the lowest possible bit rate by correlating the patterns in the past video frames to the current video frame. So, if the motion vectors directly available from motion estimation for video compression are used for moving object detection, there will be many outliers that need to be eliminated before accurate moving object detection can be performed.

Motion estimation is highly computationally expensive (Chan and Siu, 2001). To reduce the computational cost for finding the best match, there have been studies on many fast search algorithms. There is a set of reference software publicly available from Heinrich Hertz institute (HHI 2012) that is commonly used for educational purposes and for benchmarking among different implementation approaches of researchers. In addition to the generic full search algorithm that searches for the

minimum sum of absolute difference (SAD) value inside the search window, three fast search algorithms, namely Uneven Multi-Hexagon Search (UMHexgonS) (Chen et al. 2002), Simplified Hexagon Search (SHS) (Yi et al. 2005) and Enhanced Predictive Zonal Search (EPZS) (Tourapis and Tourapis 2003) are included in the reference software. These fast search algorithms mainly comprise of three steps, namely the initial predictor selection, adaptive early termination, and prediction refinement.

The initial predictor selection stage selects a MV predictor among a set of predictors that are potentially giving good estimation results. Instead of examining all possible positions in a search window to determine the best predictor, these fast search algorithms only examine a smaller set of positions according to some temporal and / or spatial constraints.

In the adaptive early termination stage, the MV search is terminated by examining the distortion evaluated by SAD. If it is smaller than a threshold determined by minimum distortion values of previously examined blocks, MV search can be terminated.

In the prediction refinement stage, the MV is refined by searching for the best predictor with a search pattern around the best predictor. The search pattern is designed to reduce the chance of being trapped in a local minimum, and to reduce the number of required search for computation efficiency.

With a fixed video frame rate of small frame-to-frame interval, optical flow can also be estimated by the block matching approach, such as that used in the motion estimation process in the H.264/AVC encoder (Davis and Karul et al., 1995, Chi and Tran et al., 2007). The optical flow can simply be estimated by the motion vector divided by the time interval between successive frames.

The equivalence of motion estimation by the block matching and optical flow methods implies that the MVs for video coding can be used for moving object detection and vice versa. However, the ultimate goal of the use of MVs in the video encoder is to achieve the best coding efficiency possible. The resultant MVs do not guarantee to represent the true motion of objects in the scene. They are therefore noisy for moving object detection. It is a challenging task to make use of such noisy motion information for moving object detection. Also, the video coding must be executed in real time for encoding live video without frame loss or reduced frame rate. This implies that the moving object detection algorithm for use with MVs from H.264/AVC video encoding must be highly efficient to allow completion within the duration between successive frames.

### **2.3.3 Ego Motion Estimation**

Since both the observer vehicle and other objects for detection are moving on the road, the MVs obtained from the H.264/AVC encoder are the result of motions of both the observer vehicle and the objects on the road. The movement of the observer, also known as ego motion, has to be compensated in the captured image sequence so that the actual motion of objects relative to the ground can be obtained. This process is known as ego motion compensation.

For image based ego-motion estimation, feature points from the captured image sequence are tracked to find the corresponding flow field which is known as optical flow. Selected flow fields between two successive images can be used to estimate the planar homography matrix or the Fundamental matrix between the two images. A motion model that transforms the motions in the 3-D space to the captured 2-D image is described by the homography matrix. For a set of point correspondences  $p_i \leftrightarrow p'_i$



between two images, the Fundamental matrix  $F$  can be represented by Equation (2.3). And for each point correspondence  $p_i \leftrightarrow p'_i$ , it satisfies Equation (2.4). Alternatively, for each point correspondences  $p_i \leftrightarrow p'_i$  on a plane (such as the road plane) between two images, it satisfies Equation (2.5), where  $H$  is the planar homography matrix of the plane in the two images.

$$F = \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \quad (2.3)$$

$$p_i^T \times F p_i = 0 \quad (2.4)$$

$$p'_i = H p_i \quad (2.5)$$

The Perspective model with eight parameters is the most sophisticated model to account for both rotational and translational movements of objects in any direction (Su and Sun et al., 2005). For accurate ego-motion estimation, the Perspective model should be applied to account for movements of objects in all directions with change of depth.

As summarised by Derpanis (2006), optical flow estimation methods can be classified as Direct Matching Methods (Horn and Weldon Jr, 1988, Stein and Mano et al., 2000, Ke and Kanade, 2003), Differential Methods (1981, 1981, Bruhn and Weickert et al., 2005, Weickert and Schnörr, 2001, Liu and Hong et al., 1998) and Frequency Based Methods (Langer and Mann, 2003, Huang and Chen, 1995). Among these methods, the Direct Matching Methods (Horn and Weldon Jr, 1988, Stein and Mano et al., 2000, Ke and Kanade, 2003) make use of a sum of squared difference (SSD) cost function to estimate the ego motion parameters. This method is similar to the motion estimation algorithm used in the H.264/AVC encoder. But the cost function used in the H.264/AVC encoder is a Rate-Distortion function (Wiegand and Sullivan et al., 2003)

that involves Sum of Absolute Difference (SAD) as the cost function and the parameter related to the resulting encoding efficiency.

To estimate the Fundamental Matrix  $F$  or the homography matrix  $H$ , a set of points in the image are selected to fit into Equation (2.4) or (2.5). This set of equations are used for the minimisation of a cost function so that the parameters in the matrix  $F$  or  $H$  can be determined (Wang and Cai et al., 2012). The selection of feature points is crucially important to the accuracy of the estimated Fundamental matrix and homography matrix. Feature points extraction methods using Harris corner (1988), “SURF” (Bay and Tuytelaars et al., 2006), “FAST” (Rosten and Drummond, 2006), “BRIEF” and “SIFT” (Lowe, 2004), have been used by many authors for feature points extraction (Xiaqiong and Xiangning et al., 2011, Nedevschi and Golban et al., 2009, del-Blanco and Garcia et al., 2009, Scaramuzza and Siegwart, 2008). The SIFT feature detector have proven to be a method robust to scale, rotation and illumination variations (Heinly and Dunn et al., 2012). Comparing with more recent methods such as SURF and ORB (Rublee and Rabaud et al., 2011), SIFT has been proven to be much more computationally expensive (Rublee and Rabaud et al., 2011).

For application to Advanced Driver Assistance Systems, the moving camera results in perspective changes of objects in the scene. The chance of having photometric variation is also high because of the rapid change of traffic and road conditions. The scene change due to rotation is relatively small since the degree of rotation is limited by the inclinations of roads and the cornering speed of the vehicle. Therefore, the feature point computation method should be robust to scale and photometric variation. It should be computationally efficient as well as robust to small rotations. According to the comparison done by Heinly and Dunn et al. (2012), ORB is able to fulfil the

requirements with computation time of an order of magnitude faster than SIFT under environments with stable illumination.

## 2.4 H.264/AVC Motion Vector Overview

MVs are generated when a motion estimation (ME) algorithm is run during the video encoding process. The MVs represent the displacement of objects between successive frames. The goal of ME in video compression is to achieve the “best” quality video with the lowest possible bit rate by correlating the patterns in the past video frames to the current video frame. MVs are therefore not necessarily representing the movement of objects in the video stream.

In addition to the eight variable block sizes ranging from 4x4 to 16x16 samples shown in Figure 2-3, there is a special mode called “SKIP” which is used with 16x16 macroblock size where the motion estimation need not be performed. The identification of macroblock using SKIP mode can reduce the computation time for finding suitable mode and MV without degrading the video coding performance (Zeng and Cai et al., 2009).

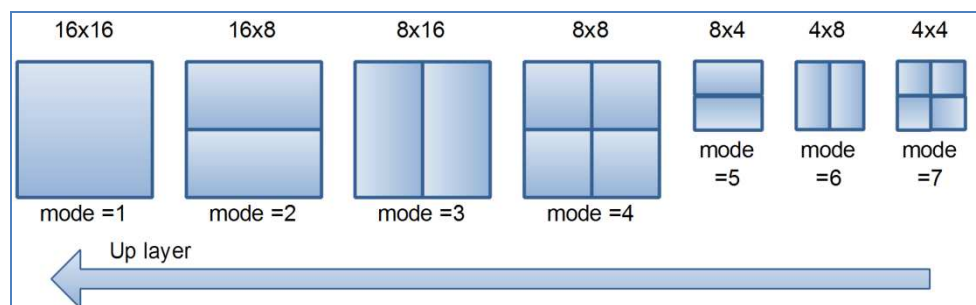


Figure 2-3: Block size for motion estimation. Primary macro-block size is mode 1 at 16x16. Smaller partitions from 16x8 down to 4x4 are possible according to the decision of the motion estimation algorithm.

The H.264/AVC standard allows each captured image to be encoded as an “I”, “P” or “B” frame. An “I” frame is an intra-coded image. It can be decoded without referencing to other frames and is regarded as the least compressible frame. A “B” frame is known

as a bi-predictive frame that contains both image and difference data. It can take multiple frames from previous or next frames as reference. A “P” frame is known as a predicted frame or a delta frame. It contains the MVs that represent changes in the image from its previous frame up to a quarter pixel precision. The proposed system developed in this project only uses the MVs from P-frames for moving object detection.

An evaluation of MVs using the open-source H.264/AVC software JM18.4 (JVT, 2012) on some sample sequences was conducted. 20 frames from four Daimler image sequences (Vaudrey and Rabe et al., 2008) were selected for H.264/AVC encoding. The Daimler image sequence name referred in this report follows the file name of the sequence used by Vaudrey et al.. Four image sequences, namely “Construction site”, “Crazy Turn Left”, “Dancing Light” and “Intern on Bike”, were used for evaluation.

#### **2.4.1 Bitrate Comparison**

The four video sequences shown were encoded with IPPP frame structure in Baseline profile. Each video sequence was encoded with Intra-mode within P-frame enabled and disabled respectively. Also, three different motion search algorithms, namely Full-Search (FS), Un-Symmetric-Hexagon (UMH), and Enhanced Predictive Zonal Search (EPZS), for encoding were used for each video sequence. Disabling intra-mode can make sure that there are MVs for all macroblocks. It provides more information for motion computation.

From the statistical output of the encoder, it was found that the average bits per frame of the video sequence can increase significantly with intra-mode disabled, as detailed in Table 2-1. For instance, the Construction Site sequence has average bitrate of 2,287.21 and 3,041.92 kb/s when intra-mode is enabled and disabled respectively using Full-Search algorithm. The difference is 33.0%. Also, the difference in the

average signal to noise ratio (SNR) of the Y component is less than 0.5dB for the same video sequence using different search algorithm and different mode. This result indicates that intra mode can effectively reduce the bitrate of the encoded sequence.

The loss in video quality when using different search algorithm is small.

Table 2-1: Average bitrate of video sequences with intra mode enabled and disabled. The video sequences were encoded with H.264 Baseline Profile with different motion search algorithm

Sequence	Mode	Bitrate (kb/s)	Motion Search Algorithm (H.264 Baseline Profile)		
			Full-search	UM Hexagon	EPZS
Construction site	Intra-on	I Slice	197.11	197.11	197.11
		P Slice	2089.89	2216.46	2137.77
		B Slice	0.00	0.00	0.00
		Total	<b>2287.21</b>	<b>2413.78</b>	<b>2335.09</b>
	SNR Y (dB)		<b>37.60</b>	<b>37.62</b>	<b>37.59</b>
	Intra-off	I Slice	197.11	197.11	197.11
		P Slice	2844.60	2814.96	2593.80
		B Slice	0.00	0.00	0.00
		Total	3041.92	3012.28	2791.12
	SNR Y (dB)		37.51	37.49	37.48
Crazy Turn Left	Intra-on	I Slice	84.75	84.75	84.75
		P Slice	582.36	665.44	594.95
		B Slice	0.00	0.00	0.00
		Total	<b>667.32</b>	<b>750.40</b>	<b>679.91</b>
	SNR Y (dB)		<b>40.95</b>	<b>40.81</b>	<b>40.91</b>
	Intra-off	I Slice	84.75	84.75	84.75
		P Slice	630.41	859.95	646.94
		B Slice	0.00	0.00	0.00
		Total	715.37	944.91	731.90
	SNR Y (dB)		40.87	40.56	40.74
Dancing Light	Intra-on	I Slice	94.61	94.61	94.61
		P Slice	808.23	886.27	832.26
		B Slice	0.00	0.00	0.00
		Total	<b>903.05</b>	<b>981.09</b>	<b>927.08</b>
	SNR Y (dB)		<b>38.50</b>	<b>38.49</b>	<b>38.47</b>
	Intra-off	I Slice	94.61	94.61	94.61
		P Slice	1058.33	1180.74	960.85
		B Slice	0.00	0.00	0.00
		Total	1153.15	1275.56	1055.67
	SNR Y (dB)		38.46	38.36	38.32
Intern On Bike	Intra-on	I Slice	68.03	68.03	68.03
		P Slice	304.02	315.85	308.73
		B Slice	0.00	0.00	0.00
		Total	<b>372.26</b>	<b>384.09</b>	<b>376.97</b>
	SNR Y (dB)		<b>41.70</b>	<b>41.60</b>	<b>41.61</b>
	Intra-off	I Slice	68.03	68.03	68.03
		P Slice	330.35	341.11	336.24
		B Slice	0.00	0.00	0.00
		Total	398.59	409.35	404.48
	SNR Y (dB)		41.67	41.44	41.48

With the use of H.264/AVC Baseline Profile, there is no B-frame in the encoded sequence. When the four video sequences were encoded using Main Profile with Intra mode enabled, and with one B-frame inserted between two P-frames, the resulting frame sequence was IBPB. The average bit rate and SNR are shown in Table 2-2. All the four video sequences show a lower average bitrate with small loss of SNR when Intra mode is enabled. While the SNR losses are less than 0.5dB for all sequences, the average bitrate is reduced by at least 14%. Therefore, for low bitrate H.264/AVC video encoding, the intra-mode and B-frame insertion should be enabled.

Table 2-2: Average bitrate of video sequences with intra mode enabled. The video sequences were encoded with H.264/AVC Baseline Profile and Main Profile with EPZS search algorithm

Sequence		Search Algorithm	
		EPZS @ H.264 Main Profile	EPZS @ H.264 Baseline Profile
Construction site	I Slice Bitrate (kb/s)	199.97	197.11
	P Slice Bitrate (kb/s)	1292.30	2137.77
	B Slice Bitrate (kb/s)	437.63	0.00
	Total Bitrate (kb/s)	<b>1930.12</b>	2335.09
	SNR Y (dB)	37.04	<b>37.59</b>
Crazy Turn Left	I Slice Bitrate (kb/s)	139.34	84.75
	P Slice Bitrate (kb/s)	104.95	594.95
	B Slice Bitrate (kb/s)	2.51	0.00
	Total Bitrate (kb/s)	<b>247.02</b>	679.91
	SNR Y (dB)	39.66	<b>40.91</b>
Dancing Light	I Slice Bitrate (kb/s)	96.42	94.61
	P Slice Bitrate (kb/s)	615.71	832.26
	B Slice Bitrate (kb/s)	94.67	0.00
	Total Bitrate (kb/s)	<b>807.02</b>	927.08
	SNR Y (dB)	38.19	<b>38.47</b>
Intern On Bike	I Slice Bitrate (kb/s)	69.16	68.03
	P Slice Bitrate (kb/s)	213.62	308.73
	B Slice Bitrate (kb/s)	37.93	0.00
	Total Bitrate (kb/s)	<b>320.93</b>	376.97
	SNR Y (dB)	41.43	<b>41.61</b>

## 2.4.2 Evaluation of Motion Vector

The MVs around the moving objects in the selected frames are examined. Figure 2-4 and Figure 2-5 show the MVs of macroblocks in the H.264/AVC encoded video streams. Green grid lines in the images in Figures are the macroblock boundaries. The

amplitude and direction of motion vectors for each macroblock are represented by the length of the green line from the centre of each macroblock. There are macroblocks shown with either green or red boundaries, representing macroblocks encoded in inter-frame or intra-frame mode respectively.

For Construction Site and Intern on Bike video sequences, the observing vehicle is moving on a straight flat road. It is expected that most of the MVs of macroblocks for static objects should apparently be emerging from the area near the centre of the image. However, as observed from Figure 2-4(b) and Figure 2-5(h) for the two sequences mentioned, there were many MV outliers pointing irregularly, especially on the road region where the texture was weak. Since the primary goal of the motion estimation algorithm for H.264/AVC encoder is to reduce the amount of information between successive frames, the resulting motion vectors are not necessarily representing the true motion of the object. When a particular macroblock has a weak texture, there is the aperture problem (Trucco and Verri, 1998) resulting in the deviation of MVs from the true motion. Also, the change in lighting conditions and the Rate-Distortion parameter will affect the block matching result of the motion estimation algorithm, leading to estimation error.

Sequence	Motion Search Algorithm (H.264 Baseline Profile)		
	Full-search	UM Hexagon	EPZS
(a) Construction site (Intra-on)			
(b) Construction site (Intra-off)			
(c) Crazy Turn Left (Intra-on)			
(d) Crazy Turn Left (Intra-off)			

Figure 2-4: An overview of H.264 encoded snapshot of 2 video sequences. Each video sequence is encoded with intra-mode on and off, using three different motion search algorithms. The red and green boxes shown in each encoded snapshot are macroblocks encoded in intra-mode and inter-mode respectively. The green lines shown at each macroblock indicates the amplitude and direction of the motion vector.

It is also noticed that the road surface usually has a smooth texture. The motion estimation algorithm will simply use SKIP mode to represent the motion of these macroblocks. This is erroneous to the true motion although it does not affect the coding efficiency significantly.



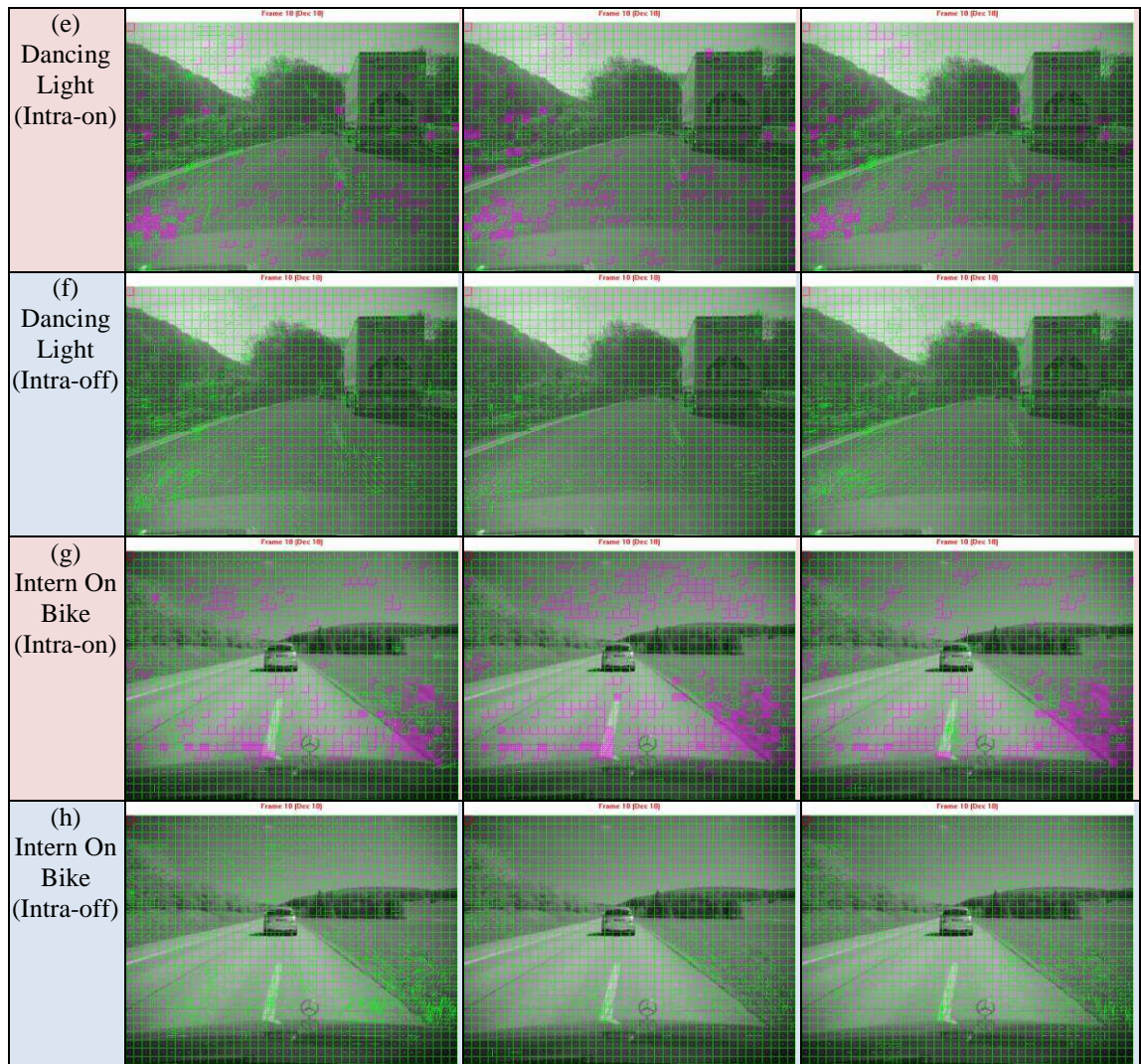


Figure 2-5: An overview of H.264 encoded snapshot of another 2 video sequences. Each video sequence is encoded with intra-mode on and off, using three different motion search algorithms. The red and green boxes shown in each encoded snapshot are macroblocks encoded in intra-mode and inter-mode respectively. The green lines shown at each macroblock indicates the amplitude and direction of the motion vector.

When examining more closely to the areas near the vehicles at the front as shown in Figure 2-6 and Figure 2-7, it was found that the mode used for encoding a particular macroblock was not necessarily the same when different motion estimation algorithm was used. When the vehicle at the front is far away from the camera, its size in the image is smaller. Also, since the relative motion between the camera and the vehicle at the front is smaller than that between the camera and background stationary objects, the encoder tends to encode the area of the front vehicle with larger macroblocks, such as 16x16, 8x16 and 16x8. For the Crazy Turn Left sequence shown in Figure 2-4,

there are more MVs around the vehicle at the front due to smaller partitions were used to represent the motion of the vehicle relative to the movement of the camera. Some of the macroblocks of relative slow moving objects can possibly be encoded in Intra-mode, leading to a loss of motion information of the moving object. When an object is far away from the camera, the number of inter-mode encoded macroblocks may be too small to determine whether it is a moving object, or is a static object.

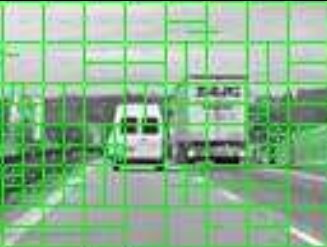
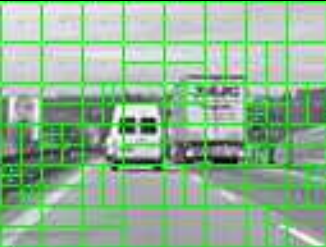
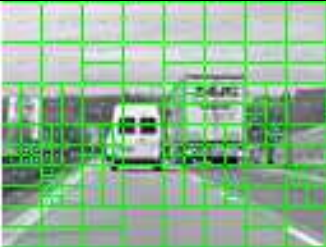
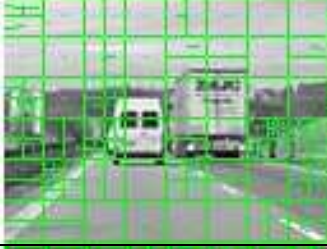

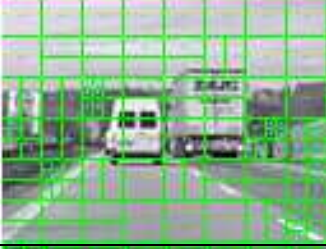



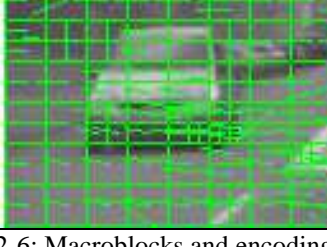


Sequence	Motion Search Algorithm		
	Full-search	UM Hexagon	EPZS
(a) Construction site (Intra-on)			
(b) Construction site (Intra-off)			
(c) Crazy Turn Left (Intra-on)			
(d) Crazy Turn Left (Intra-off)			

Figure 2-6: Macroblocks and encoding mode for different Daimler video sequences. Sequence Construction site shows small number of macroblocks to represent the movement of front vehicles. Sequence Crazy Turn Left shows MVs due to the left turn action of the subject vehicle.



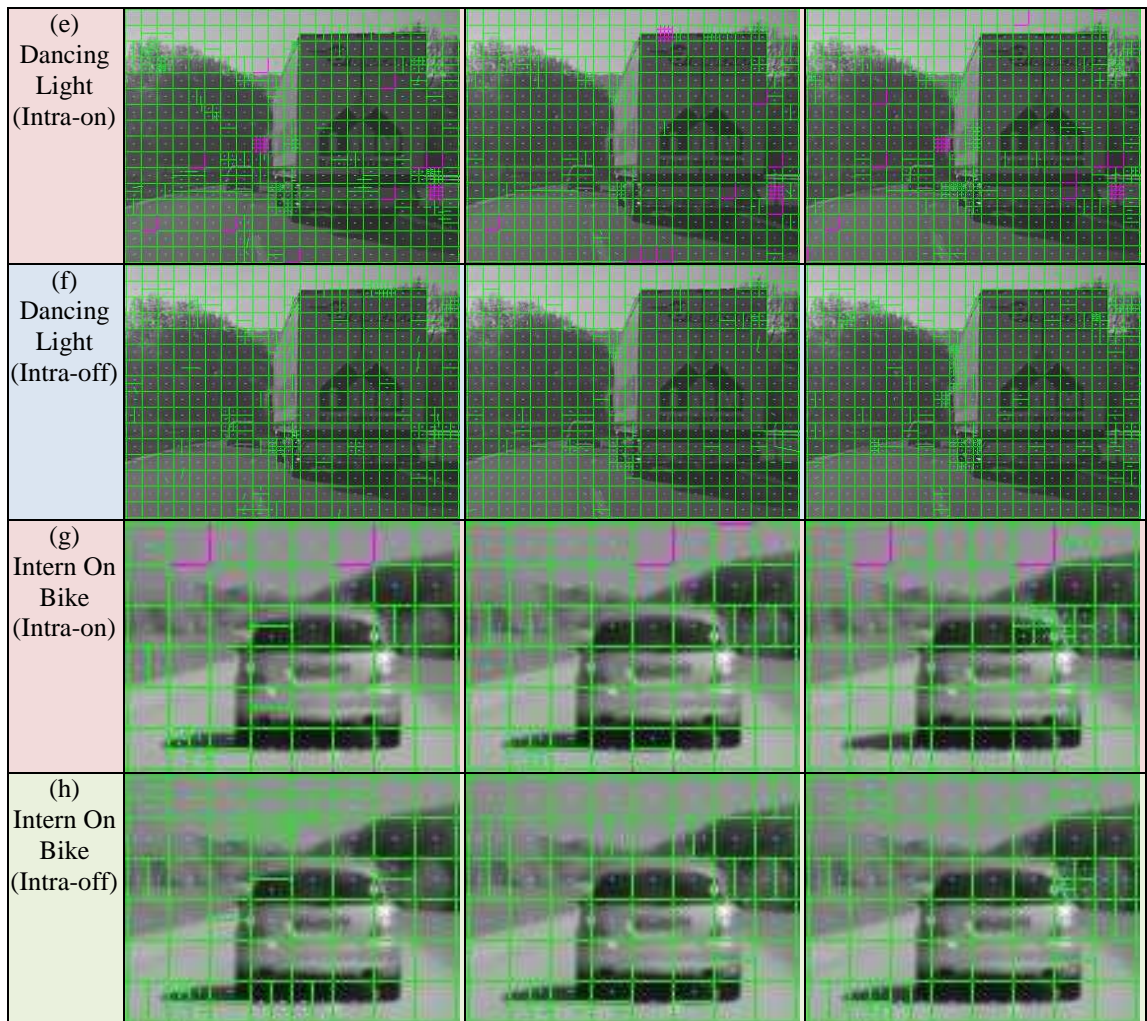


Figure 2-7: Macroblocks and encoding mode for different Daimler video sequences. Dancing Light and Intern On Bike sequences show small number of macroblocks to represent the movement of the front vehicles.

One of the important observations was that the MVs near the focus of expansion (FOE) and on relatively slow moving objects are small and not precise. Again, FOE is the point where objects in the scene are apparently emerging from when the camera is moving. This observation is summarised in the Section 2.4.3.

FOE is different from the principal point as well as the vanishing point of the camera. The Principal point refers to the point in the camera screen that corresponds to the optical axis of the camera. Vanishing point can be regarded as the point in the camera screen where two physical parallel lines, such as the railway, are merged due to the perspective view of the camera. The vanishing point is the same as the principal point

when the camera is installed with zero rotational angles. FOE is the same as the vanishing point when the camera is moving in straight line along the Z-axis. Assuming the camera is installed with zero rotational angles, the difference between FOE and the vanishing point (or the principal point in this case) can be expressed as Equation (2.6) (Trucco and Verri, 1998), where  $(x_0, y_0)$  is the FOE,  $(c_x, c_y)$  is the principal point,  $f$  is the focal length of the camera,  $V_x$  and  $V_z$  are the velocities of the camera in X- and Z-direction respectively.

$$\begin{aligned} x_0 &= c_x + V_x f / V_z \\ y_0 &= c_y + V_y f / V_z \end{aligned} \tag{2.6}$$

### 2.4.3 MVs Near FOE and on Relatively Slow Moving Objects

Figure 2-8 shows an image overlaid with two consecutive frames with frame interval of 0.33 seconds. Red lines in the image show the MVs of feature points found by Shi and Tomasi method (1994). Green lines in the image show the ideal optical flow field emerging from the FOE. Most of the MVs shown are pointing to the FOE although there are outliers due to the independently moving vehicle at the front and feature point tracking errors.

Another observation on the image shown in Figure 2-8 is that the MVs near the edges of the image have a relatively large amplitude compared to those near the centre of the image. Consider the independently moving vehicle at the front near the FOE with relatively slow speed to the observing camera, the amplitudes of MVs near the FOE are small. It is difficult to distinguish whether the MVs are the result of far static objects or the relative slow speed moving vehicle near the FOE.

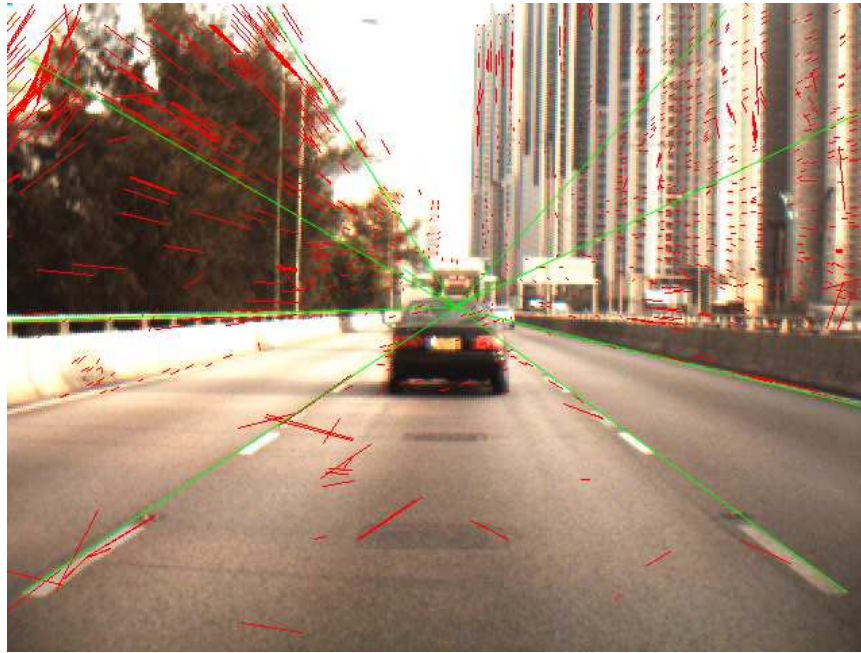


Figure 2-8: Image showing two overlaid consecutive images. Red lines show the optical flow field found by generic KLT feature point tracking algorithm. Green lines show the virtual optical flow field emerging from a point known as the FOE.

The MVs obtained by the Shi and Tomasi method are floating point vectors, whereas those obtained by H.264/AVC encoders are fixed point with up to a quarter pixel precision. Given the limited precision of MVs of the H.264/AVC encoder, the amplitudes of MVs of relatively slow moving objects, especially those near the FOE, are indistinguishable from the MVs of static objects. This observation is obvious in the “Intern-on-Bike sequence” where a P-frame encoded screenshot is shown in Figure 2-9. Some of the MVs of macroblocks on the road and the slow relative speed moving vehicle are highlighted with red stars and red circles. These MVs have the same or small amplitude difference. It is not possible to distinguish the moving object and static regions by examining the MVs alone.

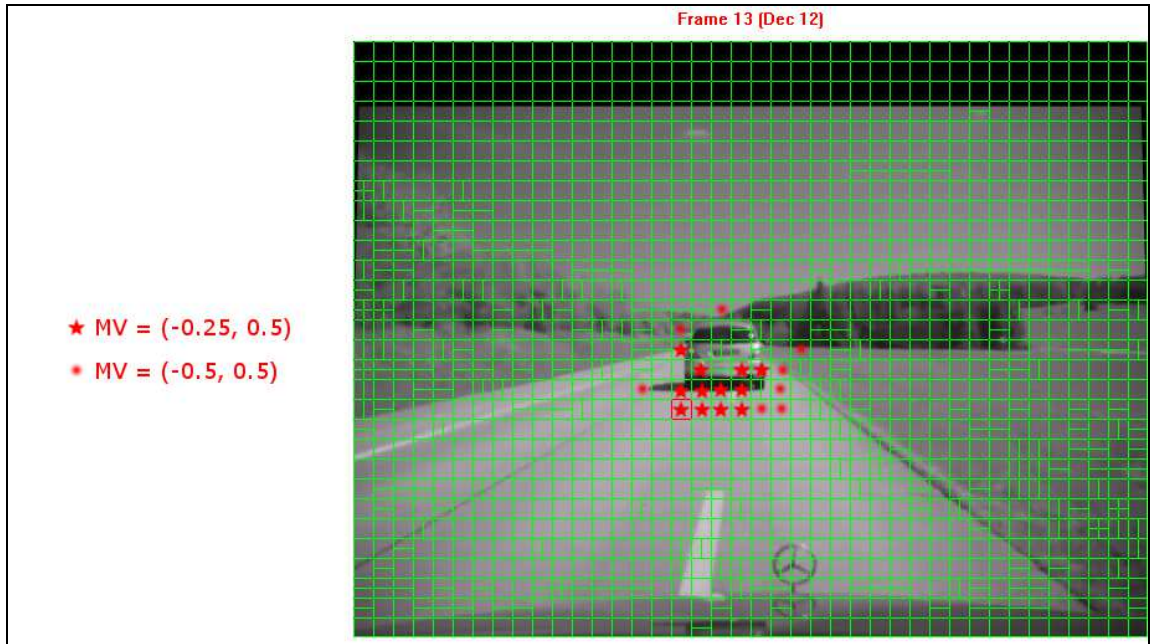


Figure 2-9: Selected MVs near the FOE on the road and on the slow relative speed moving vehicle. The amplitude of MVs is either identical or has small difference, making it difficult to distinguish moving objects and static regions.

## ***2.5 Road Region Detection***

The detection of both lane markings and moving objects on the road can result in false detections due to various reasons such as falsely recognised edge features, texts on the road, unreliable MVs from the H.264/AVC encoder and the chance of having regions with similarities to the features of moving vehicles.

The ever-changing scenarios on the road make both lane detection and moving object detection a challenging task. Knowing that the identification of road regions can help reduce the region of interest (ROI) for both lane detection and moving object detection, and that the relatively uniform colour and textures of road surfaces can provide a more generic set of features for recognition. The road region detection algorithm needs to handle situations with different objects on the road. It is also desirable to suppress the road detection error due to variations of illumination caused by shadows and reflections. It must also be computationally efficient for real-time applications.



For monocular vision based road detection algorithms, the road colour is the preferred feature to be analysed (He and Wang et al., 2004, Rotaru and Graf et al., 2008, Tan and Hong et al., 2006, Sotelo and Rodriguez et al., 2004).

For colour analysis, the popular colour space used is red-green-blue (RGB) (He and Wang et al., 2004, Tan and Hong et al., 2006) or hue-saturation-intensity (HSI) (Rotaru and Graf et al., 2008, Sotelo and Rodriguez et al., 2004). Since the HSI colour space has separate colour and light intensity components, there is less influence on colour recognition caused by the lighting variations of successive images (Ikonomakis and Plataniotis et al., 2000). However, the colour representation in HSI colour space is unreliable if the intensity component is too low. Therefore the colour analysis in dark and shadowed areas in an image is not satisfactory (Alvarez and x et al., 2011). For the use of RGB colour space for road region detection, there have been studies on the classification of road model by edge detection and perspective transformation (He and Wang et al., 2004), histogram of the colour space using green and blue channel only (Tan and Hong et al., 2006), illumination modelling by a mixture of Gaussian models (Lee and Crane, 2006, Ramstrom and Christensen, 2005). Ramstrom et al. have also reported a method with the combined use of different colour spaces, road shape models and multiple mixture of Gaussian models for improved road region detection on roads with no marking (Ramstrom and Christensen, 2005). However, how the number of Gaussians is selected and how the road models should be defined are trade-off questions for achieving good performance against different road conditions versus the computation cost.

Also, based on the study on shadow removal by the decomposition of an image into two separate images representing the variation in reflectance and the variation in illumination (Finlayson and Hordley et al., 2006), Alvarez et al. (2011) proposed a method that

combined the use of such shadow removal technique and a likelihood-based classifier using the normalised histogram of road region at the bottom part of the image in the HSI colour space. Similar to the shadow removal algorithm proposed by Finlayson et al. (2006), Alvarez et al.'s method requires a camera calibration procedure to estimate the "direction of illuminant variation". Alvarez et al.'s method reported a computation time of 600ms for images of resolution 640x480 using MATLAB code, and an estimated computation time of 40ms if the code were implemented in C++.

Nevertheless, all the methods proposed had cases with unsatisfactory road detection results. These cases include images with shadows, over-exposure, under-exposure, as well as interferences due to lane markings and worn-out roads. The ability of computing the road region in real-time remains a challenge.

## ***2.6 Practical Implications***

Recently, there have been many aftermarket products produced with a forward looking camera known as "Car Camera". They are mounted to the windshield for recording the environment during driving for security issues. Some examples of these Car Cameras are shown in Figure 2-10. These products are capable of performing real-time H.264/AVC video recording to the SD-card inserted into the cameras with resolution ranging from 640x480 (VGA) to 1920x1080 (HD).

For ADAS, a camera is also required to be mounted to the windshield. Instead of video recording, image processing algorithms are executed for objects and lane detection. Warning signal will be issued to alert the driver on the potential hazard.





Figure 2-10: Example of Aftermarket Car Cameras. (a) and (b) Built-in Infrared LEDs for night time illumination. (c) Inclined angle to fit more tightly to the windshield. (d) Movable lens for easier camera adjustment (source: <http://www.hktdc.com>).

One of the possible improvements for Car Cameras and ADAS is the shared-use of motion vectors for both video encoding and moving object detection. With the optical flow evaluation being replaced by the ME function from the video encoder, any additional computation induced by moving object detection can be reduced. This implies that the functions of Car Cameras can be enriched with ADAS functions without significantly increasing the hardware cost.

Although there have been studies on the recognition of moving object on a moving platform by using optical flow, there have been few studies on moving object detection on a moving platform using motion vectors (MVs) from the video encoder.

According to a paper that discussed the marketing strategies of Chinese automotive brands (Yan and Xu 2012), the image of Chinese automotive brands are weak. Even though the Chinese auto brands are moving upward to the manufacture of more luxury cars, Advanced Driver Assistance Systems such as LDWS and FCWS are still lacking. Chinese automotive brands are still facing high pressure on the cost and the provision of

feature rich functions to compete in the China market. The idea of shared-use of motion vectors for video coding and moving object detection can be one of the approaches to provide feature-rich ADAS function at a lower cost. According to Zhang et al. (2014), 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 optical flow evaluation.

## **2.7 Research Focus**

With the literature review performed, it was found that the algorithms and techniques for object detection for ADAS application and for video coding are advancing rapidly.

There have been studies on feature based vehicle detection methods. These methods are able to achieve a high true-positive detection rate. However, each viewing perspective of the vehicle requires a set of feature and classifier for successful detection. This implies that a larger number of samples is required for training the classifier for each viewing perspective. Also, there is an additional computational cost for detecting vehicles at each additional perspective, implying increasing hardware costs to cope with the additional detection.

In this regard, the non-parametric moving object detection based on the use of optical flow fields is favourable for its capability on detection without prior knowledge of those objects. However, the high computational cost for optical flow field and ego-motion estimation using the optical flow results is unfavourable for real-time embedded applications.

Another observation is on the problem and computational cost for ego-motion estimation based on optical flow fields. The ego-motion parameters are required to be evaluated frame-by-frame so that ego-motion compensation can be done in each successive frame. These methods also require reliable flow fields from successive frames on static objects or road surfaces. The availability of these flow fields can be a problem in case that the image is occupied mostly by moving objects, or that there is significant change in the lighting conditions. Therefore, the research direction is to make use of on-board inertial sensors to estimate the ego-motion parameters, targeting to more reliable ego-motion estimation and offload the computational resources from the embedded processor.

On the other hand, it is observed that the motion vectors from a H.264/AVC video encoder may be able to replace optical flow fields for non-parametric moving detection. H.264/AVC video encoders are widely used in consumer products and are readily available from off-the-shelf semiconductor chips.

Although there have been studies in making use of motion vectors of the H.264/AVC video stream for moving object detection in the decoder side, it was found that there are few studies related to combining the algorithms for ADAS and video coding in the encoder side so that more resources of the embedded microprocessor can be shared without a significant impact on the video coding efficiency and object detection accuracy.

The block diagram shown in Figure 2-11 illustrates the ADAS and H.264/AVC video recorder 2-in-1 system with shared use of functional blocks. It shows the shared motion estimation function and the shared motion vectors. By sharing the motion estimation block, the most time consuming and processor demanding part of the system can be combined. Successful combination gives a significant saving in computational cost to

achieve real-time performance. This also means a significant saving in the hardware cost.

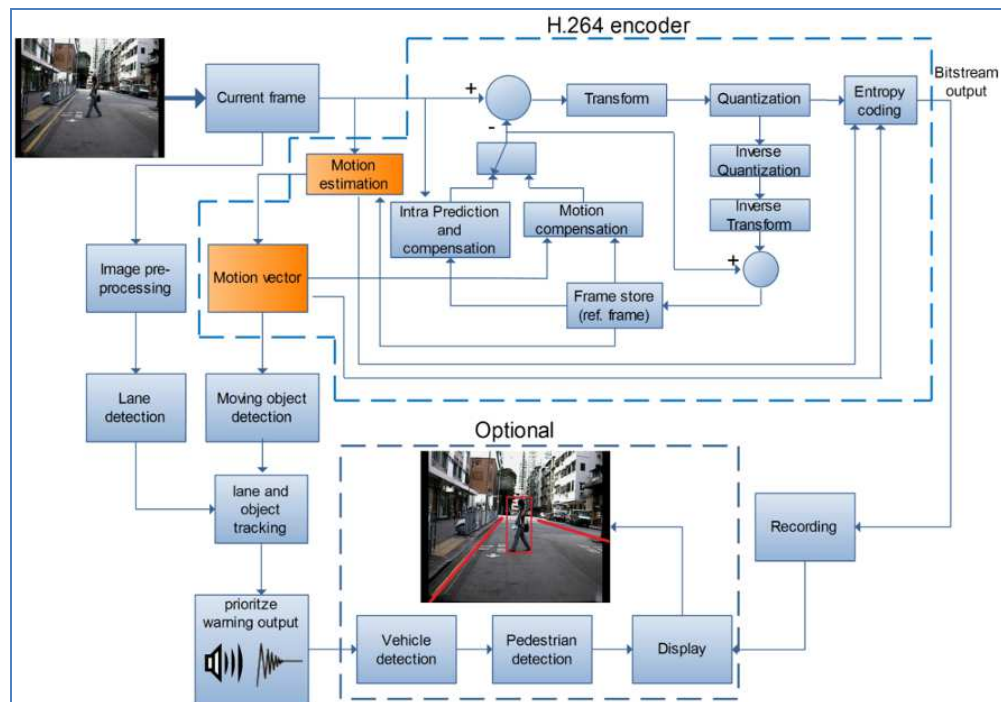


Figure 2-11: ADAS and H.264/AVC video encoding 2-in-1 system with shared functional blocks. The block diagram shows the motion estimation and motion vector functional blocks are shared for the use of video coding and moving object detection.

Therefore, the research focus was to combine the motion vector estimation for H.264/AVC video encoding and moving object detection. The technical challenge was on the research and development of suitable algorithms to perform both moving object detection and video coding functions in real-time. The resulting system required a balanced performance of moving object detection, video coding efficiency and video quality.

## **2.8 Chapter Summary**

This Chapter has reviewed the methods commonly used for vehicle detection. These methods can be classified as feature based, statistical based and optical-flow based methods. It has also described the problems with the use of H.264/AVC MVs for moving object detection. In particular, the MVs from H.264/AVC coding have limited precision of up to 1/4 pixel only, and the amplitudes on relatively slow speed moving objects are small, leading to large ego motion compensation error. Also, the design goal of the motion estimation algorithm for H.264/AVC video coding is for the best possible video compression rather than the accuracy of movements of objects. All these problems hindered the use of the MVs from a H.264/AVC encoder for moving object detection on a moving platform.

This Chapter also covered literature review on techniques associated to MV based moving object detection. This includes planar parallax evaluation, ego motion estimation and road region detection. Based on the literature review performed, it is confirmed that there can be more in-depth research on the shared use of MVs from a H.264/AVC encoder for moving object detection, leading to the research focus of this project.

### 3 Algorithm Framework

Based on the literature review and a series of experiments, an algorithm framework for moving object detection with the shared use of MVs from a H.264/AVC encoder is proposed. Figure 3-1 shows the major functional blocks of the proposed algorithm framework.

The inputs to the system consist of a camera with a six-degree-of-freedom inertial sensor mounted directly to the camera sensor board, and the vehicle speed signal from the vehicle speed sensor. They output dynamic parameters which include the 3-axis rotation angle, acceleration and angular speed, as well as the vehicle speed. The vehicle speed sensor outputs square pulses when the vehicle is moving. The number of square pulses is proportional to the moving speed of the vehicle. These dynamic parameters are used for ego motion estimation and FOE estimation. By the use of inertial sensors and speed sensors for ego motion estimation, uncertainties on the quantity and quality of good feature points for image based ego motion estimation can be eliminated. The computation cost can also be reduced significantly.

The camera captures successive colour images, feeding to the H.264/AVC encoder for video recording. During the encoding process, MVs are generated. The colour images are also used for road region detection. The result of road region detection is used for reducing the region of interest (ROI) for moving vehicle detection so that the computation can be achieved in a shorter time. The ROI is further reduced by using the results from MV output and FOE estimation. By reducing the ROI, the computation time can be reduced as only the areas that potential have moving objects will be processed.

The proposed algorithm has divided the moving objects into two categories; relatively fast and relatively slow moving objects respectively. Different algorithms are proposed

to detect moving objects in these two categories. This approach is to supplement the erroneous and imprecise MVs on relatively slow moving objects, so that the detection rate of these objects can be guaranteed. In the mean time, those objects that move relatively fast can be taken care of by the relatively fast speed moving object detection algorithm. After successful detection of moving objects, a tracking algorithm is applied to reduce the frame-to-frame computation time.

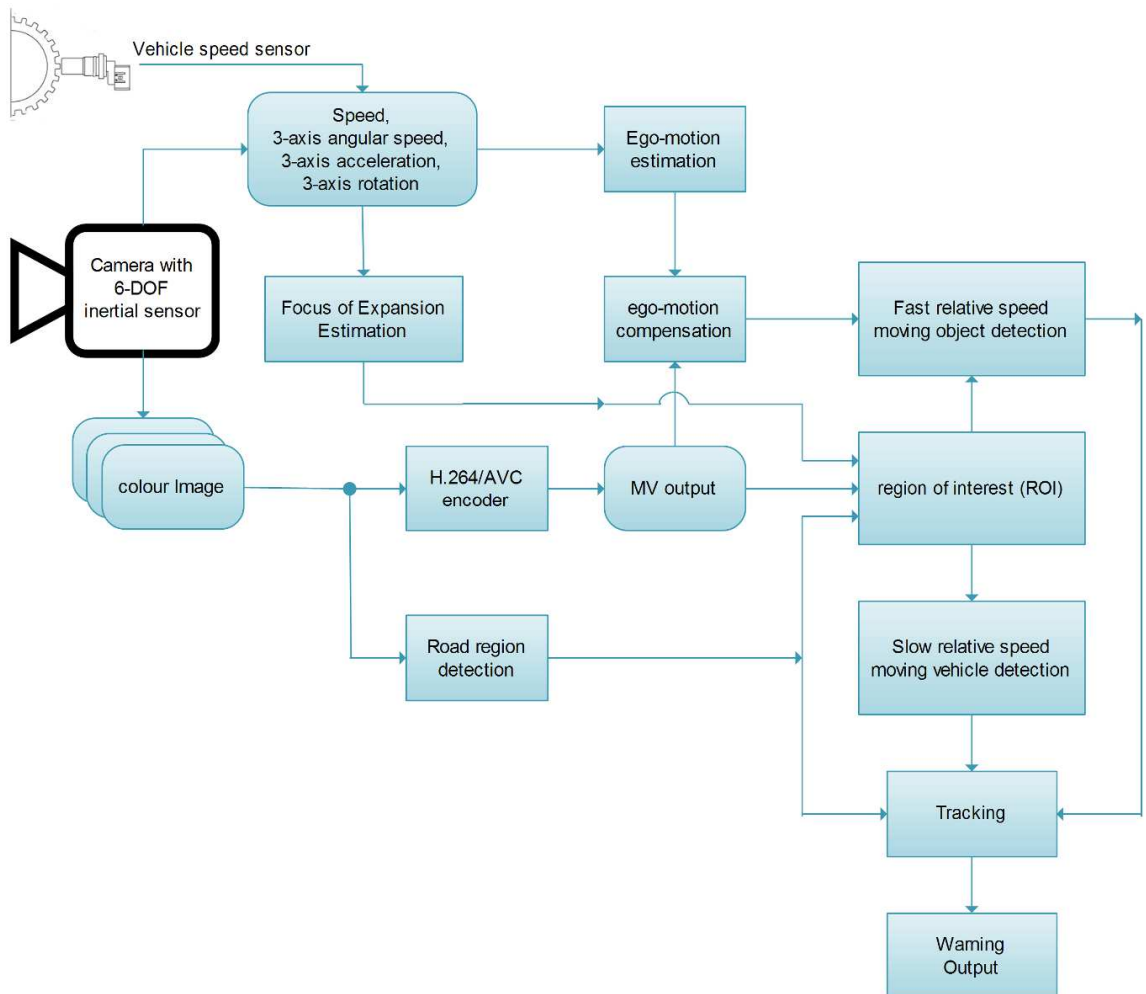


Figure 3-1: Major functional blocks of the proposed algorithm framework

### 3.1 System Preparation

#### 3.1.1 Configuration for Video Encoder

The proposed algorithm framework requires MVs from a H.264/AVC encoder. The JM18.4 open-source encoder (JVT, 2012) was chosen for modification to output the required MVs.

Since the MVs may have different block size, it is more convenient to unify the block size that each MV is representing. Off-the-shelf real-time H.264/AVC encoders usually sacrifice the minimum block size for motion estimation to 8x8 (TI, 2015a, Freescale, 2015) rather than the minimum 4x4 block size specified in H.264/AVC encoder, the block size of each MV was set to 8x8 during this project. That is, mode 5, 6 and 7 shown in Figure 2-3 are disabled. With these modes disabled, the block size of MVs in a frame will have the size of 8x8, 8x16, 16x8 and 16x16. Those MVs of block size larger 8x8 were regarded as multiple blocks of size 8x8 with the same MV value, as illustrated in Figure 3-2.

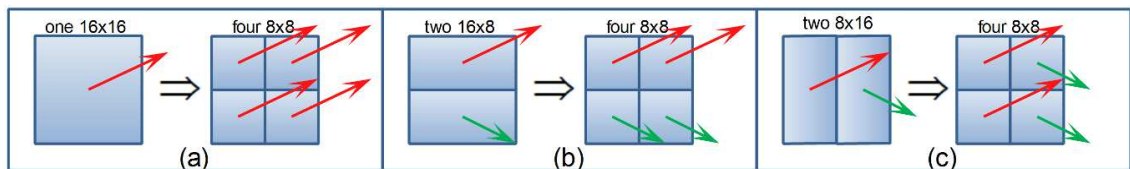


Figure 3-2: Transforming MV for different block size to represent the same block size of 8x8

During the motion estimation stage, the MVs were exported with block size of 8x8 for each inter-frame encoding process. Since blocks encoded in SKIP mode may represent weak texture blocks with small relative motion, MVs for these blocks were marked when SKIP mode was used.

The desired video frame rate is 30 frames per second (fps) or above due to the persistence of vision of human eyes. The time interval between successive frames is



therefore 33ms. This time interval is too short to give sufficiently large MVs for moving objects on the road, especially those relatively slow moving objects. Therefore, the time interval between frames for motion estimation was increased by inserting one B-frame between every two P-frames. Consequently, the time interval between P-frames for MV output was 66ms, whereas the video frame rate was kept unchanged at 30fps. So, the resulting frame sequence is IBPBP for video encoding.

### **3.1.2 Camera Calibration**

The objects appearing in the screen are required to be related to their physical locations so that the size and distance of these objects can be estimated for detection confirmation. Therefore, the camera needs to be calibrated to relate the screen coordinates to the physical World coordinates.

#### **3.1.2.1 Definition of the Coordinate Systems**

Figure 3-3 illustrates the definitions of World coordinates, camera coordinates, and screen coordinates. All the coordinate systems are right-handed. The World coordinates  $W$  is with the  $X$ -,  $Y$ - and  $Z$ -axis shown on the ground plane. The camera coordinates  $C$  is with  $X_c$ -,  $Y_c$ - and  $Z_c$ -axis and there is a transformation from the World coordinate system to the camera coordinate system with rotation  $R_w$  and translation  $T_w$ .

The rotation about the  $X$ -,  $Y$ - and  $Z$ -axis is shown as  $\theta_x$ ,  $\theta_y$ , and  $\theta_z$  respectively.

The positive direction is defined as having clockwise rotation when looking from the origin of the respective axis.

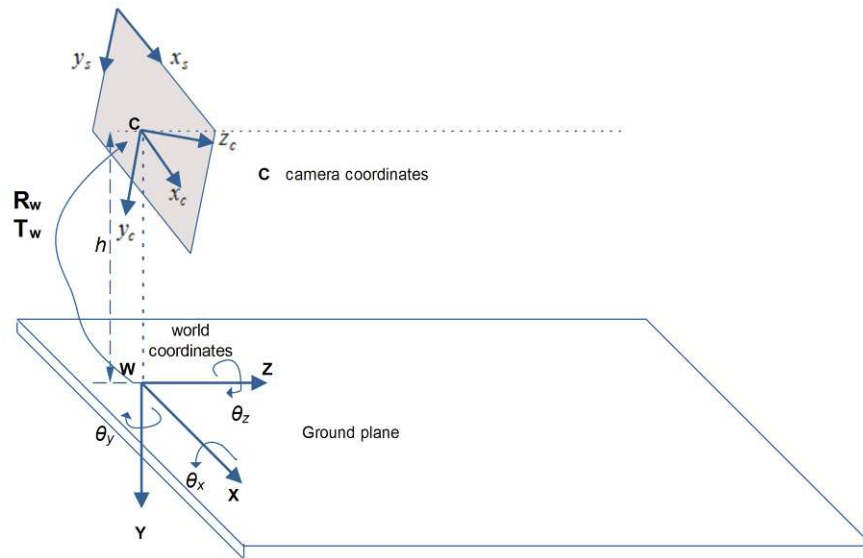


Figure 3-3: Illustration of World coordinates, camera coordinates, and screen coordinates. There are Rotation  $R_w$  and Translation  $T_w$  from the World coordinates to the camera coordinates. The screen coordinates start from the top left corner of an image.

The parameters describing the camera calibration include the intrinsic parameters and extrinsic parameters. Intrinsic parameters refer to the focal length, pixel size of the camera sensor, and the coordinates of the principal point. Extrinsic parameters refer to the height of the camera above the ground plane, pitch ( $\theta_x$ ), yaw ( $\theta_y$ ) and roll ( $\theta_z$ ) angles of the camera coordinates with respect to the ground plane.

Since there is only one camera in the system, the depth information or the distance between an object and the camera cannot be obtained directly from the two dimensional screen coordinates of the image. Instead, the distance of an object is estimated by the trigonometric calculation with reference to a flat surface, such as the ground plane. Figure 3-4 shows a camera with non-zero pitch angle  $\theta_x$ . The vanishing point on the screen can be regarded as the merging point of two parallel lines running along the Z-axis direction. If the pitch, roll and yaw angles of the camera are all zero, the optical axis is passing through the principal point, and aligning with the vanishing point.

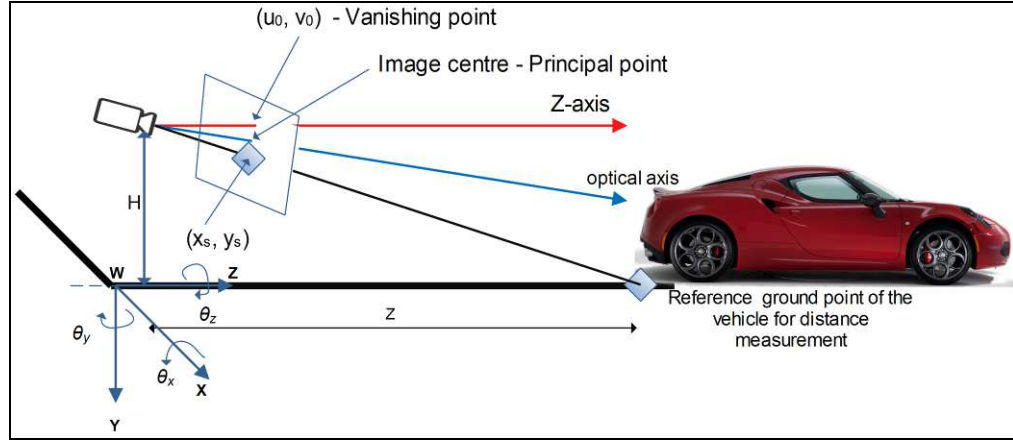


Figure 3-4: Camera with non-zero pitch angle  $\theta_x$ . The vanishing point on the screen is not aligned with the optical axis.

For a point  $P_w = [X_w \ Y_w \ Z_w]^T$  on the World coordinates, the corresponding point on the camera coordinates is  $p^c = [x_c \ y_c \ z_c]^T$ .  $p^s = [x_s \ y_s \ 1]^T$  is the corresponding point on the screen coordinates. The relationship between  $P_w$  and  $p^c$  can be expressed as Equation (3.1).

$$p^c = [x_c \ y_c \ z_c]^T = R_w(P_w - T_w) \quad (3.1)$$

The relationship between  $p^s$  and  $p^c$  can be expressed as Equation (3.2), where  $K$  is the intrinsic matrix of the camera. Therefore, Equation (3.3) can be derived by substituting Equation (3.1) to (3.2).

$$p^s = [x_s \ y_s \ 1]^T = Kp^c \quad (3.2)$$

$$p^s = [x_s \ y_s \ 1]^T = KR_w(P_w - T_w) \quad (3.3)$$

Equation (3.3) can further be written to Equation (3.4), where  $t = -R_w T_w$  and  $[R_w \ | \ t]$  is the 3x4 matrix representing the rotation and translation of the camera relative to the World coordinates.  $[R_w \ | \ t]$  is known as the extrinsic matrix of the camera.

$$p^s = [x_s \ y_s \ 1]^T = K[R_w \ | \ t][X_w \ Y_w \ Z_w \ 1]^T \quad (3.4)$$

The Intrinsic matrix  $K$  is shown in Equation (3.5), where  $f_x$  and  $f_y$  are the focal length of the camera along the  $x$ - and  $y$ -axis respectively,  $c_x$  and  $c_y$  are the principal points of the camera along the  $x$ - and  $y$ -axis respectively.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.5)$$

The extrinsic matrices  $R_w$  and  $T_w$  are shown in Equation (3.6) and (3.7) respectively, where  $R_x$ ,  $R_y$ , and  $R_z$  are the 3x3 rotational matrices about X-axis, Y-axis, and Z-axis respectively,  $\theta_x$ ,  $\theta_y$ , and  $\theta_z$  represent the pitch angle about the X-axis, yaw angle about the Y-axis, and the roll angle about the Z-axis respectively,  $h$  is the height of the camera above the ground plane,  $d$  is the horizontal distance along the Z-axis between the camera and the origin of the ground plane. Without loss of generality, the distance offset  $d$  along the Z-axis between the World coordinates and the camera coordinates is set to zero.

$$\begin{aligned} R_w &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} = R_x R_y R_z \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_y & 0 & \sin \theta_y \\ 0 & 1 & 0 \\ -\sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} \cos \theta_z & -\sin \theta_z & 0 \\ \sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} \cos \theta_y \cos \theta_z & -\cos \theta_y \sin \theta_z & \sin \theta_y \\ \sin \theta_z & \cos \theta_z & 0 \\ -\sin \theta_y \cos \theta_z & \sin \theta_y \sin \theta_z & \cos \theta_y \end{bmatrix} \quad (3.6) \\ &= \begin{bmatrix} c \theta_y c \theta_z & -c \theta_y s \theta_z & s \theta_y \\ c \theta_x s \theta_z + s \theta_x s \theta_y c \theta_z & c \theta_x c \theta_z - s \theta_x s \theta_y s \theta_z & -s \theta_x c \theta_y \\ s \theta_x s \theta_z - c \theta_x s \theta_y c \theta_z & s \theta_x c \theta_z + c \theta_x s \theta_y s \theta_z & c \theta_x c \theta_y \end{bmatrix} \\ &, \text{ where } c \text{ and } s \text{ denote cosine and sine functions respectively} \end{aligned}$$

$$\begin{aligned} T_w &= [0 \quad -h \quad d]^T \\ &= [0 \quad -h \quad 0]^T \text{ for } d=0 \end{aligned} \quad (3.7)$$

One point to note is that Equation (3.4) can be re-written as Equation (3.8), where the 3x4 matrix with element  $a_{ij}$  is the result of  $K[R_w \quad t]$ . Consider the case with  $Y_w=0$ , meaning that the corresponding coordinates are on the ground level, the 3x4 matrix

with element  $a_{ij}$  can be re-written to a 3x3 matrix shown in Equation (3.9). Equation (3.9) relates a point on the ground level in the World coordinates to a point on the screen. Since  $M$  is a 3x3 matrix, it can be inverted to  $M^{-1}$  so that a point in the screen can be related to a point on the ground surface in the World coordinates, as shown in Equation (3.10).

$$\begin{bmatrix} x_s & y_s & 1 \end{bmatrix}^T = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X_w & Y_w & Z_w & 1 \end{bmatrix}^T \quad (3.8)$$

$$\begin{aligned} \begin{bmatrix} x_s & y_s & 1 \end{bmatrix}^T &= \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X_w & 0 & Z_w & 1 \end{bmatrix}^T \\ &= \begin{bmatrix} a_{11} & a_{13} & a_{14} \\ a_{21} & a_{23} & a_{24} \\ a_{31} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Z_w \\ 1 \end{bmatrix} \\ &= M \begin{bmatrix} X_w & Z_w & 1 \end{bmatrix}^T \end{aligned} \quad (3.9)$$

$$\begin{bmatrix} X_w & Z_w & 1 \end{bmatrix}^T = M^{-1} \begin{bmatrix} x_s & y_s & 1 \end{bmatrix}^T \quad (3.10)$$

### 3.1.2.2 Calibration Method

There have been many camera calibration algorithms proposed. There are algorithms trying to estimate the camera parameters by 3-D reference objects (Tsai, 1987, Heikkila, 2000), 2-D planar objects (Zhang, 2000, Lucchese, 2005) and even 1-D objects (Zhang, 2004). Also, there have been attempts to estimate the camera parameters by methods known as auto-calibration or self-calibration (Civera and Davison et al., 2012, Mitsunaga and Nayar, 1999, Wang and Zhang et al., 2010, Pernek and Hajder, 2010). These auto-calibration methods try to use feature points from un-calibrated objects from multiple scenes. Auto-calibration methods require reliable feature points to get satisfactory results. This requirement cannot be guaranteed in the ever-changing environment on the road for automotive application.

The method proposed by Zhang (2000) is one of the most popular calibration algorithms. It uses a planar checker board that can be prepared by simply printing the pattern on a paper. The checker board is then presented in front of a camera in different orientations for capturing multiple images. Since the checker board is a planar object, the algorithm always assumes all the identified feature points has zero value in one of the axes, such as  $Z_w$  of the World coordinates.

Therefore, Equation (3.4) can be simplified to Equation (3.11), where  $r_1$  and  $r_2$  are the first and second column of the  $R_w$  matrix in Equation (3.4), and  $H = K [r_1 \ r_2 \ t]$  is known as the planar homography matrix. By substituting all identified feature points in the screen and World coordinates in each view to Equation (3.11),  $H$  can be estimated from the over-determined system by least-square optimisation method.

$$\begin{aligned} [x_s \ y_s \ 1]^T &= K [r_1 \ r_2 \ t] [X_w \ Y_w \ 1]^T \\ &= H [X_w \ Y_w \ 1]^T \end{aligned} \quad (3.11)$$

In order to minimise the computational burden due to camera installation problems, the camera was installed so that the rotation angles in the extrinsic matrix were all zero. Otherwise, sine and cosine calculations in the extrinsic matrix would introduce additional computational cost to the system, hindering the real-time performance. Zhang's camera calibration method can only help determine the intrinsic and extrinsic parameters of the camera after installation, it does not provide a method directly for camera installation. Therefore, a new camera installation method is proposed.

The camera module used in this project was designed with a six-degree-of-freedom inertial sensor on-board, meaning that the pitch, roll and yaw angles of the camera can be measured accurately. The pitch and roll angle readings from the inertial sensor are used during the camera installation so that the pitch and roll angle are zero. The yaw

angle of the camera with respect to the vehicle body cannot be determined by the yaw angle reading, an external checker pattern was used to ensure the proper installation.

### Step 1: Focal Length Estimation

The focal length was estimated before the camera was installed into the vehicle. This is because the distance of the camera to a defined object is difficult to measure after it is installed in a vehicle. Figure 3-5 shows the setup for estimating the focal lengths of the camera. It includes a checker board pattern printing on a large piece of paper, an up-right sign board, a laser distance checker and a PC program for capturing the image from the camera. The distance between the up-right board and the camera was measured by a laser distance checker. The distance difference of the camera to the left-side of the up-right board should be close to that to the right-side of the up-right board. This made sure that the up-right board is not skewed to either side of the camera, minimising the distance measurement error.

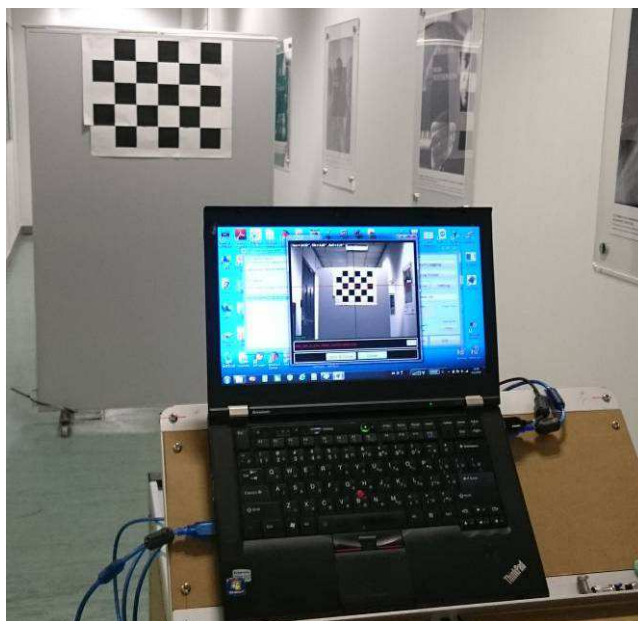


Figure 3-5: Calibration Setup for focal lengths of the intrinsic parameter of the camera

A simple pin-hole camera model was used for the distance estimation. Figure 3-6 shows the pin-hole camera model with screen coordinates shown in red, and the World

coordinates shown in blue. The green line in Figure 3-6(b) represents the location of the camera, with focal length  $f$  from the origin. For a point  $P_w$  in the World coordinates  $[X_w Y_w Z_w]^T$ , its corresponding point  $p_s$  in the screen coordinates at  $[x_s y_s f]^T$  can be estimated by the simple trigonometry. The relationship is expressed in Equation (3.12), where  $(c_x, c_y)$  are the principal point of the camera, indicating that there is an offset between the screen coordinates and the World coordinates. The principal point  $(c_x, c_y)$  is assumed to be at the centre of the screen.

$$\begin{cases} x_s = f_x X_w / Z_w + c_x \\ y_s = f_y Y_w / Z_w + c_y \end{cases} \quad (3.12)$$

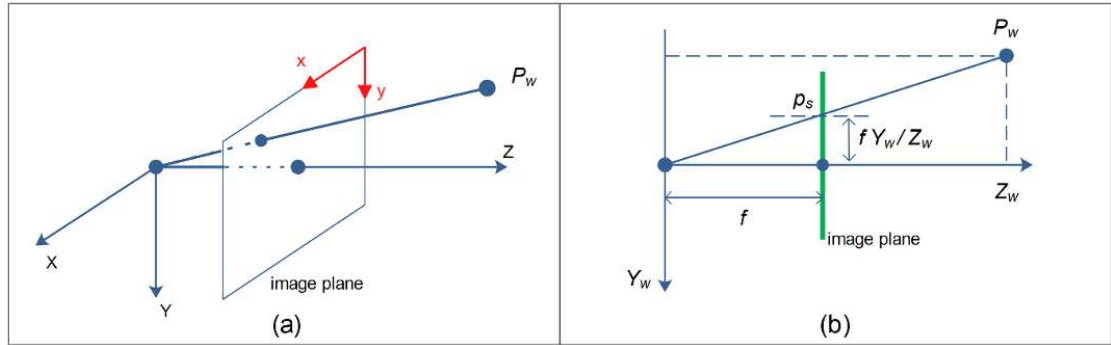


Figure 3-6: Simple pin-hole camera model. (a) Definition of camera coordinates system and the screen coordinates system. (b) Viewing from X-axis to the origin with the green line representing the image plane

For the checker pattern on the up-right board, the screen coordinates of the three control points A, B and C, shown in Figure 3-7, were initially selected manually via the computer program. The sub-pixel corner of each selected point is found by searching around the selected point with a search window of size 5x5. According to the geometrical relationship of a simple camera model, the focal lengths  $f_x$  and  $f_y$  can then be evaluated by Equation (3.13), where  $V_d$  and  $H_d$  are the distances in number of pixels between control point A and C, and A and B respectively,  $D_{ab}$  and  $D_{ac}$  are the physical distance between the control point A and B, and A and C respectively.

$$f_x = \frac{H_d Z_w}{D_{ab}}, \quad f_y = \frac{V_d Z_w}{D_{ac}} \quad (3.13)$$



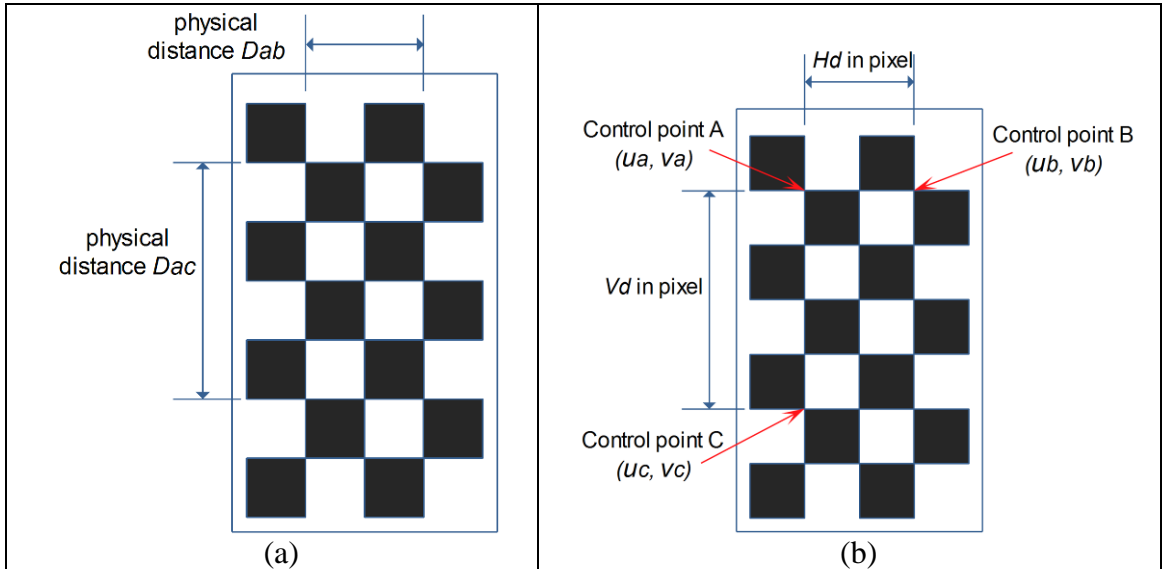


Figure 3-7: Checker pattern on the up-right board. (a) Physical dimension of the checker pattern. (b) Coordinates and distance between control points in number of pixels.

After the focal lengths  $f_x$  and  $f_y$  were estimated. More control points could be selected to check against the value of  $f_x$  and  $f_y$ . If the deviation between these sets of values was small, the values of  $f_x$  and  $f_y$  were accepted.

### Step 2: Finding the Centre Line

To physically install the camera to the vehicle with zero rotation angles, a simple method that makes use of the centre line of the vehicle is proposed. Referring to Figure 3-9,  $A_L$  and  $A_R$  are the locations of two easily identifiable points that are symmetric about the centre line  $L_C$ . They can be the corners between the front bumper and the front quarter panels or the left and right corners of the engine hood near the head lamps, or similar identifiable points, as illustrated in Figure 3-8. These points are chosen to minimise potential miss-location arising from the uncertainty of featured locations influencing the calibration result.



Figure 3-8: Illustration of good symmetric positions for symmetric geometrical line construction for camera calibration. The square boxes shown on the left and right side of the vehicle indicate example locations for good symmetrical geometrical line construction.

With reference to Figure 3-9, the checker banner was placed on the ground without initially being aligned to the centre line  $L_C$  of the vehicle. This was because the centre line was not known until the markers  $B_C$ ,  $C_C$  and  $D_C$  were identified. Two non-elastic ropes of equal length are used to make two straight line segments from  $A_L$  to  $B_C$  and from  $A_R$  to  $B_C$  respectively. The meeting point  $B_C$  of these two lines is marked with a pin. The banner was moved so that the pin could be fixed to a corner of the checkers, such as at  $P_1$  on the banner.

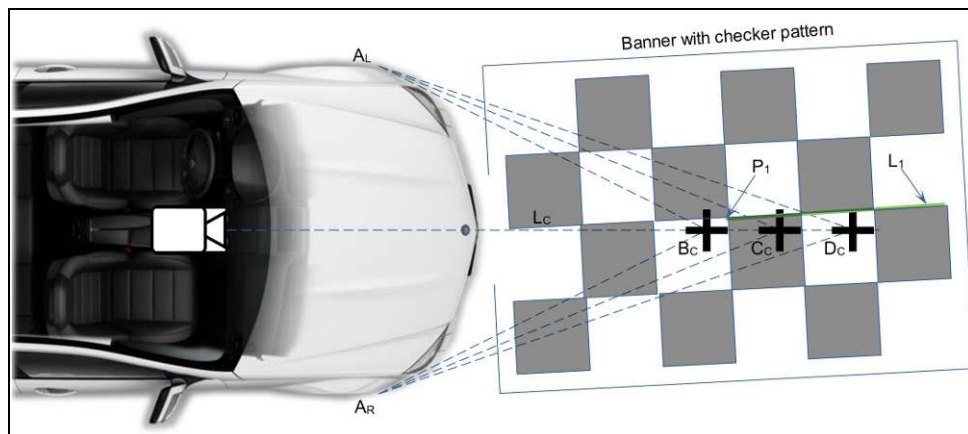


Figure 3-9: Illustration of alignment markings for the centre line for camera installation. The banner with checker pattern is not aligned to the centre line until the centre line is found.

Another pair of ropes of equal length that was longer than the length from  $A_R$  to  $B_C$  was used to mark the pin point  $C_C$ . Since the lengths of these two ropes were the same, the pin point  $C_C$  should lie on the same line  $L_C$ . The banner was moved again so that the pin point  $C_C$  is fixed at a position along the line  $L_1$  on the banner. Similarly, pin

point  $D_C$  was marked with another pair of ropes and the banner position was refined so that all the three identified pin points  $B_C$ ,  $C_C$  and  $D_C$  are fixed on the straight line  $L_1$ . After this procedure, the line  $L_1$  was aligned with the centre line  $L_C$ . Therefore, the centre line of the vehicle  $L_C$  was found, as shown in Figure 3-10.

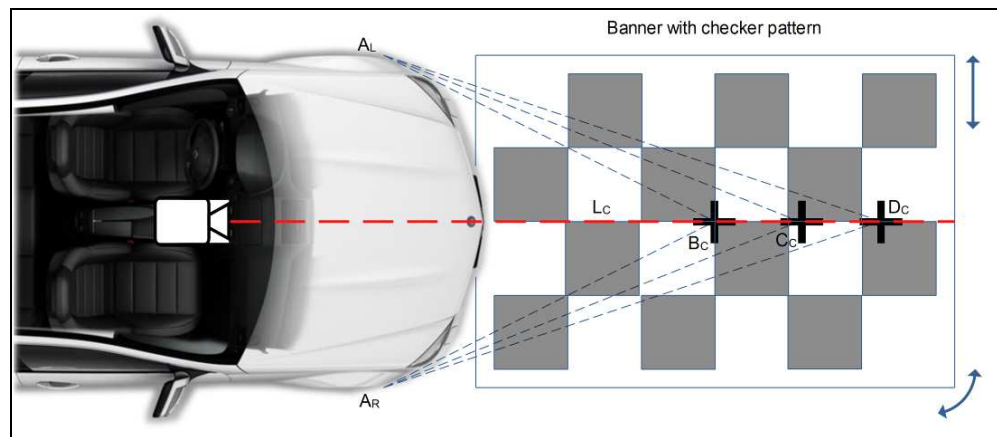


Figure 3-10: The camera installation process with the check-board banner aligned to the centre line  $L_C$  of the vehicle

### Step 3: Alignment to Zero Rotational Angle

After the centre line of the vehicle was found, the installation of the camera is continued. Figure 3-11 shows the example snapshot of the real-time video captured from the camera. Two perpendicular lines, one is drawing horizontal in red and the other is drawing vertically in orange, was overlaid onto the screen by the calibration software. When the camera was not installed properly at the beginning, the horizontal and vertical lines formed by the checker pattern on the banner placing on the level ground were not aligned with the overlaid horizontal and vertical lines.

The camera was moved to a location where the orange line was aligned with the centre line  $L_C$  on the checker pattern banner. The yaw angle of the camera with respect to the vehicle body is zero when the lines are aligned.

Similarly, the position of the camera was refined so that the red horizontal line is aligned with a horizontal line formed by the checker pattern. The position of the red line on the screen was adjustable by the calibration software to facilitate the alignment. When both the orange vertical line and the red horizontal line were aligned to the checker pattern, the camera was installed with zero roll and yaw angles.

Finally, the pitch angle of the camera was checked by the on-board sensor reading of the camera. By adjusting the pitch angle of the camera until the pitch angle reading reaches zero, the installation is completed.

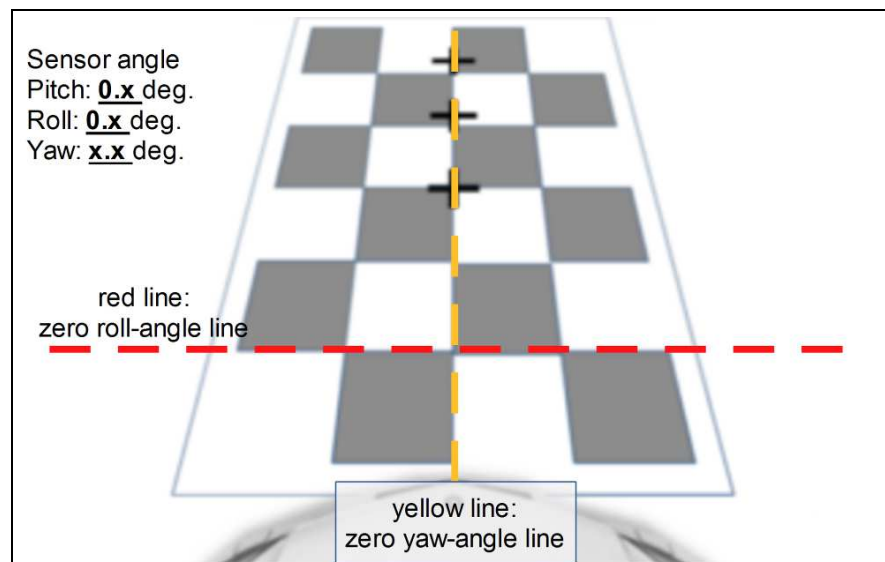


Figure 3-11: An example video display with the scene captured by the camera to be installed. It sees the banner with checker pattern on the level ground. An orange line is overlaid in the centre of the screen to indicate the line with zero yaw angle. A red horizontal line is also overlaid at the bottom of the screen to indicate the zero roll angle. The orange and red line should be aligned vertically and horizontally respectively to a line formed by the checker pattern on the banner.

#### Step 4: Camera Height Estimation

The installation height of the camera was estimated by using the banner with checker pattern placing on the level ground that was well aligned as described in Step 2 and Step 3. Figure 3-12 shows the configuration for camera height calibration based on the checker pattern placing on the level ground with the camera installation at zero rotational angles. One very important point to note is that the distance  $Z_l$  from the

camera to the checker pattern does not need to be measured, ensuring more accurate calibration result by eliminating the measurement error of  $Z_l$ . The screen coordinates of the control points  $A_c$ ,  $B_c$ ,  $P_1$ ,  $P_2$ ,  $P_3$  and  $P_4$  on the screen are first manually selected. Then the exact coordinates are refined to sub-pixel resolution. These control points result in a set of Equation shown in Equation (3.14) to (3.19). The left side of these equations are all available from the screen coordinates.

$$\text{Point } A_c \quad y_a = f_y h / Z_1 + c_y \quad (3.14)$$

$$\text{Point } B_c \quad y_b = f_y h / (Z_1 + d) + c_y \quad (3.15)$$

$$\text{Point } P_1 \quad x_1 = f_x d / Z_1 + c_x \quad (3.16)$$

$$\text{Point } P_2 \quad x_2 = -f_x d / Z_1 + c_x \quad (3.17)$$

$$\text{Point } P_3 \quad x_3 = f_x d / (Z_1 + d) + c_x \quad (3.18)$$

$$\text{Point } P_4 \quad x_4 = -f_x d / (Z_1 + d) + c_x \quad (3.19)$$

The difference of Equation (3.16) and (3.17) yields Equation (3.20). Similarly, the difference of Equation (3.18) and (3.19) yields Equation (3.21), and Equation (3.14) and (3.15) yields Equation (3.22).

$$\begin{aligned} x_1 - x_2 &= 2f_x d / Z_1 \\ \Rightarrow f_x / Z_1 &= (x_1 - x_2) / (2d) \end{aligned} \quad (3.20)$$

$$\begin{aligned} x_3 - x_4 &= 2f_x d / (Z_1 + d) \\ \Rightarrow f_x / (Z_1 + d) &= (x_3 - x_4) / (2d) \end{aligned} \quad (3.21)$$

$$\begin{aligned} y_a - y_b &= f_y h (1/Z_1 - 1/(Z_1 + d)) \\ \Rightarrow h &= (y_a - y_b) / [f_y (1/Z_1 - 1/(Z_1 + d))] \end{aligned} \quad (3.22)$$

By substituting Equation (3.20) and (3.21) into (3.22), Equation (3.23) is obtained.  $h$  can be evaluated by this equation.

$$\begin{aligned} h &= \frac{(y_a - y_b)}{f_y} / \left( \frac{(x_1 - x_2) - (x_3 - x_4)}{2f_x d} \right) \\ &= \frac{2f_x d}{f_y} \frac{(y_a - y_b)}{(x_1 - x_2) - (x_3 - x_4)} \end{aligned} \quad (3.23)$$

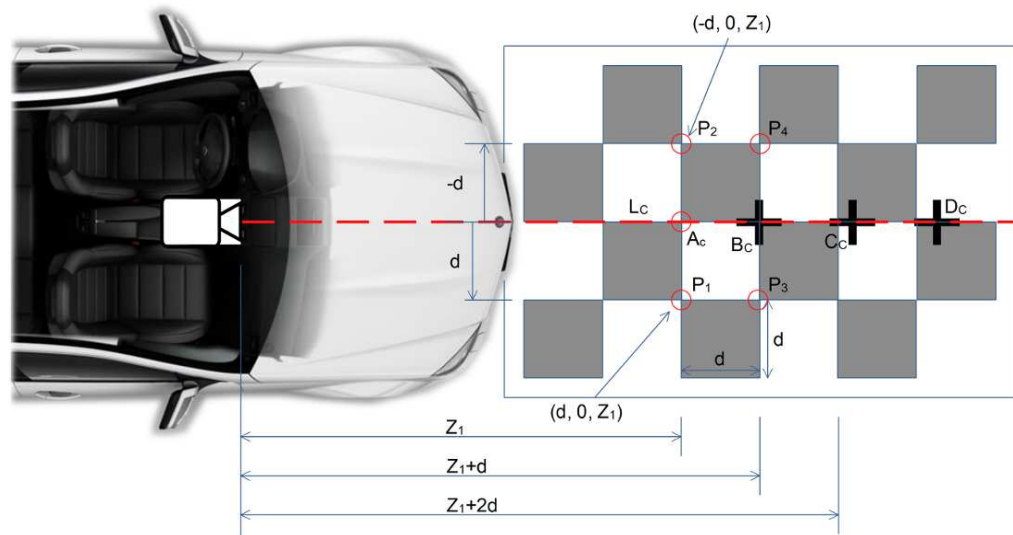


Figure 3-12: Camera height calibration using a banner with checker pattern placing on the level ground.

### 3.2 Ego Motion Estimation

The movement of the observer, also known as ego motion, has to be compensated in the captured image sequence so that the actual motion of objects relative to the ground can be obtained. This process is known as ego motion compensation. It is an important step to identify moving objects in the scene.

Although there has been much research on ego motion estimation based on successive images captured from a moving camera, there are still many exceptional cases that the estimated ego motion is not reliable. For instance, the weak texture of the road region can give ambiguous feature points for optical flow field estimation, and there can be insufficient number of feature points available for ego motion estimation. Also, there can be rapid intensity change in successive images leading to unreliable optical flow estimation.

In addition, ego motion estimation methods mentioned in Chapter 2.3.2 require the extraction of feature points, followed by estimating the optical flow fields of these points. Because of the block based nature of MVs from the H.264/AVC encoder, the MVs cannot be used directly for ego motion compensation. Additional processes such as

outlier removal and feature point detection in each block are required before the MVs can be used for more reliable ego motion estimation. These processes add computation overhead to the system and will affect the real-time performance of the system.

In order to achieve reliable ego motion estimation and the real-time performance requirement, the information provided by the six-degree-of-freedom sensor and the speed sensor were utilised.

The MVs generated from the H.264/AVC video encoder were compensated by the MVs due to the ego motion of the camera. The dynamic parameters of the moving camera, such as translational and rotational speeds, pitch, roll and yaw angles, are available from the six-degree-of-freedom inertial sensor mounting directly to the camera board. The vehicle speed of the ego vehicle is also available from the speed sensor readily available in the vehicle.

With the information from the inertial and speed sensors, the orientation and ego motion of the camera can be estimated more accurately.

### 3.2.1 Planar Homography Estimation

Given the camera intrinsic and extrinsic parameters are obtained from the initial calibration method, the dynamic yaw and pitch angles can be measured by inertial sensors with negligible roll angle, and the vehicle speed can be measured by the vehicle speed sensor. When the camera operates at a known frame rate, the distance travelled by the vehicle on a ground plane can also be calculated.

Referring to the definition of coordinate systems mentioned in Chapter 3.1.2, a point  $P_w$  at  $(X_w, Y_w, Z_w)^T$  on the World coordinates has its corresponding point  $p_{t-1}^c$  at  $(x_{t-1}, y_{t-1}, z_{t-1})^T$  on the camera coordinates at time  $T_{t-1}$ , and  $p_t^c$  at  $(x_t, y_t, z_t)^T$  on the

camera coordinates at time  $T_t$ . The time difference between  $T_{t-1}$  and  $T_t$  is the time duration between successive frames. The orientation  $\theta_x$  represents the pitch angle about the  $X$ -axis,  $\theta_y$  represents the yaw angle about the  $Y$ -axis, and  $\theta_z$  represents the roll angle about the  $Z$ -axis. The transformation from the point  $p_{t-1}^c$  to the point  $p_t^c$  is expressed in Equation (3.24), where  $A$  is the ground plane homography matrix shown in Equation (3.25) (Longuet-Higgins, 1986),  $R_w$  is the camera rotation matrix at the previous frame which is expressed in Equation (3.26),  $R_c$  is the camera rotation matrix between the successive frames which is expressed in Equation (3.27),  $T_c$  is the translation of the camera between successive frames (3.28),  $n^T$  is the unit normal vector to the ground plane in the camera coordinates at time  $T_{t-1}$  which is expressed in Equation (3.29),  $h$  is the height of the camera from the ground plane. The rotational matrix  $R_c$  is actually an approximated matrix where the rotational angles are assumed to be small between successive frames.

$$\begin{aligned}
p_t^c &= R_c \left( p_{t-1}^c - \frac{R_w T_c}{h} n^T p_{t-1}^c \right) \\
&= R_c \left( I - \frac{R_w T_c}{h} n^T \right) p_{t-1}^c \\
&= A p_{t-1}^c
\end{aligned} \tag{3.24}$$

$$A = R_c \left( I - \frac{R_w T_c}{h} n^T \right) \tag{3.25}$$

$$R_w = \begin{bmatrix} c\theta_y c\theta_z & -c\theta_y s\theta_z & s\theta_y \\ c\theta_x s\theta_z + s\theta_x s\theta_y c\theta_z & c\theta_x c\theta_z - s\theta_x s\theta_y s\theta_z & -s\theta_x c\theta_y \\ s\theta_x s\theta_z - c\theta_x s\theta_y c\theta_z & s\theta_x c\theta_z + c\theta_x s\theta_y s\theta_z & c\theta_x c\theta_y \end{bmatrix} \tag{3.26}$$

, where  $c$  and  $s$  denote cosine and sine functions respectively



$$R_c \approx \begin{bmatrix} 1 & -\delta\theta_z & \delta\theta_y \\ \delta\theta_z + \delta\theta_x \delta\theta_y & 1 - \delta\theta_x \delta\theta_y \delta\theta_z & -\delta\theta_x \\ \delta\theta_x \delta\theta_z - \delta\theta_x & \delta\theta_x + \delta\theta_y \delta\theta_z & 1 \end{bmatrix} \quad (3.27)$$

$$\approx \begin{bmatrix} 1 & -\delta\theta_z & \delta\theta_y \\ \delta\theta_z & 1 & -\delta\theta_x \\ -\delta\theta_x & \delta\theta_x & 1 \end{bmatrix}$$

$$T_c = \begin{bmatrix} v_x t_d & v_y t_d & v_z t_d \end{bmatrix}^T \quad (3.28)$$

$$n^T = \begin{pmatrix} -\cos\theta_y \sin\theta_z \\ \cos\theta_x \cos\theta_z - \sin\theta_x \sin\theta_y \sin\theta_z \\ \sin\theta_x \cos\theta_z + \cos\theta_x \sin\theta_y \sin\theta_z \end{pmatrix} \quad (3.29)$$

The normalised image coordinates of  $p_t^c$  at  $(x_t, y_t, z_t)^T$  is  $\hat{p}_t^c = z_t(x'_t, y'_t, 1)^T$  where  $x'_t = x_t/z_t$ ,  $y'_t = y_t/z_t$ . Let  $p_t^s = (u_t \ v_t \ 1)^T$  and  $p_{t-1}^s = (u_{t-1} \ v_{t-1} \ 1)^T$  be the coordinates on the screen at time  $T_t$  and  $T_{t-1}$  respectively, Equation (3.24) can be rewritten as Equation (3.30), where  $K$  is the camera intrinsic matrix expressed in Equation (3.31).  $M = KAK^{-1}$  is the homography matrix relating the point on the screen in the previous frame and the current frame.

$$p_t^s = KA\hat{p}_{t-1}^c = KAK^{-1}\hat{p}_{t-1}^s = Mp_{t-1}^s \quad (3.30)$$

$$p_{t-1}^s = M^{-1}p_t^s$$

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \quad (3.31)$$

### 3.2.2 Ego Motion Compensation

The matrix  $M$  in Equation (3.30) can be used for ego motion compensation. Figure 3-13 and Figure 3-14 show the MVs due to ego motion of the camera only. Both figures display the MVs at 16x16 pixel interval for higher clarity. The motion of the camera in the frame shown in Figure 3-13 consists of straight line motion only, there is

a clear FOE from the centre location of the screen. In contrast, the motion of the camera in the frame shown in Figure 3-14 consists of forward and angular movement. The resulting MV directions are the combined motions due to the forward motion and the angular motion.

One point to note is that the matrix  $M$  in Equation (3.30) represents the planar transformation between the point in the previous and current frame. All points are assumed to be lying on the ground plane only.

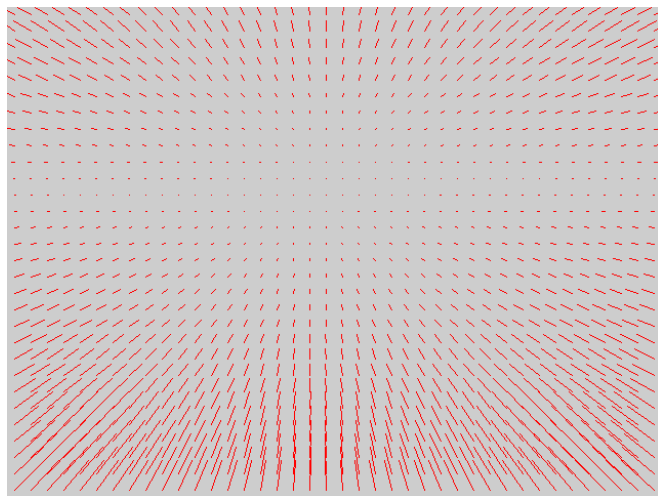


Figure 3-13: Display of MVs due to ego motion only. The MVs are displayed at 16x16 pixel interval with vehicle speed and camera parameters from frame number 10 of the “Intern on bike” sequence of the Daimler sequence.

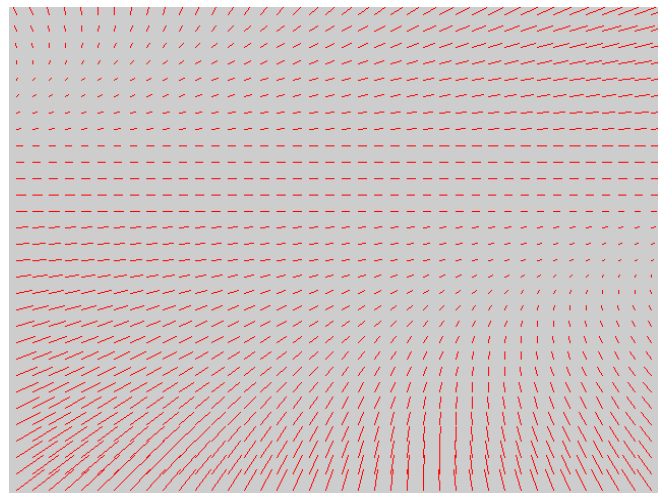


Figure 3-14: Display of MVs due to ego motion only. The MVs are displayed at 16x16 pixel interval with vehicle speed and camera parameters from frame number 202 of the “Crazy Turn” sequence of the Daimler sequence.

With the presence of the H.264/AVC encoder, an object at point  $p_2^s = (u_2 \ v_2 \ 1)^T$  on the screen at the current frame is related to a point  $p_1^s = (u_1 \ v_1 \ 1)^T$  in the previous frame by Equation (3.32), where  $\delta_V$  is the MV found by the encoder.

$$p_1^s = p_2^s + \delta_V \quad (3.32)$$

The resultant MV  $\delta_V$  evaluated by the encoder is the combined result of the MV  $\delta_G$  due to the ground truth motion of the independently moving object and the MV  $\delta_E$  due to the ego motion of the observer, as expressed in Equation (3.33). Since  $\delta_V$  and  $\delta_E$  are known from the video encoder and the ego motion of the observer respectively, the ground truth MV of the object  $\delta_G$ , also known as ego-compensated MV, can be evaluated using Equation (3.33).

$$\delta_V = \delta_E + \delta_G \quad (3.33)$$

### 3.2.3 Focus of Expansion Estimation

The Focus of Expansion (FOE) is the point in the screen where static objects are virtually emerging from. In the proposed algorithm for MV based moving object detection, the FOE is used as the reference point for finding the direction of MVs. Static objects have MVs with directions pointing to the FOE. Therefore, MVs with direction not pointing to the FOE is an indication of existence of independently moving objects. The FOE is evaluated by the on-board inertial sensor rather than the estimation methods making use of features points in the captured images.

When the ego vehicle moves on the road, the camera will experience 3-dimensional dynamic motions. By making use of the camera calibration method mentioned in Chapter 3.1.2, the camera has been installed with zero pitch, yaw and roll angles when the vehicle was stationary on a level road.

When the vehicle is moving in a straight line on a level road, the camera's pitch angle relative to the road will vary due to pot holes and un-evenness of the road surface. The camera's roll angle also varies due to the same reason. Similarly, when the vehicle is moving around a bend on the road, its roll and yaw angles vary according to the angular speed of the vehicle. The roll angle is due to the lateral acceleration during cornering leading to change of height of suspensions and tire deformation. The yaw angle is due to the angular translation of the vehicle along the bend of the road.

Since the yaw angle between the camera and the vehicle is fixed by the rigid installation, the measured yaw angle between successive frames is solely due to the angular translation on the road. Similarly, the roll angle is due to the angular speed induced lateral acceleration and the un-evenness of the road.

The pitch angle is more complicated. It is due to the un-evenness of the road, the ego vehicle's acceleration in the Z-direction, and the inclination of the road relative to the earth plane. As illustrated in Figure 3-15 with a vehicle travelling on an inclined road. If the camera is installed with zero pitch angle, the angle between the camera optical axis and the road plane is zero. However, the pitch angle measured by the inertial sensor is actually the angle between the earth plane and the road plane.

Therefore, the gradient of the road will offset the pitch angle measurement. It has to be compensated to reflect the true pitch angle between the camera optical axis and the road plane.

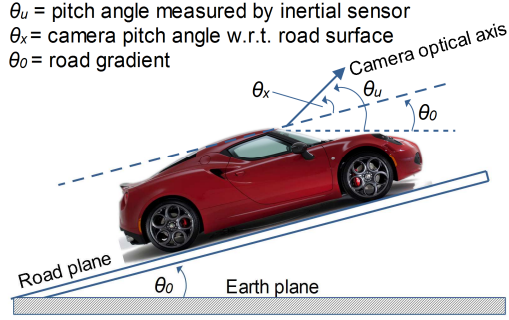


Figure 3-15: Illustration of a vehicle on an inclined road, the pitch angle measured by the inertial sensor is the angle between the road plane and the earth plane rather than the angle between the camera optical axis and the road plane.

The camera pitch angle to the road plane  $\theta_x$  can be interpreted as the summation of instantaneous change and long-term change of pitch angles represented by Equation (3.34), where  $\delta\theta_x$  is the instantaneous pitch angle measured between successive frames,  $\theta_x(t)$  is the pitch angle reported by the inertial sensor at the current frame,  $\Theta$  is the long-term accumulation of the pitch angle that can be calculated by the moving average of the reported pitch angle over the past few frames as formulated in Equation (3.35), where  $n$  is the number frame for calculating the moving average. The instantaneous change is contributed by the un-evenness of the road surface and the vehicle's acceleration. The long-term change is contributed by the gradient of the road relative to the earth plane.

$$\theta_x = \delta\theta_x + \theta_x(t) - \Theta \quad (3.34)$$

$$\Theta = \frac{1}{n} \sum_{i=0}^{n-1} \theta_x(t-i) \quad (3.35)$$

By making use of the built-in inertial sensor of the camera unit, the instantaneous change of pitch angle can be obtained by the angular speed reading  $\omega_x$  in the  $x$ -axis from the sensor, and the time interval  $\delta t$  between successive frames. The instantaneous pitch angle  $\delta\theta_x$  can be expressed as Equation (3.36).

$$\delta\theta_x = \omega_x \delta t \quad (3.36)$$

Assuming zero translational change in all axes, and knowing that the yaw angle between the camera and the vehicle body is zero due to the camera calibration,

Equation (3.4) can be simplified to Equation (3.37) and (3.38). The vanishing point  $[x_0 \ y_0 \ 1]^T$  can be evaluated by substituting  $[X_w \ Y_w \ Z_w]^T$  in Equation (3.37) by  $[0 \ 0 \ 1]^T$  (Hartley and Zisserman, 2003) and expressed in Equation (3.39)

$$p^s = [x_s \ y_s \ 1]^T = KR_w [X_w \ Y_w \ Z_w]^T \quad (3.37)$$

$$\begin{aligned} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\cos \theta_x \sin \theta_z & \cos \theta_x \cos \theta_z & \sin \theta_x \\ \sin \theta_x \sin \theta_z & -\sin \theta_x \cos \theta_z & \cos \theta_x \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \\ &= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \end{aligned}$$

, where

$$\begin{aligned} \begin{bmatrix} r_{11} \\ r_{21} \\ r_{31} \end{bmatrix} &= \begin{bmatrix} f_x \cos \theta_z + c_x \sin \theta_x \sin \theta_z \\ -f_y \cos \theta_x \sin \theta_z + c_y \sin \theta_x \sin \theta_z \\ \sin \theta_x \sin \theta_z \end{bmatrix} \\ \begin{bmatrix} r_{12} \\ r_{22} \\ r_{32} \end{bmatrix} &= \begin{bmatrix} f_x \sin \theta_z - c_x \sin \theta_x \cos \theta_z \\ f_y \cos \theta_x \cos \theta_z - c_y \sin \theta_x \cos \theta_z \\ -\sin \theta_x \cos \theta_z \end{bmatrix} \\ \begin{bmatrix} r_{13} \\ r_{23} \\ r_{33} \end{bmatrix} &= \begin{bmatrix} c_x \cos \theta_x \\ f_y \sin \theta_x + c_y \cos \theta_x \\ \cos \theta_x \end{bmatrix} \end{aligned} \quad (3.38)$$

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_x \\ f_y \tan \theta_x + c_y \\ 1 \end{bmatrix} \quad (3.39)$$

Recalling that Equation (2.6) indicated that when the camera has zero rotational angles, the FOE is  $[(c_x + V_x f / V_z) \ (c_y + V_y f / V_z) \ 1]^T$ . So, the FOE is at  $[c_x \ c_y \ 1]^T$  when both  $V_x$  and  $V_y$  are zero. It differs from Equation (3.39) by  $f_y \tan \theta_x$  in the  $y$ -axis. Therefore, combining the FOE due to camera rotation with the FOE due to ego motion, the FOE can be expressed as Equation (3.40).

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_x + V_x f / V_z \\ f_y \tan \theta_x + c_y + V_y f / V_z \\ 1 \end{bmatrix} \quad (3.40)$$

Since the vehicle is moving on the road, its vertical speed along the  $Y$ -axis can be approximated as zero. The FOE expressed in Equation (3.40) can be modified to Equation (3.41).

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_x + V_x f_x / V_z \\ f_y \tan \theta_x + c_y \\ 1 \end{bmatrix} \quad (3.41)$$

The overall pitch angle  $\theta_x$  of the camera relative to the road plane is calculated by Equation (3.34). Assuming the vehicle is moving at constant speed  $v$  with constant angular yaw rate  $\omega_y$ , the time difference between successive frames is  $\delta t$ .  $v$  can be obtained from the vehicle speed sensor, and  $\omega_y$  can be obtained from the angular speed reading of the  $y$ -axis of the inertial sensor. A simple vehicle model shown in Figure 3-16 can be used to estimate the linear motion of the ego vehicle along the  $X$ - and  $Z$ -axis. Assuming that the vehicle speed  $v$  is constant, the distance travelled by the vehicle along the  $X$ - and  $Z$ -axis is  $\delta X$  and  $\delta Z$  respectively as shown in Equation (3.42) and (3.43).

$$\delta X = v / \omega_y (1 - \cos \omega_y \delta t) \quad (3.42)$$

$$\delta Z = v / \omega_y \sin \omega_y \delta t \quad (3.43)$$

For small time difference  $\delta t$  between successive frames, the speed  $V_x$  and  $V_z$  in Equation (3.41) can be approximated by  $\delta X / \delta t$  and  $\delta Z / \delta t$  respectively. By substituting Equation (3.34), (3.35), (3.36) and (3.42) to (3.41), Equation (3.44) is obtained for evaluating the FOE at  $(x_0, y_0)$ .

$$\begin{bmatrix} x_0 \\ y_0 \\ 1 \end{bmatrix} = \begin{bmatrix} c_x + f_x (1 - \cos(\omega_y \delta t)) \sin(\omega_y \delta t) \\ c_y + f_y \tan\left(\omega_x \delta t + \theta_x(t) - \frac{1}{n} \sum_{i=0}^{n-1} \theta_x(t-i)\right) \\ 1 \end{bmatrix} \quad (3.44)$$

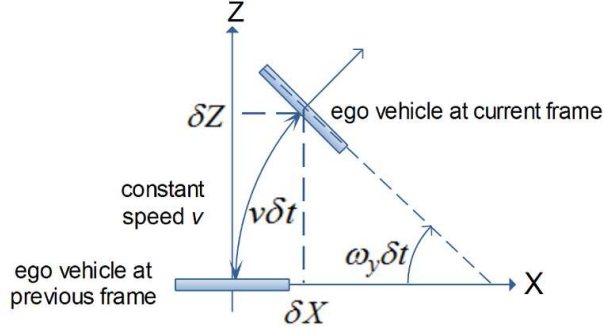


Figure 3-16: Simple vehicle motion model in bird's-eye-view. The vehicle travels at speed  $v$ , and turning at angular rate  $\omega$ . The time interval between successive frames is  $\delta t$ .

### 3.3 Road Region Detection

There are several observations from the road region detection methods reviewed in Chapter 2.5.

Firstly, they require considerable computation time. A method that can achieve real-time performance is required for the proposed ADAS.

Secondly, those road detection methods were trying to classify each pixel into a road or non-road pixel. This process can be very time consuming. For the application in the proposed system, identified road region is an indication to reject falsely detected moving objects. Since the detection of moving objects are mostly block based due to the block-based nature of MVs from H.264/AVC encoders, it is possible that the road detection boundaries can be aligned to the block boundaries of MVs. By using block-based instead of pixel-based approach, a lower computation cost is expected.

Thirdly, the bottom part of the captured image immediately in front of the ego vehicle is highly probably a road region, sampling of the characteristics of the road region in this



area is a fast and reliable method to determine the road colour model (Lee and Crane, 2006, Tan and Hong et al., 2006).

Fourthly, it is more important to detect the road region near the ego vehicle than regions that are further away. This is because the MVs from static objects near the ego vehicle are having larger distance to the FOE, meaning that the MV amplitudes near the ego vehicle are expected to be larger. Figure 3-17 shows a typical captured image with the direction and amplitude of MVs shown. The areas highlighted with red circles are closer to the ego vehicle. Motion estimation error in these regions will result in erroneous MVs of large amplitudes. There can be falsely detected moving object in these areas. Such false detection can be eliminated if the road region in this area is identified.

Finally, the target application of the proposed ADAS is on structured roads. The texture of structured roads is usually weak with uniform colour. Therefore, except those areas with markings, the variation of road colour should be small, and the colour of adjacent area of a particular road region should not deviate by a large amount.

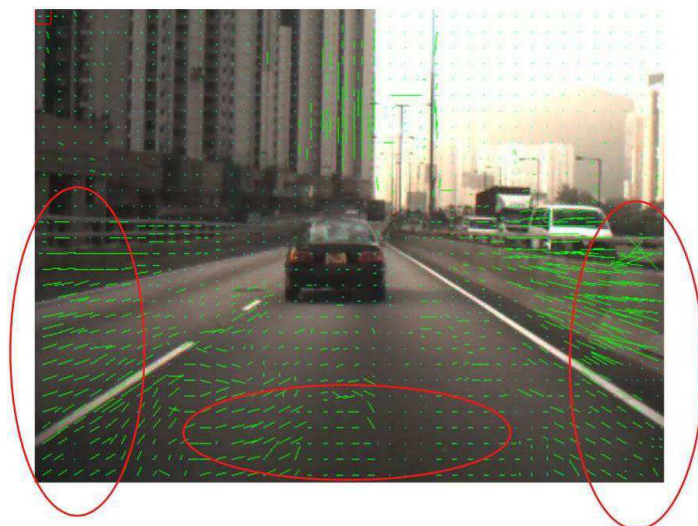


Figure 3-17: Relatively large MVs highlighted in red circles are from static objects near the ego vehicle.

Based on these observations, a new block based road region detection algorithm is proposed with steps as follows.

### 3.3.1 Building Road Colour Model

A road colour model is prepared by sampling blocks on the road from multiple images. The mean ( $\bar{x}$ ) and standard deviation ( $\sigma$ ) of each block are evaluated using Equation (3.45) and (3.46), where  $(x,y)$  are the starting coordinates of the block,  $R(u,v)$ ,  $G(u,v)$  and  $B(u,v)$  are the intensity of the red, green and blue channel of the pixel at  $(u,v)$ , and  $Hist(i)$  is the number of pixels with the average sum of the three colour channels equals to  $i$ . The block size was set to 16x16 pixels in this case. A set of images with different light intensities taken on different roads were selected manually, and blocks on the road in each image were also selected manually from the area at the bottom of the image immediately in front of the ego vehicle. No block with road marking was selected.

$$\bar{x} = \frac{1}{3} \frac{\sum_{v=y}^{y+15} \sum_{u=x}^{x+15} (R(u,v) + G(u,v) + B(u,v))}{16^2} \quad (3.45)$$

$$\sigma = \sqrt{\sum_{i=0}^{255} \frac{(\bar{x} - i)^2 Hist(i)}{16^2}} \quad (3.46)$$

A total of 2,500 samples were taken from the set of image to build a table that relates  $\bar{x}$  to a range of  $\sigma$ . This table was used as the colour models for initial selection of a road patch for region grow.

### 3.3.2 Seed Block for Road Region Grow

When an image is captured, a block near the bottom of the image is evaluated for its mean ( $\bar{x}$ ) and standard deviation ( $\sigma$ ). 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 markings on the road. If either  $\bar{x}$  or  $\sigma$  exceeds the values defined in the colour model, another block along the row will be chosen for evaluation of  $\bar{x}$  and  $\sigma$

again until the new  $\bar{x}$  and  $\sigma$  of the block can satisfy the colour model. So,  $\bar{x}$  and  $\sigma$  are compared with the range of allowable value stored in the table described in Section 3.3.1, which is the colour model for the road. Figure 3-18 shows a typical captured image from the camera mounted on the ego vehicle overlaid with lines of grid size 16x16. A road patch of size 16x16 pixels is selected from the bottom left of the image that is shown in a blue square. Figure 3-18 also shows a block highlighted in red colour near the bottom right of the image. Since this block has road marking on it, the value of  $\bar{x}$  and  $\sigma$  of this block will exceed the values specified in the road colour model and hence it will be rejected as being a seed block.



Figure 3-18: Captured image showing 16x16 grid lines in green colour. The seed block is searched from the bottom left to the bottom right of the image until a block is found with satisfactory mean and standard deviation. An example block highlighted in orange colour is compared to its neighbour blocks marked with number 1 to 8 with purple colour.

### 3.3.3 Road Region Grow

After selecting a block of size 16x16 along the row at the bottom of the captured image, the values of mean ( $\bar{x}$ ) and standard deviation ( $\sigma$ ) of this block is stored. Since the road surface usually has uniform colour and weak texture,  $\bar{x}$  and  $\sigma$  of one block of a road region should be close to those of its neighbouring blocks.

When a block  $B_i$  with mean gray level  $\bar{x}_i$  and standard deviation  $\sigma_i$  along the row of block near the bottom of the image is identified as a road region block, the eight neighbouring blocks surrounding the block  $B_i$  are also evaluated for their respective mean  $\bar{x}_k$  and standard deviation  $\sigma_k$  where  $k$  denotes one of the eight blocks. A

neighbouring block  $B_k$  is identified as a road region block if it satisfies (3.47) and (3.48), where  $b$  and  $c$  are predefined thresholds.

$$\|\bar{x}_k - \bar{x}_i\| < b \quad (3.47)$$

$$\|\sigma_k - \sigma_i\| < c \quad (3.48)$$

The region grow method is employed so that the road region can “grow” further by comparing the identified road region blocks with their neighbouring blocks. Each comparison employs the  $\bar{x}$  and  $\sigma$  of the centre block, such as the one highlighted orange in Figure 3-18, and one of the eight neighbouring blocks, such as the 8 blocks numbered from 1 to 8 in Figure 3-18, until no new road region block can be identified.

### 3.3.4 Post Road Region Grow Refinement

Since there are some blocks on the road that are actually road region but are excluded by the road region grow algorithm due to the value of  $\bar{x}$  or  $\sigma$  in comparison cannot satisfy (3.47) and (3.48), a post processing step that uses a hole filling algorithm is proposed to refine the detected road region.

Figure 3-19 shows a typical detected road region of a captured frame. The detected road region is highlighted in white colour. There are some “holes” inside the detected road region. The hole-filling algorithm scans each row of the road region detection result from the bottom of the image. The minimum and maximum road region block column number at each row  $y_b$  are represented by  $X_{Min}(y_b)$  and  $X_{Max}(y_b)$  respectively. Similarly, the minimum road region block row number at each column  $x_b$  is represented by  $Y_{Min}(x_b)$ .

The scanning of road region blocks starts from the row below the FOE, with  $x_b$  scanning from left to right, and  $y_b$  from top to bottom. When a non-road region block at

column  $x_b$  is detected and it is located below  $Y_{Min}(x_b)$ , that is, its location number is larger than  $Y_{Min}(x_b)$ , the block is re-labelled as a road region block.

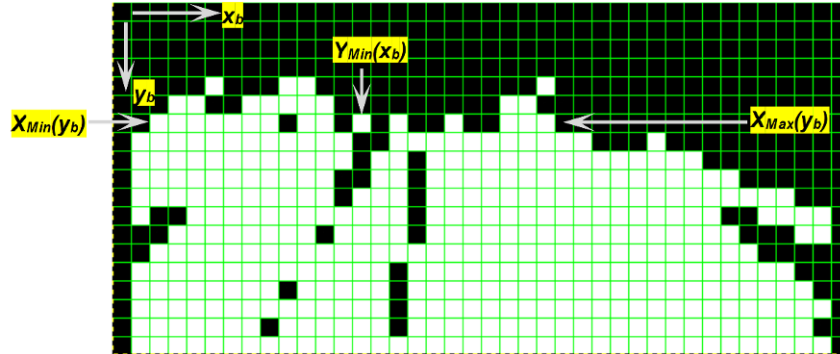


Figure 3-19: Road region blocks are highlighted in white. The minimum and maximum road region block at each row  $y_b$  are  $X_{min}(y_b)$  and  $X_{max}(y_b)$  respectively. The minimum road region block at each column  $x_b$  is  $Y_{min}(x_b)$ .

After the refinement process, the holes inside the road region are filled and re-labelled as road region. The result after the refinement process is shown in Figure 3-20.

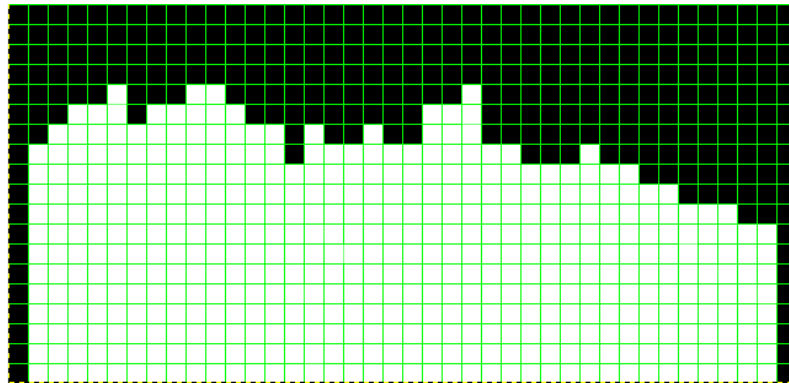


Figure 3-20: The road region detection result after the hole-filling refinement process.

### 3.4 Segmentation of Regions of Interest

The road detection result using the method mentioned in Chapter 3.3 is combined with the amplitude MVs from the H.264/AVC encoder to form the region of interest (ROI) for moving object detection.

As reported in Chapter 2.4.3, MVs near the FOE and moving objects with slow relative speed to the ego vehicle can be very small. The small MVs are inaccurate due to the limited precision of H.264/AVC encoder. Since performing ego motion compensation on

these inaccurate MVs will only result in erroneous representation of moving objects, other methods on the detection of relatively slow moving objects are required.

By making use of the amplitudes of MVs, the ROIs can be segmented into regions with relatively slow moving objects and relatively fast moving objects.

### 3.4.1 ROI for Slow Relative Speed Objects

Regions that potentially have slow relative speed moving objects exhibit MVs with small amplitudes. Therefore, the ROI for slow relative speed moving object is chosen as the regions with small MV amplitudes. In addition, the ROI can be further reduced by the detected road region and limiting the ROI to areas below the FOE.

Figure 3-21(a) shows a typical captured image from the camera. Since the area above the FOE is mostly the sky or upper parts of moving vehicles, there is no useful information for the detection of moving objects. Therefore, the upper part of the capture image is ignored. More precisely, given the  $y$ -coordinate of the FOE is  $y_0$ , the area below  $y_0+16$  is retained for moving object detection.



Figure 3-21: (a) Typical captured image that has been converted to grayscale image. (b) The FOE and the primary ROI is selected as the area below the FOE.

In addition to the primary ROI below the FOE, the construction of the ROI for slow relative speed object detection is further illustrated in Figure 3-22. Figure 3-22(a)

shows the road region detected for the captured image shown in Figure 3-21(a) by using the method described in Chapter 3.3.1 to 3.3.4. Figure 3-22(b) shows the image mask with amplitude of MVs larger than a threshold  $q_m$ . Since each MV represents a block of size 8x8, the mask has regions highlighted in block-by-block basis. The white areas in the image mask represent areas with MV amplitudes larger than  $q_m$ . Figure 3-22(c) shows the resultant ROI with only the area below the FOE shown. The white blocks are areas to be ignored for slow relative speed moving object detection.

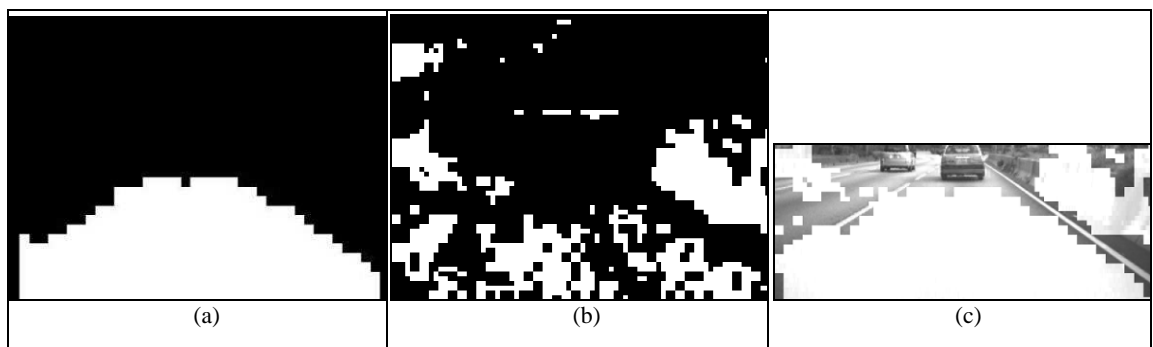


Figure 3-22: Illustration of ROI construction. (a) Image mask by road region identification. (b) Image mask by filtering MVs with amplitude larger than a threshold. (c) Cropped image that combines the image mask (a) and (b).

The result of ROI construction shown in Figure 3-22(c) still has some areas that are not eliminated for relatively slow moving object detection. These areas include those highlighted in red circles in Figure 3-23(a). These areas are removed from the ROI by noticing the maximum and minimum  $y$ -coordinates of highlighted (white) blocks at each column of the ROI image similar to the hole-filling method mentioned in Chapter 3.3.4. If a particular block in a column is inside the maximum and minimum bounds of the highlighted blocks, the block is also highlighted to white to indicate it is the block to be ignored. After running this refinement process, the modified ROI is shown in Figure 3-23(b).

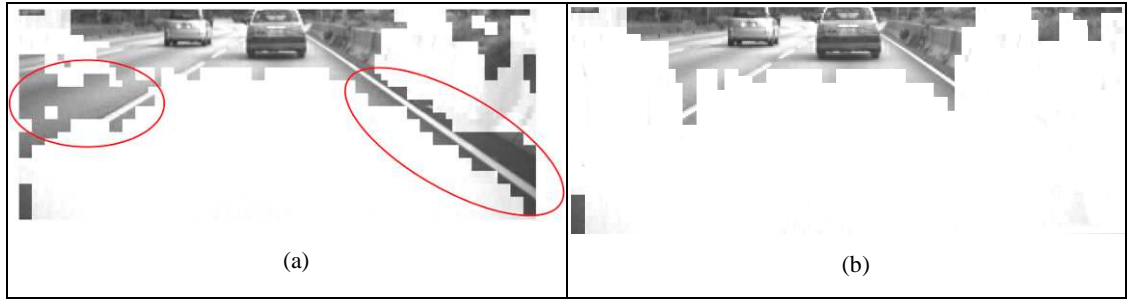


Figure 3-23: (a) Regions highlighted in red circles are areas that should be ignored. (b) ROI after refinement.

It is observed that the area that remains for detection of slow relative speed vehicles is significantly smaller than the area of the original image. For instance, the ROI is only 122 blocks for the image shown in Figure 3-23. Comparing to the 1,200 blocks for the corresponding full-size captured image, the ROI is only 10.2% of the original image. This ensures the detection algorithm can be completed in much shorter time by examining less area of interest.

### 3.4.2 ROI for Fast Relative Speed Objects

Similar to the ROI construction procedures for slow relative speed moving object detection, the ROI for fast relative speed moving object detection is illustrated in Figure 3-24. It is composed of the result from road detection shown in Figure 3-24(a), and the image mask with regions that the amplitudes of MVs are larger than or equal to a threshold  $q_m$  shown in Figure 3-24(b). Figure 3-24(b) is the inverse binary image of Figure 3-22(b). The combined result of Figure 3-24(a) and Figure 3-24(b) is shown in Figure 3-24(c). Although Figure 3-24(c) shows only a small portion of the image is remained for fast relative speed moving object detection, the area for fast relative speed moving object detection depends on whether there are relatively fast speed moving objects.



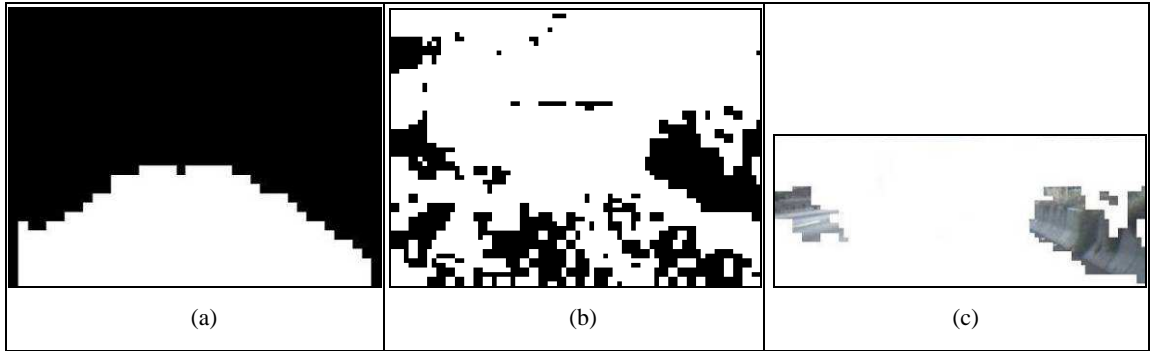


Figure 3-24: Illustration of ROI construction for fast relative speed vehicle detection. (a) Image mask by road region identification. (b) Image mask by filtering MVs with amplitude larger than a threshold. (c) Cropped image combining image mask (a) and (b).

### ***3.5 Slow Relative Speed Moving Object Detection***

MVs obtained from the video encoder are the result of both global motion (also known as ego-motion) due to the moving camera and the local motion due to moving objects on the road. As mentioned in Chapter 2.4.3, the precision of MVs of an H.264/AVC based encoder is only up to a quarter pixel, the MVs obtained for moving vehicles with slow relative speed to the observing camera will be similar to regions of far-away background and weak-texture road regions. This observation has been reported in Chapter 2.4.1. To overcome the difficulty of detecting slow relative speed moving objects, the method proposed in this research is to split the detection task to relatively slow speed and relatively fast speed moving object detection.

This Chapter proposes the method to detect rear-view vehicles with slow relative speed to the ego vehicle. 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 the H.264/AVC encoder, dividing the region of interest for slow relative speed vehicle detection.

### 3.5.1 Slow Relative Speed Vehicle Detection Method

The functional flow chart of the proposed slow relative speed vehicle detection method is shown in Figure 3-25. It consists of steps to find the true horizontal gradient, detect horizontal and U-shape contours. A tracking algorithm is also proposed, making use of an expanded detection window based on the detection window in the previous frame and the evaluation of vertical and horizontal gradients.

The region of interested is constructed according to the method mentioned in Chapter 3.4.1. Only the area outside the detected road region with amplitudes of MVs smaller than a threshold  $q_m$  is retained.

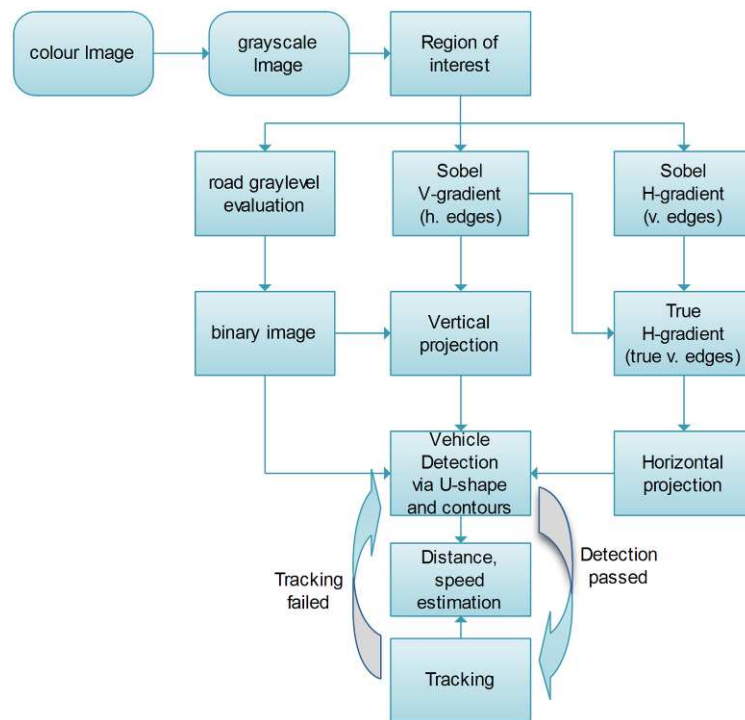


Figure 3-25: Functional block diagram of the slow relative speed vehicle detection algorithm

### 3.5.2 Binary Image Creation

During the road region identification stage, the maximum  $\bar{x}_{\max}$  and minimum  $\bar{x}_{\min}$  grey-scale levels of the road region have been evaluated. Each pixel inside the ROI of the cropped input grey-scale image is compared with  $\bar{x}_{\min}$  to create a binary image. If

a pixel inside the ROI of the grey-scale 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 Figure 3-26(a) where the white zone represents the area that is darker than  $\bar{x}_{\min}$ . Figure 3-26(b) shows the binary image overlaid on the ROI image. The horizontal contours of the rear-view of vehicles can be seen clearly.



Figure 3-26: (a) Binary image with those white areas representing regions that are darker than the minimum graylevel of the road region. (b) Binary image in (a) overlaid to the ROI of the captured image. Those horizontal contours along the rear part of vehicles at the front are identified.

### 3.5.3 Vehicle Detection

Besides the identified darkest area of the image as shown in Figure 3-26, a Sobel filter is also applied to the cropped grey-scale input image to find the horizontal gradient and vertical gradient inside the ROI. The Sobel kernels for finding horizontal and vertical gradients are shown in Figure 3-27(a) and Figure 3-27(b) respectively.

$$\begin{array}{ccc} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} & & \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \\ \text{(a)} & & \text{(b)} \end{array}$$

Figure 3-27: (a) Sobel kernel for finding horizontal gradients. (b) Sobel kernel for finding vertical gradients.

The result of finding horizontal gradient and vertical gradients are shown in Figure 3-28(a) and Figure 3-28(b) respectively. Figure 3-28 (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 found by applying the horizontal gradient kernel (Figure 3-28(a)). Some found "vertical" contours in Figure 3-28(a), such as the lane markings, are not truly vertical contours. They are

further eliminated by comparing the pixel in the vertical gradient image shown in Figure 3-28(b) with the corresponding pixel in the horizontal gradient image shown in Figure 3-28(a) using Equation (3.49), 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 Figure 3-29.

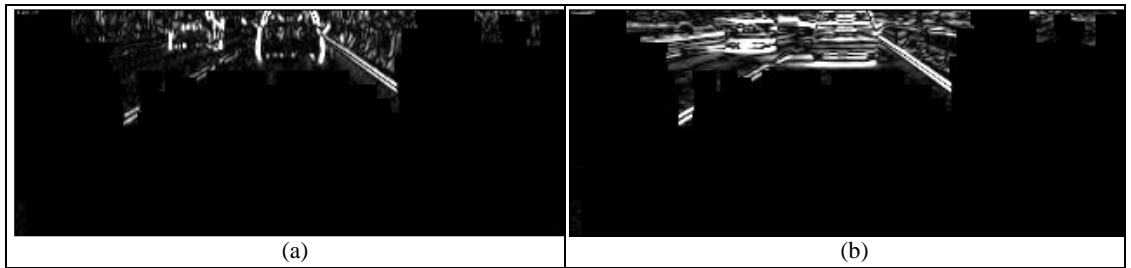


Figure 3-28: Grayscale image with Sobel filtering. (a) Resultant image after applying horizontal gradient Sobel kernel. (b) Resultant image after applying vertical gradient Sobel kernel.

Figure 3-30 is the combined result of the detected vertical contours in Figure 3-29 and the detected darkest region in the image shown in Figure 3-26(a). The white area in Figure 3-26(a) is replaced by red in Figure 3-30(a) for higher clarity. It shows clearly the position of a vehicle at the front with the U-shape highlighted by the green bracket in Figure 3-30(b).

$$P_{vv} = \begin{cases} (P_h - P_v) & \forall P_h \in [P_v, P_v + D_{hv}] \\ 0 & \text{otherwise} \end{cases} \quad (3.49)$$



Figure 3-29: Resultant image of true vertical contour image, after using Equation (3.49).

The proposed algorithm is to detect the U-shape which is the result of horizontal and vertical contours of the vehicle. The algorithm starts by searching for the position of horizontal lines from the bottom of the binary image shown in Figure 3-26(a). If a horizontal line  $L$  with two end points  $(x_1, y_1)$  and  $(x_2, y_2)$  is detected, the width of the horizontal line is evaluated by converting the two endpoints of the line in the screen coordinates to the corresponding World coordinates,  $(X_{w1}, Y_{w1})$  and  $(X_{w2}, Y_{w2})$  respectively, using Equation (3.10). If the resulting width ( $W = X_{w2} - X_{w1}$ ) is longer than  $W_U$  and shorter than  $W_L$ , where  $W_U$  and  $W_L$  are predefined upper and lower limits of the width threshold respectively, the line will be discarded.



Figure 3-30: (a) Combined result of the true detected vertical contours (shown in white colour) and the detected darkest region in the image (shown in red colour). (b) U-shape bracket drawn in green colour, indicating the found U-shape due to the vehicle.

If the line  $L$  falls within  $W_L$  and  $W_U$ , it is more likely that this line is produced by a vehicle than by the environment. 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 rectangular area as shown in Figure 3-31 that is around the line location, will be further examined to confirm if there is a vehicle.

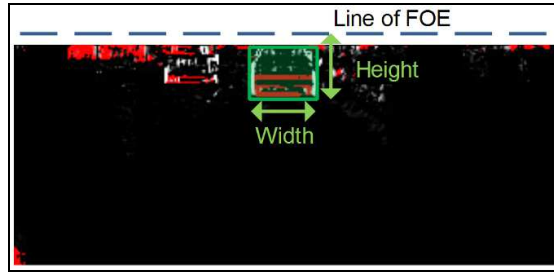


Figure 3-31: Illustration of the area shown in green rectangle identified for examination of whether a vehicle exists.

Further evaluation is performed to examine if there are two distinguished vertical contours near the left and right sides of the identified rectangular area, and if the average vertical gradient exceeds a predefined threshold.

The sum of gray levels  $S(x)$  is calculated by Equation (3.50) 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 rectangle are evaluated.

A plot of  $S(x)$  against the horizontal position is shown in Figure 3-32(b). The maximum  $S(x)$  for the left and right sides is found separately by comparing all  $S(x)$  in their corresponding side. If the maximum  $S(x)$  on both sides of the rectangle exceeds a predefined threshold  $S_h$ , the existence of the vertical contours is confirmed.

The left and right sides of the rectangle will be replaced by the identified positions of the maximum  $S(x)$ . The existence of a vehicle in the red rectangular box indicated in Figure 3-32(a) is further confirmed by evaluating the average vertical gradient  $V(x)$  inside the new rectangular box using Equation (3.51), where  $x_1'$  and  $x_2'$  are the new horizontal position of the rectangle. If the value of  $V(x)$  exceeds a predefined threshold  $V_h$ , a vehicle at the rectangular position is confirmed as detected.

$$S(x) = \sum_{y=y_0}^{y_1} I(x, y) \quad \forall x \in [x_1 - e, x_1 + e] \quad (3.50)$$

$$V(x) = \frac{\sum_{x=x'_1}^{x'_2} \sum_{y=y_0}^{y_1} I(x, y)}{[(x'_2 - x'_1 + 1)(y_1 - y_0 + 1)]} \quad (3.51)$$

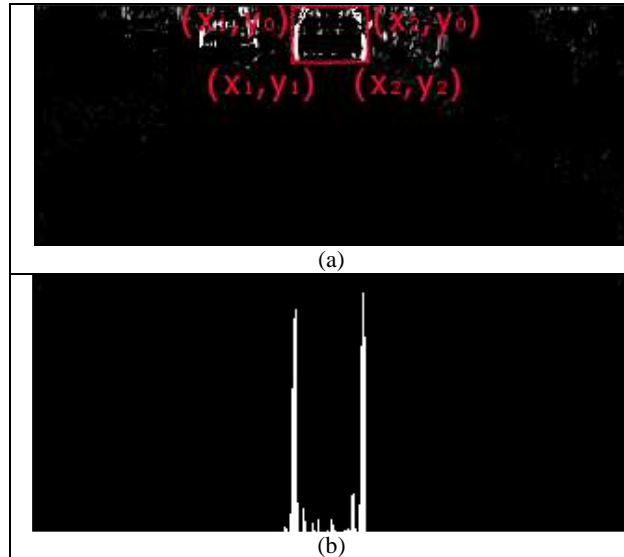


Figure 3-32: Illustration of the vertical projection of the horizontal gradient image around the rectangular position where a vehicle potentially exists. (a) Horizontal gradient image and the rectangular area under evaluation. (b) Corresponding vertical projection near the red rectangular area.

### 3.5.4 Vehicle Tracking

Since the detection of slow relative speed vehicles requires the examination of qualified line features inside the ROI, the processing speed will vary according to the number of potential line features detected. There is the possibility that the detection algorithm cannot be completed within the duration between successive frames. Since the vehicles to be detected are moving relatively slowly, their size and position would not deviate by a large amount across several frames. Therefore, even though there are chances of skipped frames, the accuracy of the detection algorithm will not be affected.

Nevertheless, the computational cost can be reduced by the use of a tracking algorithm. Since the initial position of a detected vehicle is known from the detection algorithm, the tracking algorithm can check for some invariant features inside a search window of reasonable size with reference to the detected vehicle position. The conceptual flow-chart of the tracking algorithm is shown in Figure 3-33.

Given the slow relative speed vehicle detection algorithm has identified the vehicle on the screen bounded by a rectangle at  $(x_{left}, y_{top})$  to  $(x_{right}, y_{bottom})$ , a new image is captured by the system and is converted to a grayscale image. The Sobel kernels for finding horizontal and vertical gradients shown in Figure 3-27(a) and Figure 3-27(b) respectively are used to generate the horizontal and vertical gradient images. Since only the area near the bounded rectangle is used by the tracking algorithm, the Sobel kernels are applied to the area inside  $(x_{left} - 2e_x, y_{top} - 2e_y)$  to  $(x_{right} + 2e_x, y_{bottom} + 2e_y)$  only, where  $e_x$  and  $e_y$  are the number of pixels to expand in the  $x$ - and  $y$ -coordinate of the screen respectively.

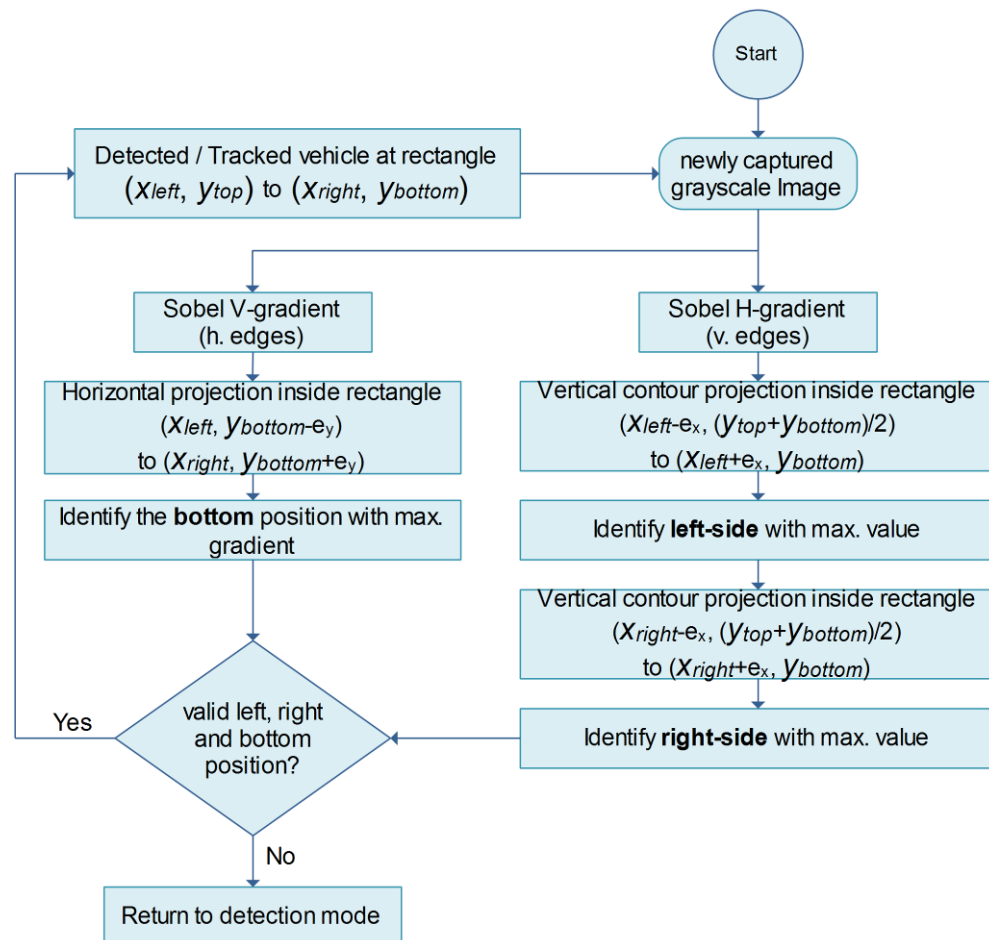


Figure 3-33: Conceptual flow-chart of the tracking algorithm for slow relatively speed moving vehicles

In the next phase, the horizontal projection inside the rectangle  $(x_{left}, y_{bottom} - e_y)$  to  $(x_{right}, y_{bottom} + e_y)$  is evaluated by making use of the vertical gradient image.  $S(y)$  in



Equation (3.52) evaluates the value of horizontal projection at each  $y$ -coordinate inside the selected boundary.

$$S(y) = \sum_{x=x_{left}}^{x_{right}} I(x, y) \quad \forall y \in [y_{bottom} - e_y, y_{bottom} + e_y] \quad (3.52)$$

The gradient at each  $y$ -coordinate inside the boundary is found by Equation (3.53). The maximum gradient  $S_{GMAX}(y)$  is found by comparing  $S_G(y)$  inside the boundary.

$$S_G(y) = S(y+1) + S(y+2) - S(y-1) - S(y-2) \quad (3.53)$$

$$\forall y \in [y_{bottom} - e_y, y_{bottom} + e_y]$$

Similarly, the horizontal contour projection for the left-side is evaluated inside the

rectangle  $\left( x_{left} - e_x, \frac{y_{top} + y_{bottom}}{2} \right)$  to  $\left( x_{left} + e_x, y_{bottom} \right)$  by making use of the

horizontal gradient image.  $S_G(x)$  in Equation (3.54) evaluates the value of the vertical contour projection at each  $x$ -coordinate inside the selected boundary. It essentially accumulates the intensity vertically along the  $y$ -axis with a reduced value according to the difference between two pixels. This can help reduce the sensitivity to the discontinuity in a vertical line. The maximum  $S_{GMAX}(x)$  is found by simply comparing the values of  $S_G(x)$ . The horizontal contour projection for the right-side is similar to that for the left-side. The difference is that the boundary changes to

$\left( x_{right} - e_x, \frac{y_{top} + y_{bottom}}{2} \right)$  to  $\left( x_{right} + e_x, y_{bottom} \right)$ .

$$S_G(x) = \sum_{y=\frac{y_{top} + y_{bottom}}{2}}^{y_{bottom}} \left[ I(x, y) - \|I(x, y) - I(x, y-1)\| \right] \quad \forall x \in [x_{left} - e_x, x_{left} + e_x] \quad (3.54)$$

The values of  $S_{GMAX}(\cdot)$  are compared with predefined thresholds. If they are within the allowable range, the tracking of the vehicle is successful. The bounding rectangle is then updated for being used in the next frame for continued vehicle tracking. Therefore,

assuming the detected vehicle does not deviate from the last position by a large amount, the tracking algorithm only needs to evaluate three small regions to identify the left, right and bottom sides of the vehicle. If the tracking fails, the detection algorithm mentioned in Chapter 3.5.3 will be used.

### 3.5.5 Distance and Speed Estimation

After confirming the position of the vehicle, its true moving speed is estimated by the MVs at the bottom line of the rectangle. In a Driver Assistance System, the potential risks of identified moving objects to the driver are related to the time-to-collision between the ego vehicle and the moving objects. Unlike stereo cameras that can estimate the object distance by disparity estimation (Brown and Burschka et al., 2003), the distance estimation for monocular vision system used in this research relies on the geometric information available from the captured image.

The concept of distance estimation reported in Chapter 3.1.2 for camera calibration can also be applied to real-time distance estimation for use in the proposed system.

The distance estimation has to make use of points on the ground plane for correct trigonometric calculation. By substituting the camera mounting height  $h$  to Equation (3.12), the distance  $Z_w$  and  $X_w$  between the camera and the detected object can be estimated as expressed in Equation (3.55) and (3.56). The  $y$ -coordinate of the point on the screen must be larger than the principal point  $c_y$ , otherwise the calculated result is invalid.

$$Z_w = \frac{f_y h}{y_s - c_y} \quad \forall y_s > c_y \quad (3.55)$$

$$X_w = \frac{f_y (x_s - c_x) h}{f_x (y_s - c_y)} \quad \forall y_s > c_y \quad (3.56)$$

With reference to Figure 3-32(a), the bottom corners of the bounding rectangle of the identified vehicle are  $(x_1, y_1)$  and  $(x_2, y_2)$ .  $y_1$  equals to  $y_2$  as they are lying on the same horizontal line. The centre position on the bottom line of the bounding rectangle, which is at  $\left(\frac{x_1+x_2}{2}, y_1\right)$ , is used as the reference point for estimating the distance of the vehicle. The distance can be calculated by Equation (3.55) and (3.56). However, they are only valid when the camera's rotational angles to the ground plane are zero, which is not true when the ego vehicle has motion induced non-zero rotational angle.

A more accurate distance estimation can make use of Equation (3.38), and take only pitch angle  $\theta_x$  into account. This is because the roll angle is the rotation about the Z-axis, its effect on the distance along the Z-axis is minimal. Yaw angle is zero due to the installation of the camera. Further assuming that the pitch angle is small,  $\sin\theta_x$  can be approximated by  $\theta_x$ ,  $\cos\theta_x$  can be approximated as 1. Taking all these assumptions and substituting the installation height  $h$  of the camera to Equation (3.38), it is simplified to Equation (3.57). After expansion, the distance on the World coordinates  $(X_w, Z_w)$  can be calculated by Equation (3.58) and (3.59). Therefore, by substituting  $x_s$  and  $y_s$  by  $\frac{x_1+x_2}{2}$  and  $y_1$  respectively, the position of the vehicle in the World coordinates can be estimated.

$$\begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & -c_x\theta_x & c_x \\ 0 & f_y - c_y\theta_x & f_y\theta_x + c_y \\ 0 & -\theta_x & 1 \end{bmatrix} \begin{bmatrix} X_w \\ h \\ Z_w \end{bmatrix} \quad (3.57)$$

$$Z_w = \frac{f_y + (y_s - c_y)\theta_x}{-f_y\theta_x + y_s - c_y} h \quad (3.58)$$

$$X_w = \frac{(x_s - c_x)(-\theta_x h + Z_w)}{f_x} \quad (3.59)$$

For the speed estimation of a slow relative speed moving vehicle, it can be erroneous if using only one MV at the point  $\left(\frac{x_1+x_2}{2}, y_1\right)$  for ego motion compensation. Since the amplitude of MVs of relative slow moving vehicle is small, a small error in the MV will 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 bounding the detected vehicle to improve accuracy.

Figure 3-34 shows a detected vehicle bracketed in a green rectangle. The image also shows the boundaries of image block of size 8x8. The MVs of blocks, highlighted in red dots in Figure 3-34, along the bottom line of the green rectangle, are read to calculate the position on the World coordinates in the current frame and the previous frame.



Figure 3-34: Speed measurement based on binning of the MVs along the bottom line of the rectangle bracketing the detecting vehicle. Red dots in the image indicate the MV samples for true ground speed evaluation.

Equation (3.58) and (3.59) are 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.

Since the video frame rate and the ego vehicle speed are known, the speed  $u_i$  of each block  $i$  along the bottom green line can be calculated by Equation (3.60), where  $v$  is the

speed of the ego vehicle,  $\Delta t$  is the time interval between two successive frames,  $Z_{W1}$  and  $Z_{W2}$  are the distance along Z-axis of block  $i$  found by Equation (3.58) for the previous frame and current frame respectively.

$$u_i = \frac{Z_{W2} + v\Delta t - Z_{W1}}{\Delta t} \quad (3.60)$$

The deduced speed of each block along the bottom green 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.

### **3.6 Fast Relative Speed Moving Object Detection**

A flow-chart showing the functions of the proposed algorithm for fast relative speed moving object detection is shown in Figure 3-35. The detection of fast relative speed moving object is based on a two-step approach, namely the Hypothesis Generation (HG) and the Hypothesis Verification (HV) steps. The MVs from the H.264/AVC encoder is used in the HG mode for planar parallax residual evaluation. When certain criteria are met, a template for comparison is formed and the algorithm will switch to the HV mode.

In HG stage, a rectangular region inside the image is identified as being potentially having a moving object. The rectangular region is used as a template for matching in the HV stage. During the HV stage, the template is searched for a best match in the successive frames. If a match is found and the displacement of the template is consistent to the previously estimated displacement, a moving object is confirmed, i.e. the yellow box in Figure 3-35. The template is updated with the newly found rectangular area and tracking is performed based on template matching.

This two-step approach is proposed to address the problem of erroneous MVs generated by the H.264/AVC encoder. If the MVs are erroneous in the current frame that are not the result of the actual movement of a moving object, the consistency check on the displacement of the MVs between successive frames during the HV stage will fail. Therefore, the HV stage can reject this kind of erroneous MVs to reduce the false detection rate.

In order to evaluate the planar parallax residual vector (PPRV), ego-motion compensation is required to get the true ground movement of the object involved. The concept on planar parallax residual and the steps involved in HG are described in more details in the following Sections.

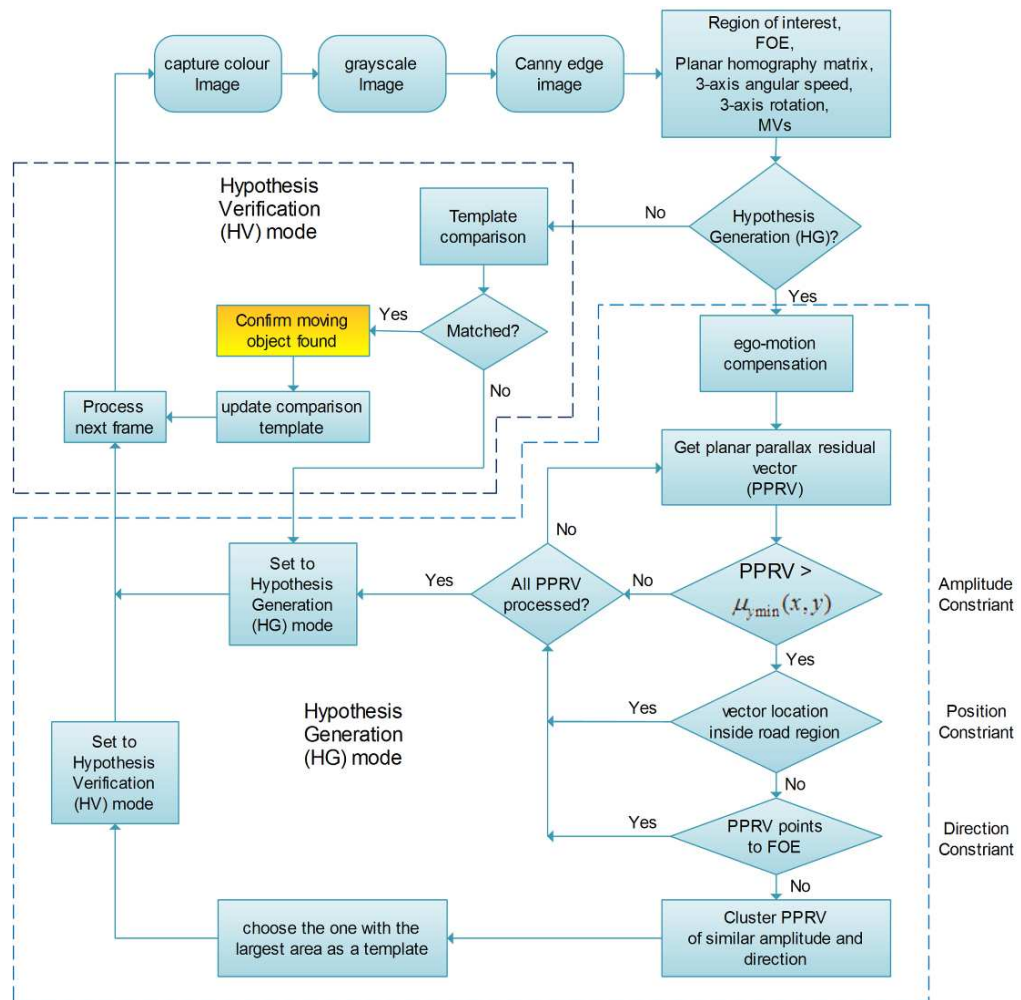


Figure 3-35: Conceptual algorithm flow chart for fast relative speed moving object detection

### 3.6.1 Planar Parallax Residual

As mentioned in Chapter 2.3.1, the amount of planar parallax residual can be used to detect independently moving objects.

Recalling that Equation (3.30) is used to map the pixel position on the ground plane at time  $T_{t-1}$  from its corresponding pixel position at time  $T_t$ , the relationship for points lying outside the ground plane is yet to be described. The relationship of a point on the World coordinates and its corresponding point on the screen can be represented by the planar parallax diagram shown in Figure 2-2 (Baehring and Simon et al., 2005). The green plane is the camera plane at the current frame at time  $T_t$ , the red plane is the camera plane at the previous frame at time  $T_{t-1}$ ,  $P_w$  is a point above the ground plane,  $p_1$  and  $p_2$  are the projected point of  $P_w$  on the image planes at time  $T_{t-1}$  and  $T_t$  respectively,  $P_1^G$  and  $P_2^G$  are the points on the ground plane due to  $P_w$  when viewing by camera at  $C_{t-1}$  and  $C_t$  respectively,  $p_{2G}$  is the point virtually projected to the image plane at  $T_{t-1}$  due to  $P_2^G$  on the ground plane. Substituting  $p_{2G}$  and  $p_2$  to Equation (3.30), Equation (3.61) is obtained.

$$p_{2G} = M^{-1} p_2 \quad (3.61)$$

Therefore, using the planar homography matrix  $M^I$  is able to get the point correspondence of the projected point on the ground plane only, rather than the true point correspondence at  $p_1$  in the previous frame. Since  $p_{2G}$  only maps the projected ground point  $P_2^G$  of  $P_w$  to the screen at time  $T_{t-1}$ , any point  $P_w$  above the ground plane will result in a difference  $\mu = p_{2G} - p_1$  known as planar parallax residual (Baehring and Simon et al., 2005). Referring to the derivation from Baehring et al. (2005) and Trucco et al. (1998), the planar parallax residual  $\mu$  can be expressed as Equation

(3.62), where  $t_d$  is the time duration between successive frames,  $V_x$  and  $V_z$  are the speed of the ego vehicle in X- and Z-axis direction respectively,  $(c_x, c_y)$  is the principal point of the camera,  $(x_0, y_0)$  is the FOE defined in Equation (3.63),  $Z^c$  and  $Z_G^c$  are the Z-coordinate of a point and its corresponding projection point on the ground plane in the camera coordinates,  $p^s = (x^s, y^s)$  is the corresponding point in the screen coordinates of the point  $P = [X^c \ Y^c \ Z^c]^T$  in the camera coordinates.

$$\mu / t_d = \begin{bmatrix} V_z \left( x^s - c_x - \frac{V_x f}{V_z} \right) \left( \frac{1}{Z^c} - \frac{1}{Z_G^c} \right) \\ V_z \left( y^s - c_y - \frac{V_y f}{V_z} \right) \left( \frac{1}{Z^c} - \frac{1}{Z_G^c} \right) \end{bmatrix} = \begin{bmatrix} V_z (x^s - x_0) \left( \frac{1}{Z^c} - \frac{1}{Z_G^c} \right) \\ V_z (y^s - y_0) \left( \frac{1}{Z^c} - \frac{1}{Z_G^c} \right) \end{bmatrix} \quad (3.62)$$

$$\begin{aligned} x_0 &= c_x + V_x f / V_z \\ y_0 &= c_y + V_y f / V_z \end{aligned} \quad (3.63)$$

According to Equation (3.62), MVs lying on the ground plane exhibit zero planar parallax residual because  $Z^c = Z_G^c$ . Also, for stationary objects above the ground plane, the corresponding planar parallax residual vectors (PPRVs) point towards the FOE at  $(x_0, y_0)$ . For a camera mounted at height  $h$  above the ground plane with zero pitch, roll and yaw angle, Equation (3.62) can further be simplified to Equation (3.64), where  $[X \ Y \ Z_w]^T$  are the camera coordinates of a point  $P$ .  $Z_w = Z_c$  in this case because there is no rotational angle and translation difference between the two coordinate systems. The proposed algorithm makes use of Equation (3.64) to derive the threshold of planar parallax residual to determine if there is independently moving object.

$$\mu / t_d = \begin{pmatrix} V_z (x^s - x_0) \left( \frac{h - Y}{h Z_w} \right) \\ V_z (y^s - y_0) \left( \frac{h - Y}{h Z_w} \right) \end{pmatrix} \quad (3.64)$$



For potential collision with the ego vehicle, a static point at  $[X \ Y \ Z_w]^T$  in the camera coordinates will collide with the vehicle at time  $T_{tc}$  evaluated by Equation (3.65)

$$T_{tc} = Z_w / V_z \quad (3.65)$$

Substituting Equation (3.65) to Equation (3.64), a value of planar parallax residual can be expressed as Equation (3.66).

$$\mu_{\min} = \begin{pmatrix} V_z t_d (x^s - x_0) \left( \frac{h - Y}{h V_z T_{tc}} \right) \\ V_z t_d (y^s - y_0) \left( \frac{h - Y}{h V_z T_{tc}} \right) \end{pmatrix} = \begin{pmatrix} t_d (x^s - x_0) \left( \frac{h - Y}{h T_{tc}} \right) \\ t_d (y^s - y_0) \left( \frac{h - Y}{h T_{tc}} \right) \end{pmatrix} \quad (3.66)$$

The amplitude of  $\mu_{\min}$  from Equation (3.66) can be used as the minimum threshold for moving object detection. That is, when the resultant amplitude of the planar parallax residual vector at a particular screen point  $(x^s, y^s)$  is smaller than the amplitude of  $\mu_{\min}$  from Equation (3.66), the corresponding planar parallax residual vector can be ignored.

### 3.6.2 Hypothesis Generation

Hypothesis generation is a process to identify an area in the image that potentially has a moving object. The ROI for relative fast moving object detection is identified by the method mentioned in Chapter 3.4.2, where the ROI is reduced by eliminating the detected road region and the area with small MVs. Ego motion compensated vectors that possess strong planar parallax residual are used to indicate the presence of relatively fast moving objects.

#### 3.6.2.1 Evaluation for Vectors with Strong Planar Parallax

Each MV exported from the H.264/AVC encoder represents an image block of size 8x8. The MVs inside the ROI found by the method mentioned in Chapter 3.4.2 are compensated by the ego motion of the observing vehicle.

### 3.6.2.2 Ego Motion Compensation and Planar Parallax Residual Vector

Given the screen coordinates of a point at  $(x_2, y_2)$  in the current frame, its point correspondence in the previous frame is  $(x_{1G}, y_{1G})$ , estimated by Equation (3.30) and is expressed as Equation (3.67).

$$\begin{bmatrix} x_{1G} & y_{1G} & 1 \end{bmatrix}^T = M^{-1} \begin{bmatrix} x_2 & y_2 & 1 \end{bmatrix}^T \quad (3.67)$$

One point to note is that Equation (3.67) assumes points are lying on the ground plane. Therefore  $(x_{1G}, y_{1G})$  is the corresponding point of the ground plane projection of  $(x_2, y_2)$  in the previous frame.

If the true MV of point  $(x_2, y_2)$  is  $(mv_x, mv_y)$ , the point correspondence in the previous frame is  $(x_1, y_1)$  and their relationship is expressed in Equation (3.68) and (3.69).

$$x_1 = x_2 + mv_x \quad (3.68)$$

$$y_1 = y_2 + mv_y \quad (3.69)$$

According to the definition of planar parallax residual mentioned in Chapter 3.6.1, the planar parallax residual vector is  $\mu = (\mu_x, \mu_y)$ , which is expressed in Equation (3.70) and (3.71).

$$\mu_x = x_{1G} - x_1 \quad (3.70)$$

$$\mu_y = y_{1G} - y_1 \quad (3.71)$$

### 3.6.2.3 Filtering of Planar Parallax Residual Vector

The proposed filtering method makes use of three constraints to retain only useful planar parallax residual vectors (PPRVs) inside the ROI. These three constraints are Amplitude, Position and Direction constraints.

### Constraint 1: Amplitude

After PPRVs inside the ROI are evaluated, some of these PPRVs are erroneous due to various reasons, such as motion estimation error due to the motion estimation algorithm of the H.264/AVC encoder, the use of SKIP mode for motion estimation, occlusion and change of light intensity. These erroneous PPRVs are required to be excluded for more reliable moving object detection.

Equation (3.66) represents the minimum amplitude of planar parallax residual at each point in the screen with time-to-collision taking into account.  $\mu_{\min}(x, y)$  at each point  $(x, y)$  can be calculated by substituting  $Y=0$  and  $T_{tc}=2$  to Equation (3.66).  $Y=0$  means the expected height of the moving object is  $h$ , the same as the mounting height of the camera,  $T_{tc}=2$  means the time to collision for detection is two seconds. Therefore,  $\mu_{\min}(x, y)$  can be expressed as Equation (3.72).

$$\mu_{\min}(x, y) = \begin{pmatrix} \mu_{x\min} \\ \mu_{y\min} \end{pmatrix} = \begin{pmatrix} \frac{t_d(x - x_0)}{2} \\ \frac{t_d(y - y_0)}{2} \end{pmatrix} \quad (3.72)$$

One point to note is that a longer  $T_{tc}$  will result in a smaller  $\mu_{\min}(x, y)$ , and an object of height smaller than the camera mounting height will also result in a smaller  $T_{tc}$ . Since the MVs from the H.264/AVC encoder are block based with limited precision,  $\mu_{\min}(x, y)$  cannot be too small, or otherwise there will be too many false detections.

Similarly, the value of  $\mu_{\min}(x, y)$  near the FOE  $(x_0, y_0)$  is also very small as the time duration  $t_d$  between successive frames is only 66ms. Therefore, the detection criteria of PPRVs with strong planar parallax for moving object detection are the amplitude of the PPRV at  $(x, y)$  is larger than  $\mu_{\min}(x, y)$  and a threshold  $\mu_{thres}$ .

## Constraint 2: Position

Since the PPRVs represented by Equation (3.70) and (3.71) are the direction of motion of the screen coordinates  $(x_2, y_2)$  at the current frame after ego motion compensation, it can be used to estimate the position of its corresponding screen coordinates after a time period  $T$ . If an area in the image is defined as an alert zone and a PPRV enters into the zone after time  $T$ , the corresponding object can be regarded as entering the dangerous area after time  $T$  which may collide with the ego vehicle.

Figure 3-36 shows a screen shot of a vehicle moving from the left to the right. A point at  $(x_2, y_2)$  has a resulting planar parallax residual vector of  $(\mu_x, \mu_y)$ .  $T_y$  is the time for the y-component of the PPRV  $\mu_y$  to travel from position  $y_2$  to the top y-axis position  $Y_u$  of the alert zone, and is calculated by Equation (3.73). The corresponding x-coordinate at  $x_a$  of  $(x_2, y_2)$  after time  $T_y$  is calculated by Equation (3.74).  $t_d$  is the time duration between successive frames.  $\mu_y$  must be positive for it to become nearer to the ego vehicle.

$$T_y = \frac{(Y_u - y_2)t_d}{\mu_y} \quad (3.73)$$

$$x_a = x_2 + \frac{\mu_x T_y}{t_d} \quad (3.74)$$

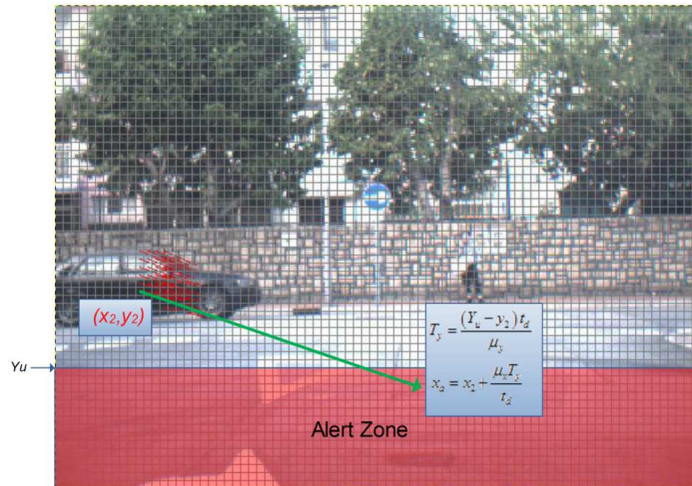


Figure 3-36: Illustration of MV position after time  $T_y$

Therefore, if the position of  $x_a$  is also inside the alert zone, the PPRV is accepted for further processing. Otherwise, the PPRV is discarded.

### Constraint 3: Direction

After examining the amplitudes of PPRVs and their positions after time  $T_y$ , another detection criterion is the direction of the PPRVs. As mentioned in Chapter 3.6.1, objects with PPRVs pointing to the FOE are moving in parallel to the ego vehicle or belonging to static objects. They can be vehicles with slow relative speed to the ego vehicle travelling in parallel, or be stationary objects on the road. Nevertheless, PPRVs pointing to the FOE should be excluded. This is because the ROI has eliminated objects with small amplitudes due to relatively slow moving speed vehicles, and these PPRVs are highly likely to belong to static objects.

A PPRV at point  $(x,y)$  can be excluded if the slope between the PPRV and the slope of the point to the FOE are smaller than a threshold  $m_{thres}$ .

#### **3.6.2.4 Clustering of Planar Parallax Residual Vector**

After having determined valid PPRVs, they are further clustered to represent different objects. Since PPRVs on the same objects should have similar amplitude and direction, the clustering of PPRVs can be done by comparing the amplitude, direction and distance of PPRVs to some thresholds.

For a PPRV  $\mu_j$ , where  $j$  is an integer smaller than the total number of PPRVs available for clustering, is compared to the mean amplitude and direction of all clusters, starting from cluster  $C_1$ .  $\mu_j$  is grouped to cluster  $C_i$ , where  $i$  is an integer with value smaller than the total number of clusters, if it meets all the three constraints listed in Table 3-1, where  $\bar{m}(C_i)$  and  $\bar{a}(C_i)$  are the mean slope and mean amplitude of the cluster  $C_i$ ,  $m(\mu_j)$  and  $a(\mu_j)$  are the slope and amplitude of PPRV  $\mu_j$  respectively.

Otherwise, a new cluster is created to store  $\mu_j$ . The mean amplitude  $\bar{a}(C_i)$  and slope  $\bar{m}(C_i)$  of all PPRVs in a cluster  $C_i$  are updated if the size of the cluster is changed.

Table 3-1: The three constraints for clustering decision for PPRVs  $s_j$ .

“D” Direction constraint	Slope comparison. $m(\mu_j)$ and $\bar{m}(C_i)$ differ by less than the threshold $m_{diff}$ .
“A” Amplitude constraint	Amplitude comparison. $a(\mu_j)$ and $\bar{a}(C_i)$ differ by less than the threshold $a_{diff}$ .
“S” Spatial constraint	Distance comparison. $\lambda_{ci}(j,k)$ is smaller than a threshold $\lambda_{thres}$ . The distance $\lambda_{ci}(j,k)$ between any point $\mu_j(x,y)$ inside the cluster $C_i$ to the point at $\mu_k(x,y)$ , denoted by $\lambda_{ci}(j,k)$ where $j$ is an integer with value smaller than the size of cluster $C_i$ , $k$ is an integer with value smaller than the number of PPRVs, i.e. $\lambda_{ci}(j,k) = \sqrt{(\mu_j(x) - \mu_k(x))^2 + (\mu_j(y) - \mu_k(y))^2}$ .

### 3.6.2.5 Cluster Refinement

After all PPRVs are clustered, the maximum size of a bounding rectangle that can include all the points in the cluster is evaluated. The coordinates of the bounding rectangle are also stored in this process.

The refinement is started from the bottom side of the bounding rectangle. The bottom row of blocks of the rectangle, each of size 8x8, is examined for “homogeneity”. Homogeneity is a measure to determine if the texture in a block is rich. A simple method is used in this algorithm by summing the corresponding area of the block inside the Canny edge image obtained by applying Canny edge filter to the grey-scale image converted from the captured colour image. If the sum is smaller than a threshold, the block is regarded as having weak texture. A block is marked as ‘invalid’ either if it is having weak texture or is a block of the road (recalling that road region is found using the method mentioned in Chapter 3.3). If the number of ‘invalid’ blocks exceeds two-third of the total number of blocks along the bottom row of the rectangle, the

whole row at the bottom of the rectangle is excluded from the cluster. Then, the next bottom row is compared for homogeneity. A maximum of eight rows can be excluded in this process.

The top-side refinement is simply performed by extending the top side of the rectangle to eight pixels below the FOE. This is because moving objects in front of the ego vehicle will appear to have vertical span beyond the y-axis of the FOE.

Similar to the refinement done towards the bottom side of the rectangle, the blocks at the left and right sides are also compared for homogeneity. If the total number of invalid blocks on one side is larger than two-third of the total number of blocks on the side of the rectangle, the column of blocks on one side is excluded from the cluster.

Finally, the size of the bounding rectangle is updated according to the refinement result.

#### **3.6.2.6 Cluster Selection**

After all PPRVs are clustered and refined, the clusters are checked for overlapping. Overlapping clusters are removed from the cluster list leaving only the cluster of the largest area among the overlapped clusters. Among all remaining clusters after overlapping cluster removal, only the cluster with the largest area will be kept for HV in the proposed algorithm.

#### **3.6.2.7 Template Registration**

For the selected cluster with the largest area, the image inside the cluster is cropped for being used in the Hypothesis Verification stage. The corresponding position  $p_{C_i}^s = (x_{C_i}, y_{C_i})$  in the image, mean amplitude  $\bar{a}(C_i)$  and direction  $\bar{m}(C_i)$  of the MVs of the selected cluster are also stored for comparison purposes in the Hypothesis Verification stage.

### 3.6.3 Hypothesis Verification

For the selected cluster in the HG mode, the mean MV amplitude  $\bar{a}(C_i)$  and direction  $\bar{m}(C_i)$  are used as the target displacement of the template to be matched in the next captured frame. The target displacement is expressed in Equation (3.75).

$$s_t = \begin{bmatrix} \bar{a}(C_i) \cos(\bar{m}(C_i)) \\ \bar{a}(C_i) \sin(\bar{m}(C_i)) \end{bmatrix} \quad (3.75)$$

A search window of size 16x16 is defined as the search range around the position indicated by the target displacement  $s_t$  and direction of the selected cluster, as expressed in Equation (3.76).

$$p_{j,k}^s = p_{C_i}^s + s_t + \begin{bmatrix} j \\ k \end{bmatrix} \quad \forall j, k \in \mathbb{Z}\{-16:+16\} \quad (3.76)$$

The search is performed spirally inside the search window. Sum of Absolute Difference (SAD) between the template stored from the last frame and a candidate template in the search window is used to indicate a potential match. The template matching is successful if the SAD is within a predefined threshold and a local minimum is found inside the search window. The resultant displacement  $s_r = (u, v)$  of the match template from the initial position of the cluster at  $p_{C_i}^s$  is compared with the target displacement  $s_t$ .

When a match is found and the percentage difference between the resultant displacement and the target displacement is smaller than a threshold, the hypothesis verification is successful. A moving object is therefore identified. This process reduces the number of false detection due to the inaccuracy of H.264/AVC MVs. The resultant displacement  $s_r$  of the template is stored as the target displacement in the successive frame for tracking purposes. The bottom row of the template is also expanded for one



more row, in order to account for potential change in dimension of the selected object on the screen due to the perspective change of the moving object.

### **3.6.4 Tracking**

After successful Hypothesis Verification, the size of the template is refined by the method mentioned in Chapter 3.6.2.5. The image inside the template is cropped for being compared with the successive captured images. Being the same as the Hypothesis Verification stage, template matching with 16x16 search window around the target displacement  $s_r$  continues to be used. The tracking is successful if the percentage difference between the resultant displacement and the target displacement is within a defined threshold. The tracking mode is only used for one successive frame after the Hypothesis Verification stage. This is to make sure the system can be responded more swiftly to the change of scene so that new moving objects can be detected and tracked.

The template matching method is proposed as one of the simplest methods for identifying similar pattern in the successive frames for hypothesis verification and tracking. The advantage of using a simple template matching method is its low computational cost that fulfils the real-time processing requirement.

However, the template matching method cannot account for significant perspective change of the target object effectively. That is, the matching may fail if the target has a significant perspective change in successive frames. Since the time duration between successive frames is only 66ms, the perspective change of the target object is assumed to be small, therefore template matching is regarded as an effective method for hypothesis verification given the assumption is valid.

### **3.7 Chapter Summary**

This Chapter has described the proposed algorithm framework in details. It also mentioned the proposed technique on camera calibration which is one of the important preparation steps for successful application of the proposed algorithm. The algorithm is proposed based on the identified problems on using H.264/AVC MVs for moving object detection, and the requirement on the real-time performance for being used as an ADAS. The proposed algorithm includes the technique on ego motion estimation, road region detection, segmentation of regions of interest, slow relative speed moving object detection, and fast relative speed moving object detection.

## 4 Test and Evaluation

A series of tests were performed to evaluate the effectiveness of the proposed algorithm framework for moving object detection.

### 4.1 Evaluation of Camera Calibration Results

The goal of the calibration process was to obtain an accurate estimation of the point correspondence in the World coordinates from the screen coordinates. The accuracy of distance estimation is the parameter of the performance of the calibration method.

#### 4.1.1 Focal Lengths and Principal Point Estimation

Figure 3-5 shows the setup for intrinsic parameter calibration. The setup consisted of a computer and software to capture the image from the camera, a flat checker board pattern fixing vertically to an up-right board, and a laser distance checker. The computer software was able to overlay horizontal and vertical lines on the screen for easier alignment of the checker pattern to the desired positions. The checker pattern shown in Figure 3-7 was used.

Figure 3-7 also shows the physical dimension and screen coordinates of the checker board pattern printed on the paper. The checker pattern was created with the dimension of  $D_{ab}$  and  $D_{ac}$  equal to 0.447m and 0.894m respectively. The distance of the checker board from the camera was measured as 4.582m. The left- and right-side distances of the checker board to the camera were both 4.604m. Since the difference in the left- and right-side distance was almost zero, the checker board was regarded as being placed perpendicularly to the camera.

Sub-pixel search for the corners at control points A, B and C shown in Figure 4-1 was performed by the calibration program. The values of these control points are shown in Table 4-1.

Table 4-1: The control points values for calibration.

Control point	Screen Coordinates
$A = (u_a, v_a)$	(266.973, 205.074)
$B = (u_b, v_b)$	(372.417, 204.913)
$C = (u_c, v_c)$	(266.261, 417.258)
$H_d = u_b - u_a$	105.444
$V_d = v_c - v_a$	212.184

Recalling Equation (3.13), the focal lengths  $f_x$  and  $f_y$  were calculated as 1080.853 and 1087.504 respectively.

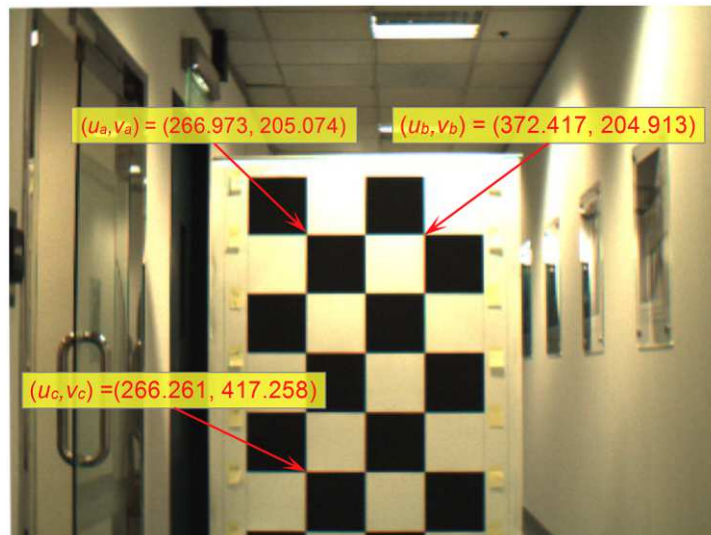


Figure 4-1: Sub-pixel coordinates of control point A, B and C on the upright board

For simplicity reasons during the experiment, the principal point was set to the centre of the image. Sirisantisamrid et al. (2011) has shown that the influence of principal point on the calibration result is minimal when the lens distortion of the camera is small. Since the resolution of the selected camera is 640x480, the centre of the camera was at (319.5, 239.5).

### 4.1.2 Intrinsic Parameter

With the focal lengths and principal point available, the camera intrinsic parameter could be expressed as the 3x3 matrix  $K$  shown in Equation (4.1). Since the difference in focal length  $f_x$  and  $f_y$  is very small, the average value of  $f_x$  and  $f_y$ , which is 1084.179, could be used as the overall focal length  $f$  for the camera.

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1080.853 & 0 & 319.5 \\ 0 & 1087.504 & 239.5 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.1)$$

### 4.1.3 Camera Installation

Figure 4-2 shows the captured screen after the camera was installed correctly. The correct installation is indicated by the aligned horizontal and vertical lines to the edges of checkers on the screen, and the near-to-zero roll and pitch angles read by the computer software.



Figure 4-2: Captured image with aligned horizontal and vertical line. The roll and pitch angles are almost zero.

#### 4.1.4 Extrinsic Parameter Estimation

The extrinsic parameters were estimated using the method mentioned in Chapter 3.1.2. Since the camera was installed carefully so that the rotational angles are all nearly zero, the extrinsic parameters in this case include only the installation height  $h$  of the camera. The screen coordinates of checker corner points labelled in Figure 3-12 are shown in Table 4-2. The physical dimension of each square in the banner was  $d$ , which was equal to 0.5m in this setup.

By using Equation (3.23), the camera installation height  $h$  was estimated as 1.08m.

Table 4-2: Screen coordinates of checker corner points

Point label	Screen Coordinates	Sub-pixel value
A <sub>c</sub>	$(x_a, y_a)$	(319.229, 444.131)
B <sub>c</sub>	$(x_b, y_b)$	(319.200, 427.001)
P <sub>1</sub>	$(x_1, y_1)$	(416.442, 443.763)
P <sub>2</sub>	$(x_2, y_2)$	(222.503, 444.500)
P <sub>3</sub>	$(x_3, y_3)$	(408.627, 426.187)
P <sub>4</sub>	$(x_4, y_4)$	(230.384, 427.522)

By using Equation (3.20), the distance  $Z_I$  between the camera and the checker point P<sub>1</sub> and P<sub>2</sub> can also be calculated. The estimated value of  $Z_I$  was 5.57m in this setup.

Since the installation height  $h$  of the camera has been calculated and the installation method has set the rotation angles to nearly zero (smaller than 0.1 degree in the experiment), the extrinsic parameters of the camera can be represented by the 3x4 matrix shown in Equation (4.2).

$$[R_w \mid t] = \left[ \begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1.08 \\ 0 & 0 & 1 & 0 \end{array} \right] \quad (4.2)$$

### 4.1.5 Distance Accuracy Evaluation

With the intrinsic and extrinsic parameters found, a point on the screen coordinates could be related to its corresponding point in the World coordinates according to Equation (3.11). Equation (3.11) is expanded and shown in Equation (4.3). Since points are lying on the ground,  $Y_w$  can be set to zero. Then Equation (4.3) is modified to Equation (4.4), where  $M$  is the planar homography matrix.

$$\begin{aligned}
 p^s = [x_s \quad y_s \quad 1]^T &= K [R_w \mid t] [X_w \quad Y_w \quad Z_w \quad 1]^T \\
 &= \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & h \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \\
 &= \begin{pmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & f_y h \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}
 \end{aligned} \tag{4.3}$$

$$p^s = \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & c_x & 0 \\ 0 & c_y & f_y h \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_w \\ Z_w \\ 1 \end{bmatrix} = M \begin{bmatrix} X_w \\ Z_w \\ 1 \end{bmatrix} \tag{4.4}$$

So, the corresponding point on the World coordinates is evaluated by Equation (4.5) which is obtained by multiplying  $M^{-1}$  on both sides of Equation (4.4).

$$\begin{aligned}
 [X_w \quad Z_w \quad 1]^T &= M^{-1} [x_s \quad y_s \quad 1]^T \\
 &= \begin{bmatrix} f_x & c_x & 0 \\ 0 & c_y & f_y h \\ 0 & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} x_s \\ y_s \\ 1 \end{bmatrix}
 \end{aligned} \tag{4.5}$$

$M^{-1}$  in this case is equal to that shown in Equation (4.6). It has been normalised for easier matrix multiplication at later times.

$$M^{-1} = \begin{bmatrix} -4.53714 \times 10^{-3} & 0 & 1.44961 \\ 0 & 0 & -4.90398 \\ 0 & -4.17537 \times 10^{-3} & 1 \end{bmatrix} \tag{4.6}$$

The matrix  $M^{-1}$  is then used to estimate the World coordinates of points on the ground surface. The error comparing to the World coordinates obtained from physical measurement is shown in Table 4-3. For points that were farer away from the camera, the estimation error was larger. This is because the unit of distance representing by each pixel for far away objects is larger, leading to larger estimation error.

Nevertheless, the accuracy for distance of control points can be kept below 3.5%, which is within the expected accuracy of the system.

Table 4-3: The deviation of calculated World coordinates from the measured World coordinates

Screen Coordinates	Measured World Coordinates (X,Z)	Calculated World Coordinates (X,Z)	Estimation Error (Z- direction)
$(x_a, y_a) = (319.229, 444.131)$	(0, 5.57)	(-0.0014, 5.74)	3.05%
$(x_b, y_b) = (319.200, 427.001)$	(0, 6.07)	(-0.0017, 6.26)	3.13%
$(x_1, y_1) = (416.442, 443.763)$	(0.5, 5.57)	(0.516, 5.75)	3.23%
$(x_2, y_2) = (222.503, 444.500)$	(-0.5, 5.57)	(-0.514, 5.73)	2.87%
$(x_3, y_3) = (408.627, 426.187)$	(0.5, 6.07)	(0.518, 6.29)	3.62%
$(x_4, y_4) = (230.384, 427.522)$	(-0.5, 6.07)	(-0.515, 6.25)	2.97%

#### 4.1.6 Comparison to Zhang's Method

Zhang's calibration method (Zhang, 2000) is a well known method for generic calibration of cameras. The method proposed in this study is a simplified version of Zhang's method. Zhang's method implemented in OpenCV was evaluated for comparison to the result obtained by using the new proposed method. Figure 4-3 shows nine captured images for input to Zhang's algorithm for calibration.

Since the camera's principal point is assumed to be at the centre of the screen, the calibration program from OpenCV was run with fixed principal point. The result of the



estimated intrinsic matrix was  $\begin{bmatrix} 1094.04272 & 0 & 319.5 \\ 0 & 1093.78186 & 239.5 \\ 0 & 0 & 1 \end{bmatrix}$ . The values of  $f_x$

and  $f_y$  shown in Chapter 4.1.2 found by the proposed method was very close to that found by Zhang's method. They differ by less than 1.3%.

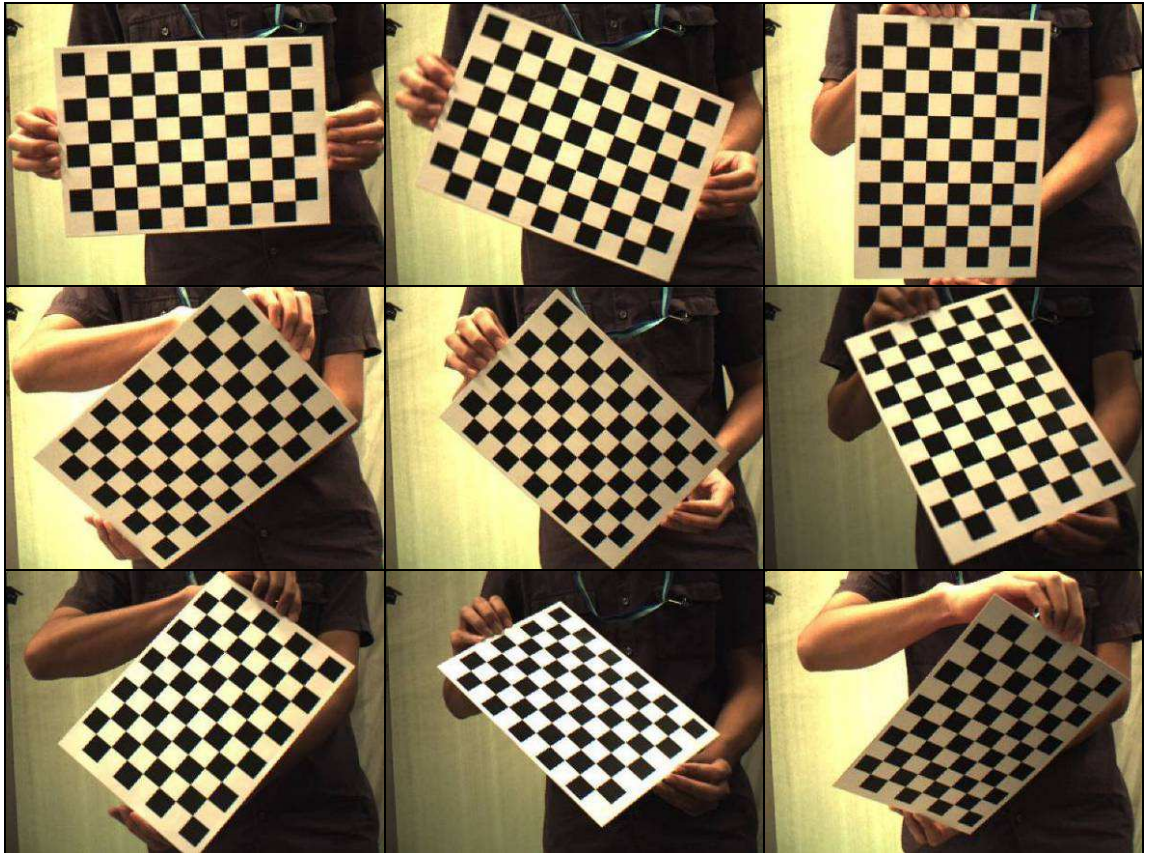


Figure 4-3: Checker board images used for the estimation of intrinsic parameters of the camera using Zhang's method.

For the extrinsic parameter estimation using Zhang's method, the image shown in Figure 4-2 was used. The OpenCV program was used for the evaluation. The determined rotation angles in radian about X-, Y- and Z-axis were 0.0095813, -0.0012957 and -0.0085325 respectively. These results were close to the ideal rotation angles of zero. The translations in meter from X-, Y- and Z-axis were  $4.47 \times 10^{-6}$ , 1.05, and 5.09 respectively. Therefore, the estimated mounting height of the camera was

1.05m which was the translation along the Y-axis. Similarly, the distance of the first control point  $A_c$  on the banner was 5.09m, which was the translation along Z-axis.

Compared to the results with the proposed method, the deviation in the estimated camera height was 2.86%. The deviation in the estimated distance to the control point  $A_c$  was 8.62%. The intrinsic and extrinsic parameters estimated by Zhang's method were used to calculate the world coordinates from the supplied screen coordinates. The result shown in Table 4-4 revealed that the estimated translation of 5.09m along the Z-axis was not so accurate as the estimation error was larger than 12% for all points. Although the result from the proposed calibration method was close to the result obtained by Zhang's method, it is more preferable to use the proposed method for more accurate estimation of World coordinates from the screen coordinates. In addition, the proposed method has addressed the problem of installing the camera correctly into a vehicle.

Table 4-4: Deviation of calculated World coordinates from the measured World coordinates using Zhang's method

Screen Coordinates	Measured World Coordinates (X,Z)	Calculated World Coordinates (X,Z)	Estimation Error (Z- direction)
$(x_a, y_a) = (319.229, 444.131)$	(0, 5.09)	(-0.0014, 5.77)	13.36%
$(x_b, y_b) = (319.200, 427.001)$	(0, 5.59)	(-0.0017, 6.30)	12.70%
$(x_1, y_1) = (416.442, 443.763)$	(0.5, 5.09)	(0.512, 5.78)	13.56%
$(x_2, y_2) = (222.503, 444.500)$	(-0.5, 5.09)	(-0.510, 5.76)	13.16%
$(x_3, y_3) = (408.627, 426.187)$	(0.5, 5.59)	(0.515, 6.33)	13.24%
$(x_4, y_4) = (230.384, 427.522)$	(-0.5, 5.59)	(-0.512, 6.28)	12.34%

## 4.2 Verification of Ego Motion Compensation

To verify the ego motion compensation algorithm, a simple synthesized sequence was produced to serve the purpose. Figure 4-4 shows four image frames of a simple image sequence. The frame to frame duration was 160ms. The sequence was synthesized with an ego vehicle moving at 20m/s. A moving object of physical size 2m x 2m that was represented by the square at the centre of the screen was moving forward at 10m/s. A house on one side of the road with physical dimension shown in Figure 4-5 was located at 5.5m left from the centre line. It was at 80m from the ego vehicle at the beginning. The camera focal length and principal point were 830 and (320, 240) respectively. The mounting height of the camera was 1.26m with zero rotational angles.

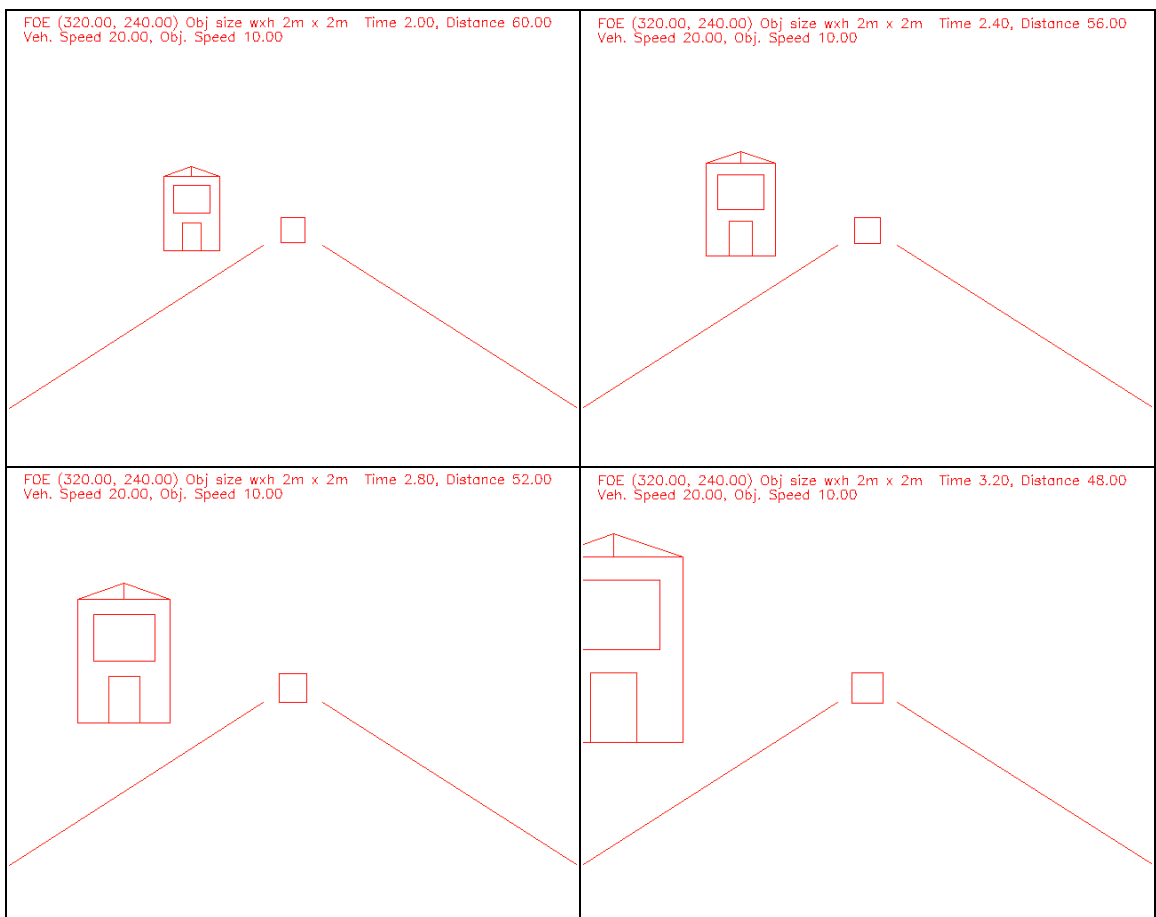


Figure 4-4: Simple image sequence containing a static (the house) and a moving object (the square box).

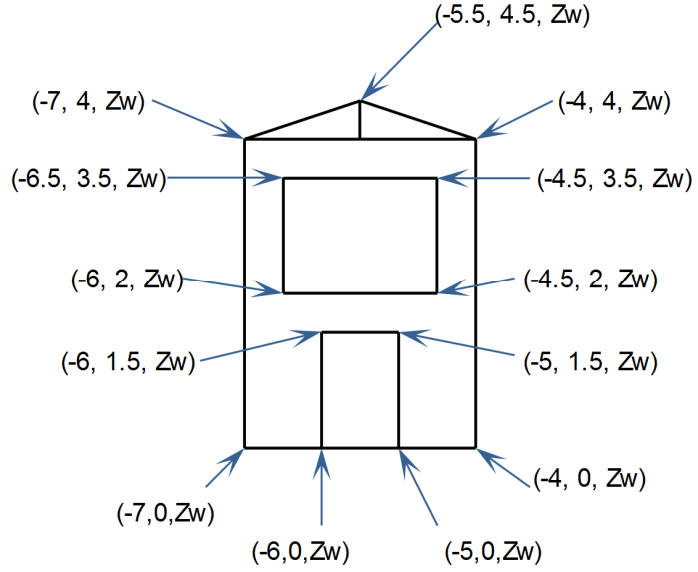


Figure 4-5: The physical dimension of the house in the simple synthesized sequence. The World coordinates of vertices of the house are shown. Values are shown in meter.  $Z_w$  is the distance of the house from the ego vehicle. Negative X-axis value means the object is on the left-hand side of the World coordinates.

The equation concerned for ego motion compensation is shown in Equation (3.30) in Chapter 3.2.1. It relates a point on the ground plane in the screen coordinates between the current and the previous frames. The evaluation was performed by examining the screen coordinates of selected points of successive frames, comparing the difference in the calculated screen coordinates and the actual observed coordinates. The matrix  $A$  in Equation (3.30) is evaluated in Equation (5.6) by substituting the parameters from Equation (5.2) to (5.5). Then matrix  $M^{-1}$  can be found by Equation (5.6) and (5.1) as shown in Equation (5.7).

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 830 & 0 & 320 \\ 0 & 830 & 240 \\ 0 & 0 & 1 \end{pmatrix} \quad (5.1)$$

$$T_c = [0 \quad 0 \quad Veh\_spd \cdot T_d]^T = [0 \quad 0 \quad 20 \cdot 0.16]^T = [0 \quad 0 \quad 3.2]^T \quad (5.2)$$

$$R_w = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I \quad (5.3)$$

$$R_c = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = I \quad (5.4)$$

$$n^T = (0 \ 1 \ 0) \quad (5.5)$$

$$\begin{aligned} A &= R_c \left( I - \frac{R_w T_c n^T}{h} \right) \\ &= I \left( I - \frac{1}{1.26} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3.2 \end{pmatrix} (0 \ 1 \ 0) \right) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2.5397 & 1 \end{pmatrix} \end{aligned} \quad (5.6)$$

$$\begin{aligned} M &= KAK^{-1} \\ &= \begin{pmatrix} 830 & 0 & 320 \\ 0 & 830 & 240 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2.5397 & 1 \end{pmatrix} \begin{pmatrix} 0.0012048 & 0 & -0.3855422 \\ 0 & 0.0012048 & -0.2891566 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} 1 & -0.9791547 & 234.99713 \\ 0 & 0.265634 & 176.24784 \\ 0 & -0.0030599 & 1.7343660 \end{pmatrix} \end{aligned} \quad (5.7)$$

Figure 4-6 shows two successive frames of the synthesized sequence. The lower right corner of the house, which is a point on a static object in the sequence, was used to verify the equation for ego motion compensation. As seen from the current frame shown in Figure 4-6(b) the screen coordinates of the lower right corner of the house is (261, 259). It is substituted to Equation (3.30) to find the corresponding point in the previous frame. The calculated point correspondence in the previous frame was (264.24, 257.96), as shown in Equation (5.8). The calculated result is close to the actual coordinates at (264, 258) recognising directly from the image. Therefore, the equation for ego motion compensation is verified.

$$p_{t-1}^s = M^{-1} p_t^s = \begin{bmatrix} 1 & 0.979155 & -234.9971 \\ 0 & 1.734366 & -176.2478 \\ 0 & 0.003060 & 0.265634 \end{bmatrix} \begin{bmatrix} 261 \\ 259 \\ 1 \end{bmatrix} = \begin{bmatrix} 264.24 \\ 257.96 \\ 1 \end{bmatrix} \quad (5.8)$$

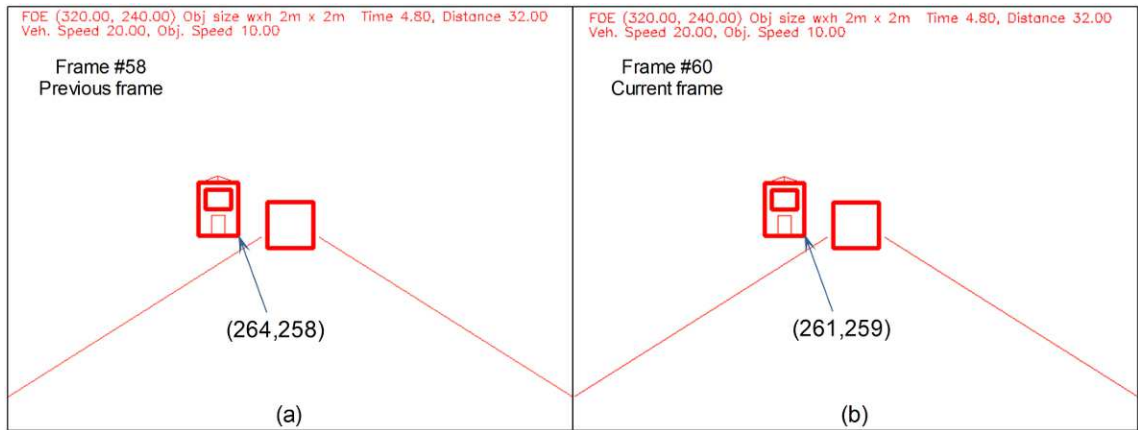


Figure 4-6: Two successive frames of the synthesized sequence. The coordinates of the lower right corner of the house is read from the image. The coordinate reading for the previous frame is compared with the calculated result.

### 4.3 Slow Relative Speed Vehicle Detection and Tracking Method

The effectiveness of the proposed slow relative speed vehicle detection algorithm and its computational speed are summarised in this Chapter.

#### 4.3.1 Region of Interest Formation

The region of interest (ROI) in the captured image is the area outside the detected road region and having small MV amplitude. Figure 4-7 shows the sequence of images on the formation of the ROI for relative slow speed vehicle detection. Figure 4-7(a) is the original captured image. The road region in front of the ego vehicle is identified and shown in white colour blocks in Figure 4-7(b). The white blocks in Figure 4-7(c) represent those with MV amplitude larger than a threshold  $q_m$ .  $q_m$  is set to 12 according to the experiment results. Figure 4-7(d) shows the result of combined road region mask in Figure 4-7(b) and the MV region mask in Figure 4-7(c). It is noticed that there are many small black blocks surrounded by the white mask. These small black blocks are removed from the image mask by a hole-filling algorithm mentioned in Chapter 3.3.4. The result is shown in Figure 4-7(e). The resulting ROI with the upper part of the

image cropped is shown in Figure 4-7(f). Therefore, only the lower half of the image with areas outside the image mask would be evaluated by the algorithm for relatively slow moving objection detection.

The example shown in Figure 4-7 is a typical scenario on the road. The image area to process is less than 20% of the original image size. Therefore the reduced ROI can lower the computational time to help achieve the real time performance requirement of the system.

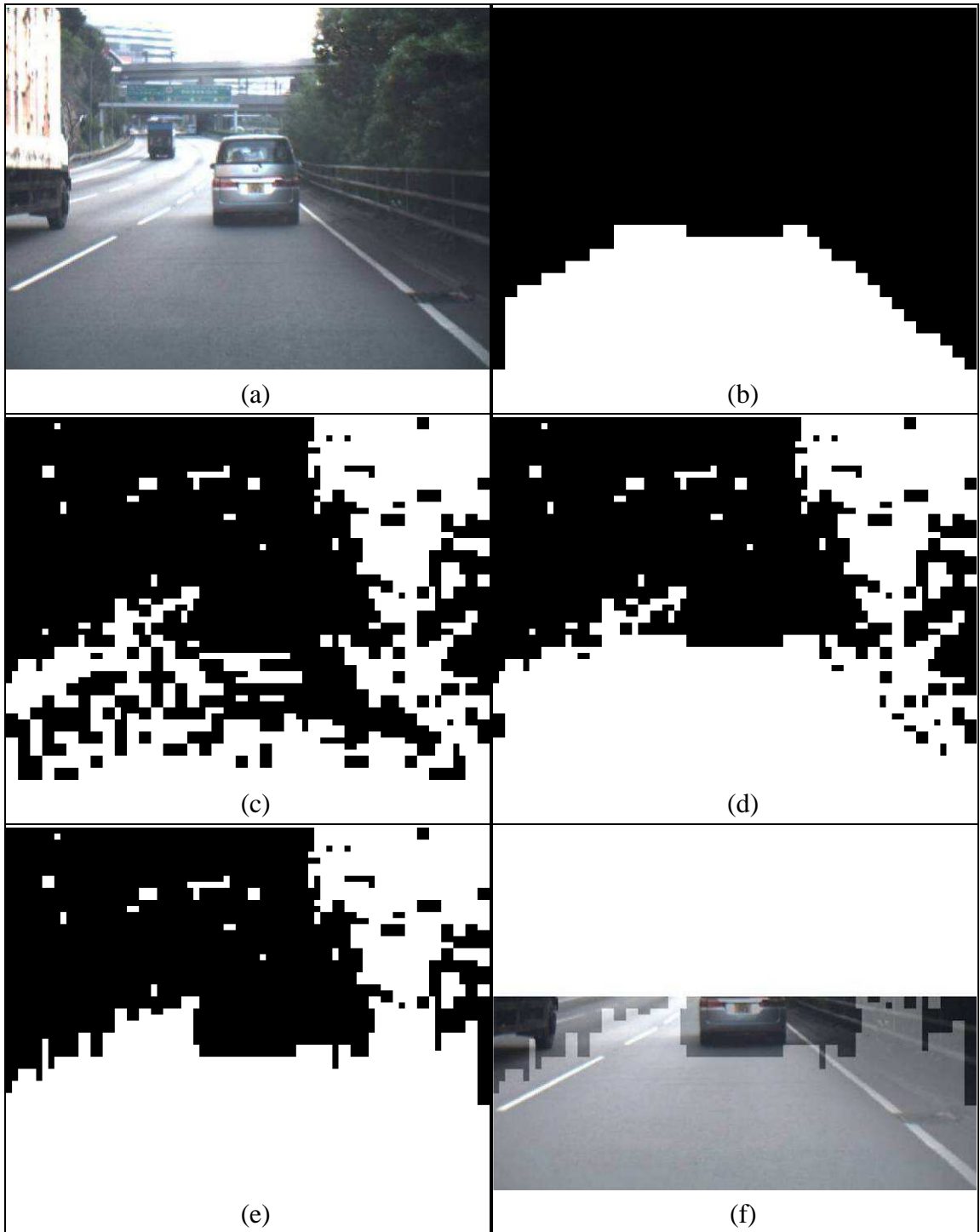


Figure 4-7: Illustration of road region detection and exclusion of areas with large MV amplitude. (a) Original captured image. (b) Result of block based road region detection. White colour represents the detected road region. (c) Blocks with MV amplitude larger than a threshold with white colour. (d) Combined mask from road region and large MV areas. (e) Combined region mask with holes filled. (f) Final ROI overlaid to the original image. ROI is the regions outside the white colour mask.



### 4.3.2 Detection and Tracking

The detection algorithm was evaluated against different kinds of vehicles. This included passenger cars, minivans, buses and trucks. Figure 4-8 shows some of the vehicles that were detected in the video sequences captured from different roads in Hong Kong. The rectangular bounding boxes are able to locate the bottom and the sides of the detected vehicles. The upper bound of the rectangle is obtained by the aspect ratio of the rectangle or the  $y$ -coordinate of the FOE. Therefore, the computation time for finding the upper bound of the vehicle can be reduced. It is also noticed from the examples shown in Figure 4-8 that the light intensity of the scenes varies quite significantly. Even if there are shadows and strong sunlight on the road, the vehicles can still be detected.

The threshold values for successful vehicle detection are listed in Table 4-5.  $q_m$ , the threshold for segmenting the region of interest according to the amplitude of MVs, was determined with the assumption that relatively fast speed moving objects have relatively large MV amplitudes. It was set to 12, which is 1.5 times of the block size used. A larger value of  $q_m$  will mean increasing the region of interest for relatively slow speed moving object detection, and vice versa.  $D_{hv}$  is the threshold for comparing point correspondence in the vertical gradient and horizontal gradient images in order to eliminate non-horizontal contours in the vertical gradient image. The main noise to eliminate in a vertical gradient image is the lane markings on the road. This parameter should be adjusted if the focal length of the camera is changed. For instance, if a camera with smaller focal length is used, the camera field of view will be increased. Lane markings will appear to be more 'horizontal'. This means the lane markings will have larger amplitudes in the vertical gradient image, A smaller value of  $D_{hv}$  should be used in this regard.  $W_U$  and  $W_L$  are the upper and lower limits of the width of vehicles

to be detected. They were determined according to the actual width of vehicles that will appear on the road. Similarly,  $R_{WHU}$  and  $R_{WHL}$  are the upper and lower limits of the width to height ratio of the detected vehicle. These values should be set according to the actual width and height of vehicles appearing in the image. A smaller range will reduce the true-positive and false positive rates.  $e$  is the parameter that defines the range of x-coordinates for evaluating the vertical projection near the end points of the horizontal line at the bottom of the detected vehicle. This parameter was set according to the observation that most vertical edges of vehicles would fall within this search range. A larger value of this parameter will increase the search range. This may include more vertical edges that are not belonging to the detected vehicles. On the other hand, a smaller value of this parameter may be insufficient to include the vehicle edges in the search range, leading to reduced detection rate.  $V_h$  is the threshold to confirm that a vehicle exists if it is smaller than the average vertical gradient in a bounded rectangle. This value is determined by observations in experiments conducted. A smaller value will result in increased false positive rate.  $e_x$  and  $e_y$  are number of pixels to expand the bounding rectangle in the x- and y-coordinate of the screen respectively for tracking of detected vehicle in successive frames. They were determined according to observations during experiments with the ego-vehicle travelling in straight line, following a vehicle at the front at constant speed. Increasing these values will increase the search range, and hence increasing the chance of successful tracking. But this will also increase the computation time during tracking. On the other hand, more false positive tracking is expected due to the inclusion of more sources of interference in the increased search range.

Although reducing the ROI by MV amplitude and road region detection can successfully reduce the computation time for vehicle detection, the detection time is

still affected by how many vehicles are identified in each captured image. Since the detected vehicle will not disappear immediately in successive frames, a tracking algorithm that largely limits the search window can reduce the computational cost.

Table 4-5: The parameters used in the algorithm for successful detection of vehicles. These parameters are determined by repeated testing to the video sequences taken in Hong Kong for this project.

	<b>Parameter</b>	<b>Description</b>	<b>Value</b>
<b>1</b>	$q_m$	Parameter mentioned in Chapter 3.4.1. It is the threshold in number of pixels for segmentation of the image to regions for relatively slow and fast speed vehicle detection.	12
<b>2</b>	$D_{hv}$	Parameter mentioned in Chapter 3.5.3. It is for the comparison between vertical gradient image and horizontal gradient image for the elimination of unwanted vertical contours.	50
<b>3</b>	$W_L$ and $W_U$	Parameter mentioned in Chapter 3.5.3. They are predefined lower and upper limits of the width of detected horizontal line. The width must be within $W_L$ and $W_U$ .	1.0 and 2.5
<b>4</b>	$R_{WHL}$ and $R_{WHU}$	Parameter mentioned in Chapter 3.5.3. They are the ratio between the width and the height of the detected vehicle.	0.5 and 4.1
<b>5</b>	$e$	Parameter mentioned in Chapter 3.5.3. It is the range of x-coordinate for evaluating vertical projection near the end point of the detected horizontal line.	10
<b>6</b>	$V_h$	Parameter mentioned in Chapter 3.5.3. It is the threshold for confirming the existence of strong vertical edge from the vehicle.	40
<b>7</b>	$e_x$ and $e_y$	Parameter mentioned in Chapter 3.5.4. The number of pixels to expand the bounding rectangle in the x- and y-coordinate of the screen respectively for tracking of detected vehicle in successive frames	3 and 5



Figure 4-8: Detection of different vehicles on the road.

### 4.3.3 Detection Rate

The detection and tracking algorithms were evaluated with self-prepared video sequences. These video sequences are listed in Table 4-6. These sequences were prepared with the simplest scenario and shortest duration as in Sequence A, and to more complex scenarios and longer duration as in Sequence F and G. The challenges contained in these sequences are illustrated in Figure 4-9. They include shadows from the environment, broken road with non-uniform colour, text or symbol on the road, fences on the road side, increasing or decreasing of distance to the front vehicle, and lane change due to the front vehicle or the ego vehicle.

Table 4-6: Video sequences with different challenges to the proposed algorithm

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

The image sequences were then encoded by the JM18.4 H.264/AVC encoder. The block size of MVs in a frame was configured so that it varied from 8x8 to 16x16. Those MVs of block size larger than 8x8 were regarded as multiple blocks of size 8x8 with the same MV value, as proposed in Chapter 3.1.1.





Figure 4-9: 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.

Different Quantization Parameters (QP) of value 9, 17, 28, 35 and 45 were chosen to test the sequences from low to high compression with target bit-rate ranging from 8.0Mb/s down to 1.5Mb/s. The smaller is the QP, the higher is the video quality. The MVs from the encoder with different QP have similar patterns. With higher QP for higher compression, larger block size MVs and more SKIP mode blocks are used. This can be observed from Figure 4-10(i) versus Figure 4-10(a). The number of macroblocks using 16x8 and smaller partitions is higher for lower QP, but the resulting ROI for moving object detection was similar, as illustrated in the right side of Figure 4-10. The resulting video quality using QP35 and QP45 are nearly un-usable, but the MV amplitudes around the relatively slow speed vehicles at the front are small, not being affected by the selected QP value.

One point to note is that the detection algorithm is run on a system with an H.264/AVC encoder, the source image is available for being used by the algorithm. Since the source image is used for the identification of road region and the U-shape feature of vehicles, the degraded video quality due to compression will not affect the detection result.

For the detection result, it was categorised into true-positive, false-positive and un-successful detection. True positive detection is achieved by either successful detection or tracking of vehicles. The detection results are summarised in Table 4-7. The last row of Table 4-7 combines the result of Sequence A to G by summing all the frames in the sequences.

Table 4-8 shows the combined detection rate of all these sequences. The results show that the true-positive detection can reach more than 90%, while the false-positive rate remains at very low level of less than 1%. Figure 4-12 to Figure 4-18 show a snapshot of the image sequences.

It is noted that the detection rate remained relatively stable with all QP values used. There is no significant gain in detection rate with the increase in video quality. This is because the source image was used for object detection. Therefore, the detection algorithm was not affected by the degradation in picture quality due to video compression. Since the source image was the same, the difference in detection rate in the same sequence with different QP was because of erroneous MVs on the vehicles at the front. Some region of the vehicle was masked during the ROI evaluation as illustrated in Figure 4-11. This led to corrupted U-shape features, affecting the detection algorithm.

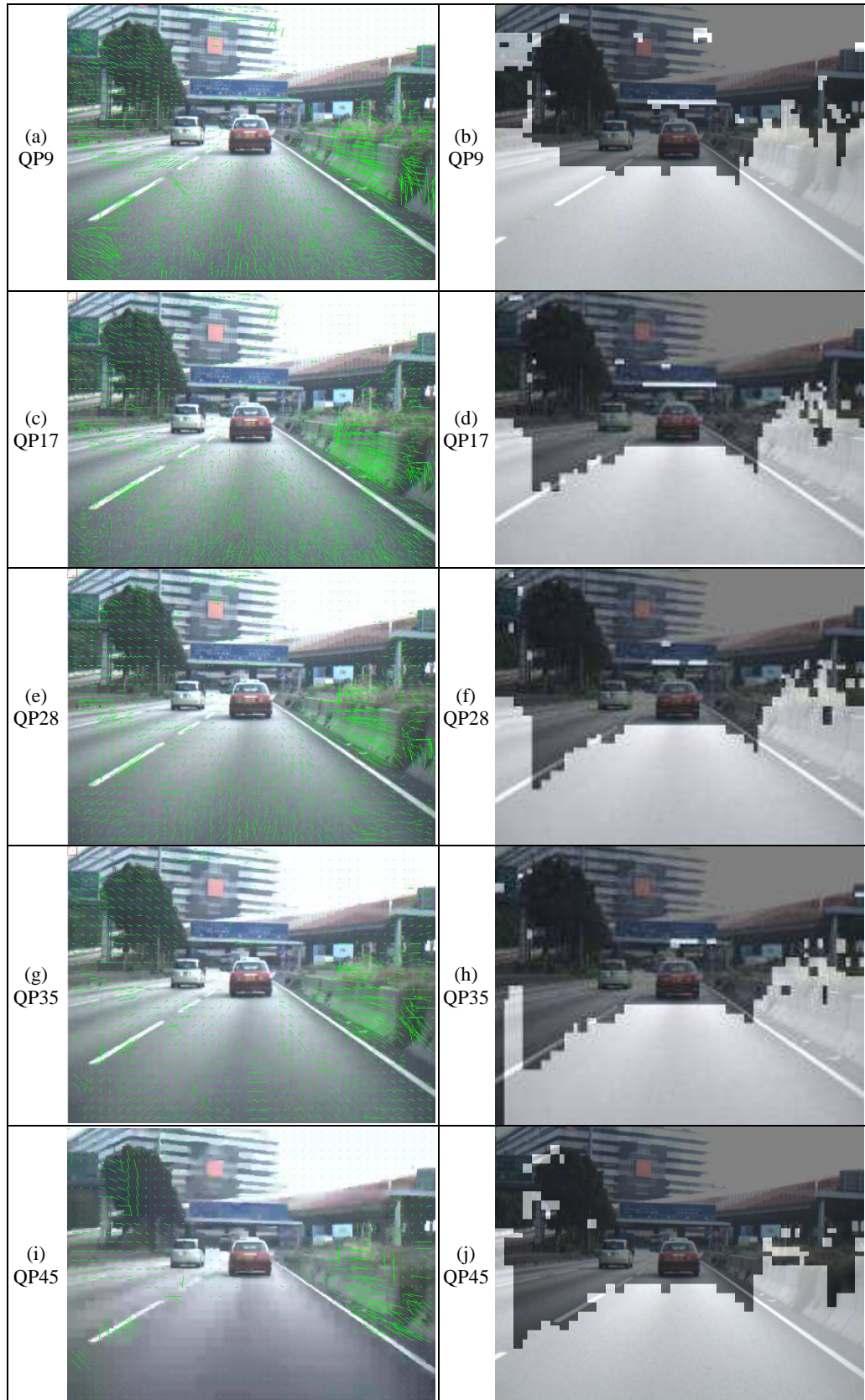


Figure 4-10: The left side shows MVs of macroblocks of a typical frame with different QP. (a) QP=9, (c) QP=17, (e) QP=28, (g) QP=35, (i) QP=45. The right side shows the corresponding ROI with different QP constructed by using the amplitudes of MVs and the identified road region. It is observed that more coarse macroblocks (16x16) were used with higher QP, but the resulting ROI was essentially the same, preserving the regions with relatively slow speed vehicles at the front.



Table 4-7: Detection result of seven image sequences. The last row shows the combined result of sequence A to G.

Sequence	Total frame	True-positive					False-positive					Unsuccessful				
		QP 9	QP 17	QP 28	QP 35	QP 45	QP 9	QP 17	QP 28	QP 35	QP 45	QP 9	QP 17	QP 28	QP 35	QP 45
A	42	42	42	42	42	42	0	0	0	0	0	0	0	0	0	0
B	100	97	97	97	97	97	0	0	0	0	0	3	3	3	3	3
C	246	239	239	239	237	237	1	1	1	1	1	6	6	6	8	8
D	275	262	262	261	262	262	0	0	0	0	0	13	13	14	13	13
E	440	425	425	399	399	399	1	1	1	1	1	14	14	40	40	40
F	208	188	188	189	188	188	0	0	0	0	0	20	20	19	20	20
G	342	331	331	332	331	331	0	0	0	0	0	11	11	10	11	11
A to G	1653	1584	1584	1559	1556	1556	2	2	2	2	2	67	67	92	95	95

Table 4-8: Detection rate of the seven image sequences. The last row shows the detection rate of the sequence combined from A to G

Sequence	Total frame	Detection rate %				
		QP9	QP 17	QP 28	QP35	QP45
A	42	100	100	100	100	100
B	100	97.0	97.0	97.0	97.0	97.0
C	246	97.2	97.2	97.2	96.3	96.3
D	275	95.3	95.3	94.9	95.3	95.3
E	440	96.6	96.6	90.7	90.7	90.7
F	208	90.4	90.4	90.9	90.4	90.4
G	342	96.8	96.8	97.1	96.8	96.8
A to G	1653	95.8	95.8	94.3	94.1	94.1

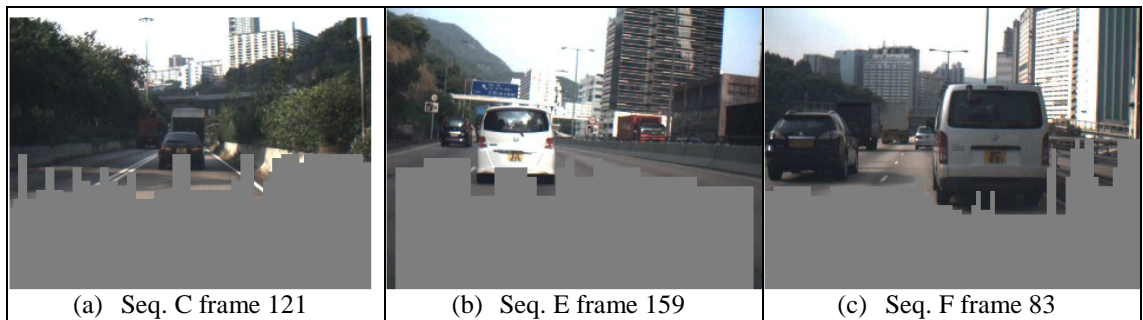


Figure 4-11: Illustration of images with un-successful detection. These images show the region of interest in grey which was constructed by combining the detected road region and the region with MV amplitude larger than a threshold. All images show significant masking of the U-shape feature of these vehicles, leading to unsuccessful detection.

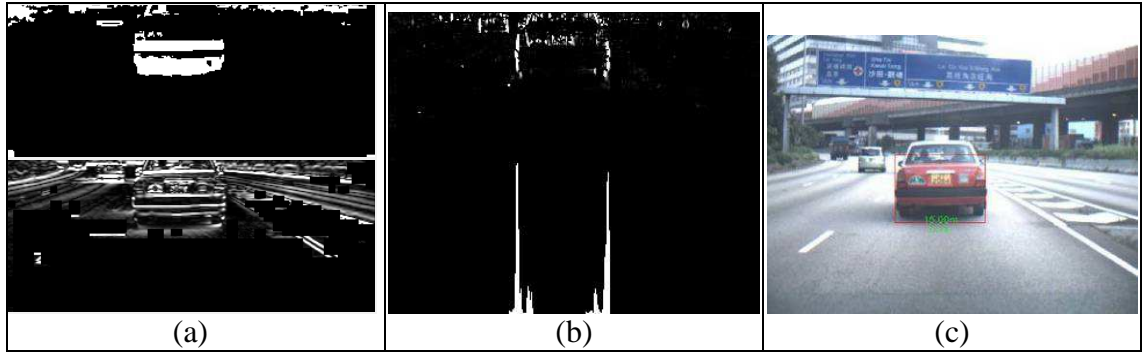


Figure 4-12: A snapshot of sequence A at frame 84. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the overlaid rectangle representing the identified position of the detected vehicle.

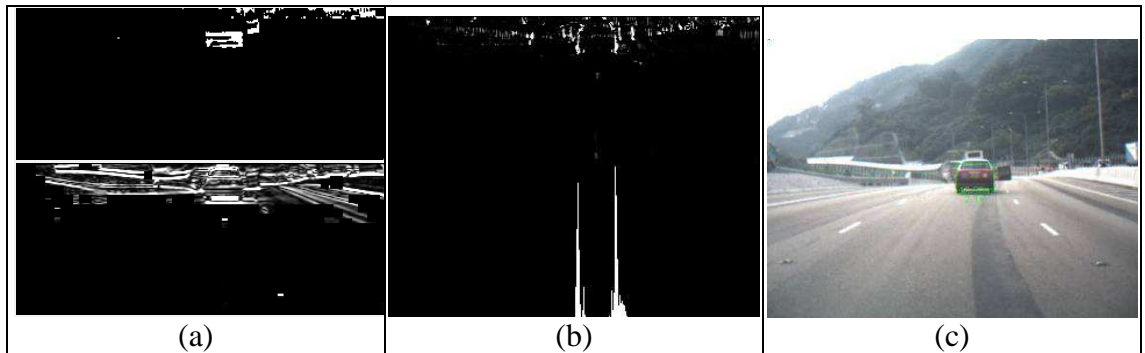


Figure 4-13: A snapshot of sequence B at frame 4. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the overlaid rectangle representing the identified position of the detected vehicle.

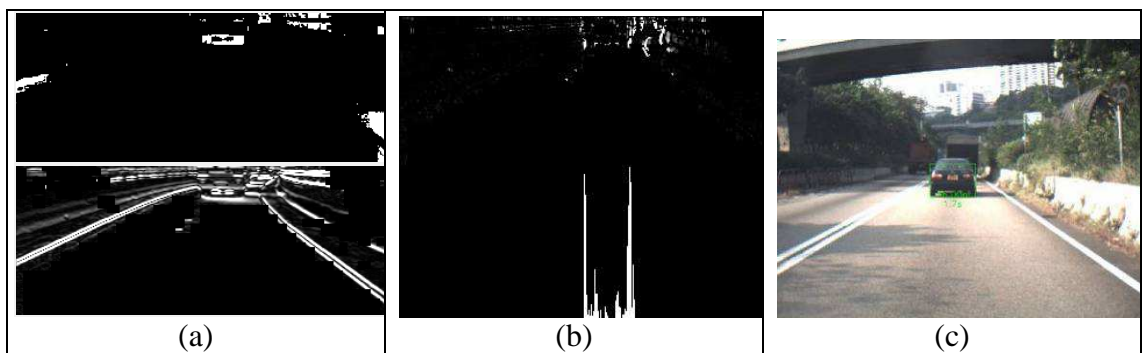


Figure 4-14: A snapshot of sequence C at frame 92. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle.

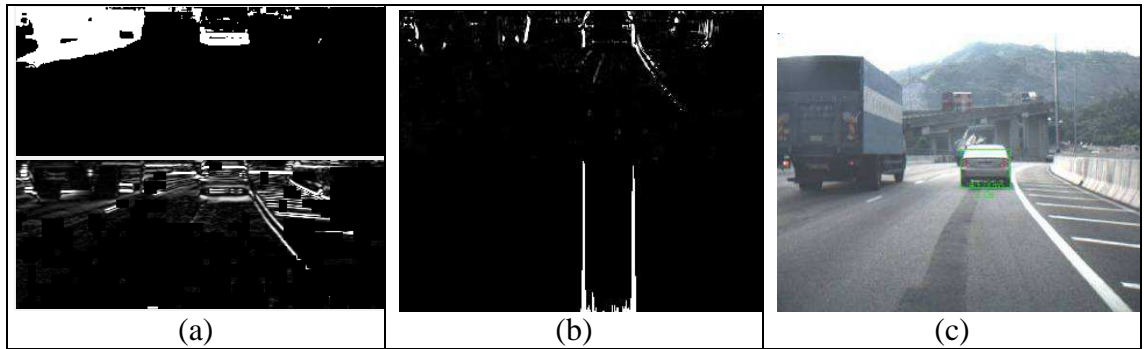


Figure 4-15: A snapshot of sequence D at frame 6. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle.

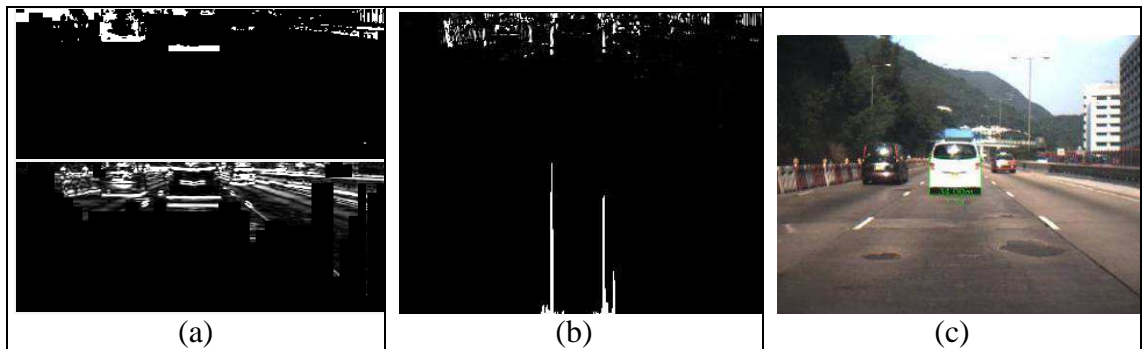


Figure 4-16: A snapshot of sequence E at frame 790. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle.

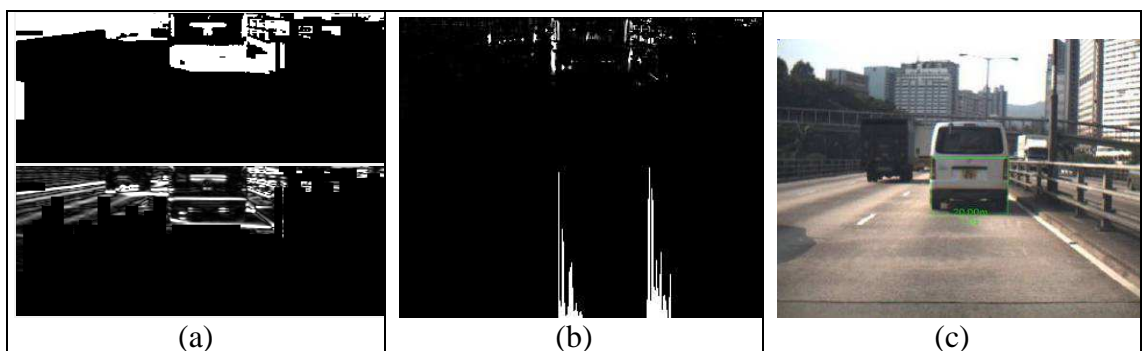


Figure 4-17: A snapshot of sequence F at frame 228. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle.

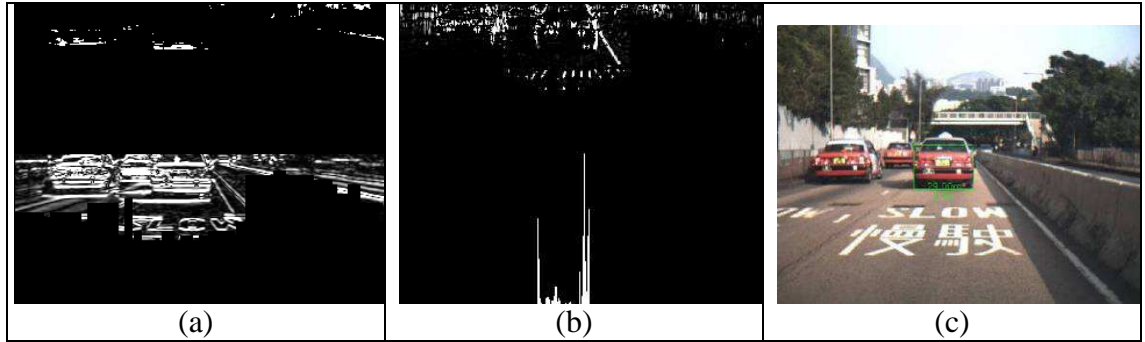


Figure 4-18: A snapshot of sequence G at frame 742. (a) Binary threshold and the vertical gradient images. (b) Filtered horizontal gradient image and its corresponding horizontal projection near the edges of the front vehicle. (c) Original image with the green rectangle representing the identified position of the detected vehicle.

#### 4.3.4 Computation Time Analysis

Since the proposed algorithm was targeted for real-time application in automobiles, the computation load of the algorithm is also analysed.

The algorithm was developed in C++ language and was tested with a PC with x86 processor running at 2.6GHz clock speed. Captured image sequences of size 640x480 each were stored to the PC for offline processing. The JM18.4 H.264/AVC video encoder (JVT, 2012) was used for offline encoding the video to H.264 format. The video encoder was modified to output MV map for each P-frame and was stored in the PC. Furthermore, the encoder was set to IBPBP frame structure, with frame rate of 30fps, using EPZS motion estimation algorithm, with intra-frame encoding in P-frame disabled and macroblock partition smaller than 8x8 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 low relative 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 test sequences mentioned in Table 4-6 is shown in Table 4-9. It was found that the processing time for finding the ROI and for vehicle tracking was relatively low with a maximum of 21.5ms. However, the average processing time for relatively slow vehicle detection varied from 15.7ms to 192.9ms. The large deviation in the detection time was due to the existence of multiple regions with U-shape features. For instance, Figure 4-19(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. Figure 4-19(b) highlights the areas in red circles that are required 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 was 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 is less than 66.7ms, the detection cycle is fast enough to catch up with the designed video frame rate.

Ignoring the file input/output (I/O) time (IV in Table 4-9) 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 4-9: Average processing time in ms for low relative speed vehicle detection. The time for finding ROI and tracking is relatively stable. The detection time varies due to the difference in area for potential vehicle detection.

Sequence	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



Figure 4-19: (a) Binary image with white colour representing the area that is darker than the minimum grey-level of the identified road region. (b) Multiple regions for potential vehicle detection, circled in red colour.

It was noticed that the cycle time for vehicle detection (I+II) can vary from tens of milliseconds to hundreds of milliseconds. This is why the tracking algorithm was required to shorten the computational time upon successful detection of a vehicle so that more computation resources can be reserved for the need of other processes. Nevertheless, if the detection takes hundreds of milliseconds to complete, the detection algorithm is still effective for a real-time application. This is because the expected movement of low relative speed vehicles 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.

The computation time increases with increase in the number of U-shape features that indicates the number of potential vehicles in the image. Since the algorithm tries to find U-shapes by scanning the ROI in the image, the increase in the image resolution and the number of pixels in the ROI will also increase the computation time for vehicle detection. The worst timing in those sequences were taken as the worst case figures for estimating the detection time and ROI evaluation time with increase in number of vehicle and image resolution. There were up to four U-shapes in the entire ROI in Sequence F requiring 192.9ms for detection. So each U-shape requires 48.2ms to process. The time taken for ROI evaluation was 17.2ms in Sequence G, and the time for tracking was 9.9ms in Sequence D. The experiment was carried out with an image

resolution of 640x480, the corresponding processing time at higher resolutions was projected from this resolution according to the percentage change in resolution. The time required for processing the image at different resolutions for one vehicle is shown in Table 4-10. Considering that there are multiple vehicles to detect in the same image and only one vehicle is tracked after the detection, a plot of processing time versus the detection cycle time is shown in Figure 4-20. It shows that the increase in image resolution can impact the detection cycle time significantly. For instance, the processing cycle time for only 1 vehicle in the image at 1920x1080 can exceed 500ms. Further increase in the number of vehicle in the image will make the detection time being unpractical for real-time application. However, the algorithm is able to maintain the cycle time at less than 500ms for up to 9 vehicles, which is usable for real-time applications. Nevertheless, the algorithm computation time can be improved by employing methods proposed in Chapter 7.

Table 4-10: Processing time of the algorithm at different resolutions. The processing time for 1280x720 and 1920x1080 was projected from the result at 640x480.

	@ 640x480	@ 1280x720 (x3)	@ 1920x1080 (x6.75)
<b>Time to find ROI</b>	<b>17.2</b>	<b>51.6</b>	<b>110.9</b>
<b>Detection Time per vehicle</b>	<b>48.2</b>	<b>144.6</b>	<b>325.4</b>
<b>Tracking per vehicle</b>	<b>9.9</b>	<b>29.7</b>	<b>66.8</b>

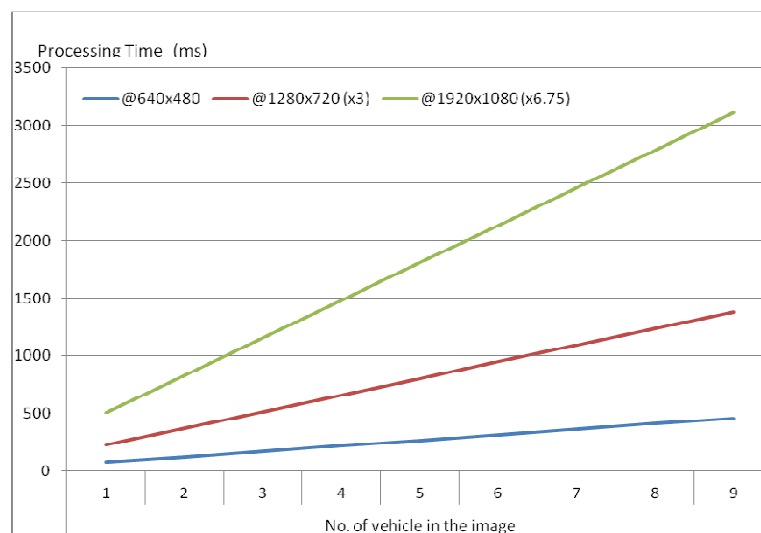


Figure 4-20: Processing time for multiple vehicles in an image at different image resolution.

### 4.3.5 Comparison of Results

Since the test results from other publications were making use of different test sequences for evaluations, direct comparison to results from this project was not possible. The algorithm proposed in this report requires to use the information from the inertial sensors and the vehicle speed sensor, using the publicly available database for evaluation directly was also not possible.

However, the comparison can still give a reference for the performance of the proposed algorithm. Monocular vision based algorithms using feature based methods (Wang and Lien, 2008, Jazayeri and Hongyuan et al., 2011) and statistical based methods (Sun and Bebis et al., 2006, Chang and Cho, 2010, Sivaraman and Trivedi, 2010, Yuan and Thangali et al., 2011, Chen and Chen et al., 2013, Wen and Shao et al., 2015, Cheon and Lee et al., 2012) that were published in recent years were selected for comparison.

Table 4-11 shows the comparison among the selected algorithms for the detection rate and false positive rate. The detection rate of the algorithm proposed in this study shown in Table 4-11 is the combined detection rate on test sequence A to G mentioned in the last row of Table 4-7.

Within the comparison, the detection rate and false positive rate of the proposed algorithm is on a par with the detection rate of other algorithms in the comparison. One of the reasons for the high detection rate and low false positive rate is the elimination of non-vehicle objects by limiting the ROI to the regions with small MV. Another reason is the good image threshold value obtained during the road region detection stage which was used for constructing a binary image with essentially the dark contours of vehicles and their respective shadows.



Table 4-11: Comparison on the detection rate and false-positive rate of the proposed algorithm vs. other selected algorithms from renowned journals.

Research Study	Description	Detection Rate	False Positive Rate	Remarks
Sun et al. (2006)	Statistical based vehicle detection using HOG and Gabor features, followed by SVM and neural network classification	96.1%	2.29%	
Wang and Lien (2008)	Feature based vehicle detection using local features of vehicles, extracted by principal component analysis (PCA) and independent component analysis (ICA) for hypothesis generation. Hypothesis verification is done by a posterior probability function.	95.6%	<0.5%	Evaluation is done on selected static images. 8.7fps was achieved.
Chang and Cho (2010)	Statistical based detection using Haar-like feature and Adaboost classification. It also features on-line continuous learning to refine the trained classifier	96%	8%	
Sivaraman and Trivedi (2010)	(Similar to Chang et al.). Statistical based detection using Haar-like feature and Adaboost classification. It also features on-line continuous learning to refine the trained classifier	95%	6.4%	
Yuan et al. (2011)	Statistical based using HOG features and SVM classification.	82%	1 per frame	
Jazayeri et al. (2011)	Motion based using optical flow, followed by feature based hidden Markov model classification	86.6%	13.2%	
Cheon et al. (2012)	Statistical based using HOG symmetry features and a classifier based on total error rate minimisation using reduced model.	93%	5%	
Chen et al. (2013)	Road modelling followed by Haar-like feature and eigencolour based detection using Adaboost classifier	94.32%	5.52%	
Wen et al. (2015)	Haar like feature based followed by SVM	94.1%	3.26%	
Proposed Algorithm	Vehicle detection by road region estimation, MV amplitude for ROI selection, horizontal contours, horizontal projection and vertical projection.	95.8%	<0.5%	True-positive detection rate of all the test sequences A to G using QP=17 for video coding.

## **4.4 Fast Relative Speed Vehicle Detection and Tracking Method**

The effectiveness of the proposed fast relative speed vehicle detection algorithm and its computational speeds are summarised in this chapter.

### **4.4.1 Region of Interest Formation**

The region of interest (ROI) for relatively fast moving object detection is the area that is outside the detected road region and having MV amplitude larger than the threshold  $q_m$ . Figure 4-21 shows the sequence of images in the formation of the ROI for relative fast speed vehicle detection. Figure 4-21 (a) is the original captured image. The road region in front of the ego vehicle is identified and shown in white blocks in Figure 4-21(b). The white blocks in Figure 4-21(c) represent those with MV amplitude larger than a threshold  $q_m$ .  $q_m$  is set to 12 according to the experiment results. Figure 4-21(d) shows the result of combine the road region mask in Figure 4-21(b) and the MV region mask in Figure 4-21(c). It is noticed that there are many small black colour blocks surrounded by the white mask. Unlike the ROI for slow relative speed object detection in which these black blocks were removed by a hole filling algorithm mentioned in Chapter 3.4.1, these small black blocks were not removed from the image mask so that more blocks were retained. This increased the number of blocks with larger amplitude for better relatively fast speed moving object detection. Figure 4-21(e) shows the resultant ROI for fast relative speed moving object detection overlaid with the original captured image. Figure 4-21(f) is the ROI of the lower half of the image overlaid with the original captured image. It is the image mask used for relatively fast moving objection detection. As clearly seen from Figure 4-21(f), the ROI contains essentially the vehicle moving from the left to the right of the screen. There were some outliers due to the motion estimation error of the H.264/AVC encoder. Some blocks on the

moving car body were also masked. This is because of the motion estimation error of the H.264/AVC encoder to blocks with weak texture or repetitive pattern.

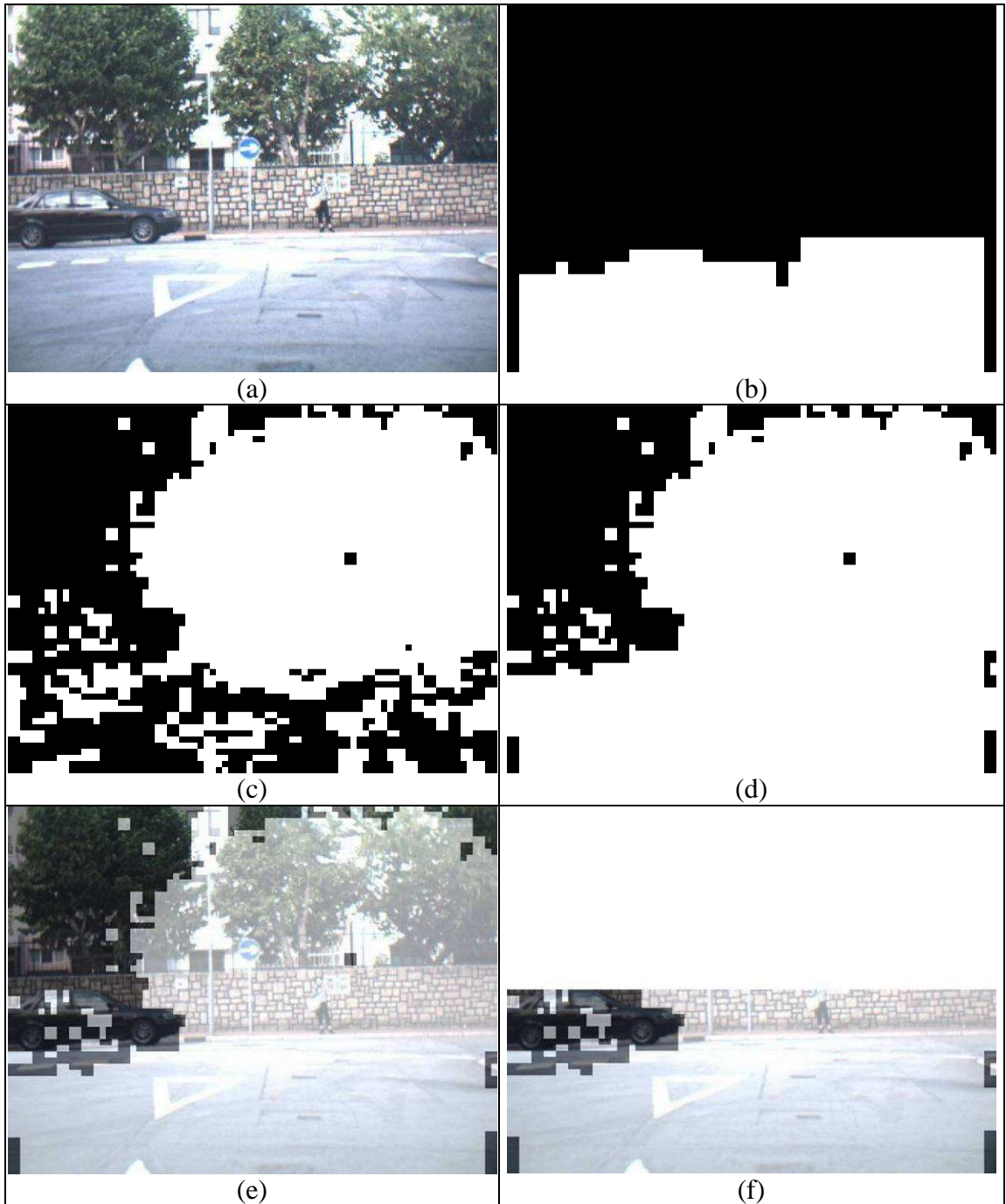


Figure 4-21: Illustration of road region detection and exclusion of areas with small MV amplitude. (a) Original captured image. (b) Result of block based road region detection. White colour represents the detected road region. (c) Blocks with MV amplitude smaller than a threshold filling with white colour. (d) Combined mask from road region and small MV areas. (e) Combined region mask. (f) Final region of interest, with only the areas in the lower half of the image that are not filled with white. The overlaid image shows that most of the MVs on the relatively fast moving object are not masked.

It was observed that the ROI using MV amplitude alone as shown in Figure 4-21(c) contains many disconnected blocks in the areas corresponding to the road region. After combining with the detected road region, the ROI could be reduced to minimise the computational time for moving object detection, as well as to reduce the false detection rate.

#### **4.4.2 Setup of Experiments**

For normal driving on the road, most vehicles are moving at similar speed to the ego vehicle. However, there are occasions when some vehicles are moving relatively fast and are not necessarily moving at the same direction as the ego vehicle. Table 4-12 and Figure 4-23 show the description and a snapshot respectively of eight video sequences for the evaluation of the fast relative speed moving object detection algorithm. These video sequences were created with a test vehicle moving across junctions, or driving at the front of the ego vehicle with sudden lane changes. In addition, two sequences V and W were created with an air inflatable dummy vehicle as a fast relative speed moving object and with the ego vehicle having head-on collision to it, to investigate the effectiveness of the algorithm.

The inflatable dummy vehicle is shown in Figure 4-22. The size of the dummy vehicle was similar to a standard compact private car. Since the mass of the dummy vehicle was small, typically less than 5kg, no damage was introduced to the ego vehicle during collisions. Also, there was no motorised component in the dummy car, it was required to be pulled by a human across the road during the test.

Table 4-12: Video sequences with different challenges to the proposed algorithm

Sequence	Description
Seq. P	Two vehicles move from the left to the right in the screen. The ego vehicle is moving with distance of around 10-20m from these vehicles
Seq. Q	One vehicle moves from the left to the right in the screen. The ego vehicle is moving with distance of around 10-20m from the vehicle.
Seq. R	One vehicle moves from the right to the left in the screen. The ego vehicle is moving with distance of around 20-30m from the vehicle.
Seq. S	One vehicle is moving from opposite lane and is crossing the road from the right to the left in the screen. The ego vehicle is moving with distance of around 30-45m from the vehicle.
Seq. T	One vehicle is moving on the left side of the ego vehicle, then changing lane from left to right. It then changes its lane again from the right to the left. The ego vehicle is moving with distance of around 10-20m from the vehicle.
Seq. U	One vehicle is moving at the front of the ego vehicle, then changing lane from the centre to the right and then back to the centre lane again. The ego vehicle is moving with distance of around 10-20m from the vehicle.
Seq. V	One dummy vehicle is moving from the left to the right, the ego vehicle is having direct collision with the dummy vehicle. The ego vehicle is accelerating hardly from standstill, whereas the motion of the dummy vehicle is pulled by human force with moderate acceleration only.
Seq. W	One dummy vehicle is moving from the right to the left, the ego vehicle is having direct collision with the dummy vehicle. The ego vehicle is accelerating hardly from standstill, whereas the motion of the dummy vehicle is pulled by human force with moderate acceleration only.

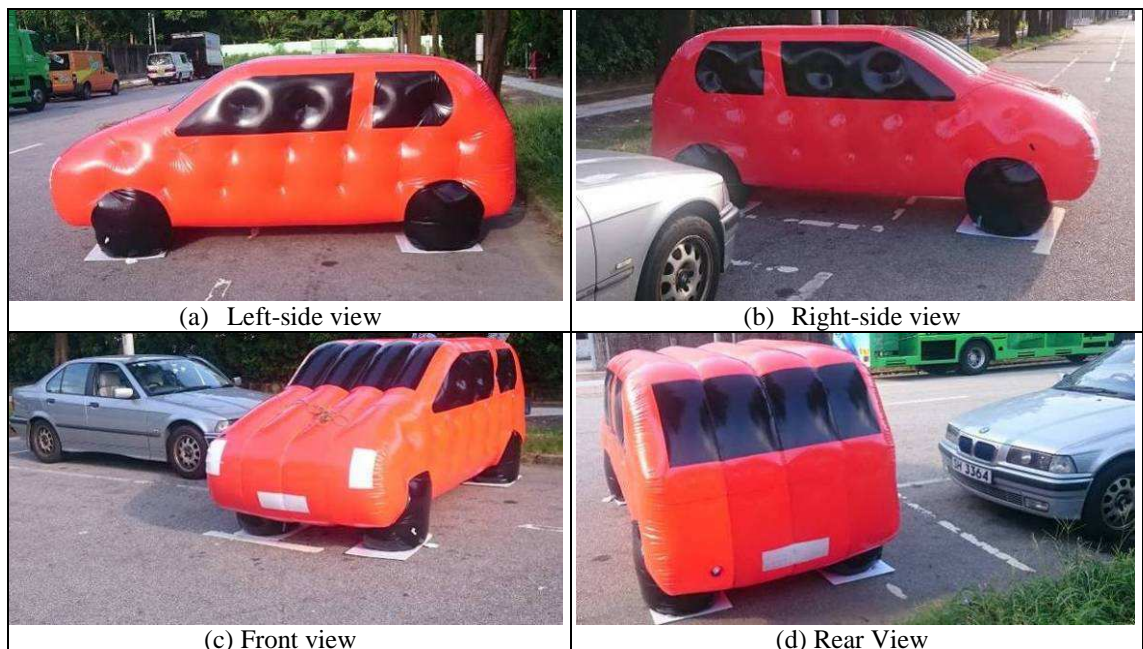


Figure 4-22: The air inflatable dummy car used for testing in this project. Its size is similar to a compact private car as shown in the pictures.

One point to note is that the detection of fast moving objects is targeted to identify part of the object for generating a warning alert. It is not possible to detect the whole moving object in real-time. This is because of the erroneous MVs from the H.264/AVC encoder mentioned in Chapter 2.4. Also, the different MV amplitude and direction on the object due to perspective transformation would lead to the same object being clustered into different objects for verification. Another problem is the additional computational resources required to interpret the image and to cluster correctly with real-time performance. Nevertheless, the driver only needs to know that an object may pose danger to his driving direction. Therefore, the target of successful detection is to detect only part of the moving object so that the driver can be alerted of potential danger during driving.

Since only moving objects that may collide with the ego vehicle need to be detected for a warning output to be given to the driver, those ego motion compensated MVs that represent objects moving away from the ego vehicle were discarded. The detection criteria were the same as that proposed in Chapter 3.6.2, which include the amplitude, position and direction constraints.





Figure 4-23: Eight video sequences to evaluate the proposed fast relative speed moving object detection.

### 4.4.3 Detection Results

The parameters for relative fast speed moving object detection are listed in Table 4-13.  $q_m$  is the threshold for segmenting the region of interest according to the amplitude of MVs. It was determined with the assumption that relatively fast speed moving objects have relatively large MV amplitudes. It was set to 12, which is 1.5 times of the block size used. A larger value of  $q_m$  will mean increasing the region of interest for relatively slow speed moving object detection, and vice versa.  $Y_u$  is the parameter to determine the y-position in the image where the alert zone starts. Since the origin starts from the top left corner, a smaller  $Y_u$  means a larger alert zone, accepting more PPRVs for processing.  $Y_u$  was set to 350 in the experiments conducted. This represents around 3.5m from the ego vehicle, which is approximately twice the distance between the camera and the front nose of the test vehicle.  $T_y$  is the time to collision. It was set to 2 seconds to provide enough of a time buffer between the ego-vehicle and the moving object. A larger  $T_y$  will also mean accepting more PPRVs for processing.  $x_a$  is the x-position of the PPRV after entering the alert zone.  $x_a$  was set to 650, a larger value than the maximum range of the x-coordinate of 639, in order to retain more PPRVs for processing.  $m_{thres}$  is the parameter to determine the gradient difference between the point to the FOE and the corresponding PPRV at the point concerned.  $m_{thres}$  was set to 30 degrees. A larger value of this parameter means that less PPRVs will be retained for processing. The setting of 30 degrees is a balance between the number of PPRVs to be included for processing, and the number of outliers belonging to static objects but with erroneous MVs.  $m_{diff}$  is the parameter for slope comparison in clustering. It was set to 15 deg in the experiments. A smaller threshold that used for filtering PPRVs can further narrow down the difference in MVs. But to account for the existence of erroneous and the limited precision of MVs, smaller value will result in many clusters



with only a few PPRVs.  $\lambda_{thres}$  is the parameter for distance comparison in clustering. It was set to 3, meaning that PPRVs will belong to different cluster if they are more than 3 blocks of size 8x8 apart. Setting a smaller value will result in more clusters being adjacent to each other.  $a_{diff}$  is the amplitude comparison for clustering. Of the amplitude of a PPRV is less than that of the average amplitude of the PPRVs in a cluster for comparison by  $a_{diff}$ , it will be included in the cluster. A larger value of this parameter will allows larger difference among PPRVs in a cluster, It was set to 15% to account for the precision and erroneous MVs from the encoder.

Table 4-13: Parameters for relative fast speed moving object detection

	<b>Parameter</b>	<b>Description</b>	<b>Value</b>
<b>1</b>	$q_m$	It is the threshold in number of pixels for segmentation of the image to regions for relatively slow and fast speed vehicle detection. It is the same parameter used in relative slow speed moving object detection.	12
<b>2</b>	$Y_u$	y-position in the image where the alert zone starts.	50
	$T_y$	$T_y$ is the time to collision associated with the PPRV entering the alert zone.	2 s
	$x_a$	$x_a$ is the x-position of the PPRV after entering the alert zone.	650
<b>3</b>	$m_{thres}$	Parameter to determine the gradient difference between the point to the FOE and the corresponding PPRV at the point concerned. $m_{thres}$ was set to 30 deg. A larger value means that less PPRVs will be retained for processing.	30 deg.
<b>4</b>	$m_{diff}$	Parameter for gradient comparison in clustering. .	15 deg.
<b>5</b>	$\lambda_{thres}$	Parameter for the distance between clusters.	3
<b>6</b>	$a_{diff}$	Threshold for amplitude comparison for clustering.	15%

Figure 4-24 to Figure 4-31 show the detection result for Seq. P to Seq. W. Figure 4-24(a) shows the detection result for Seq. P. The detected moving object indicated by

the coloured rectangle includes part of the moving vehicle from the left to the right of the screen. The overlaid rectangle shown in Figure 4-24(a) indicates a successful Hypothesis Generation (HG). The content inside the rectangle is used as the template for matching with the most similar pattern in the next frame for Hypothesis Verification (HV). Figure 4-24(b) shows a rectangle indicating a successful HV with the template defined in the previous frame. Detection of relatively fast moving object was confirmed when the template matching was successful in Figure 4-24(b). The template update was continued for matching in the next frame for tracking purpose, as shown in Figure 4-24 (c) and Figure 4-24 (d).

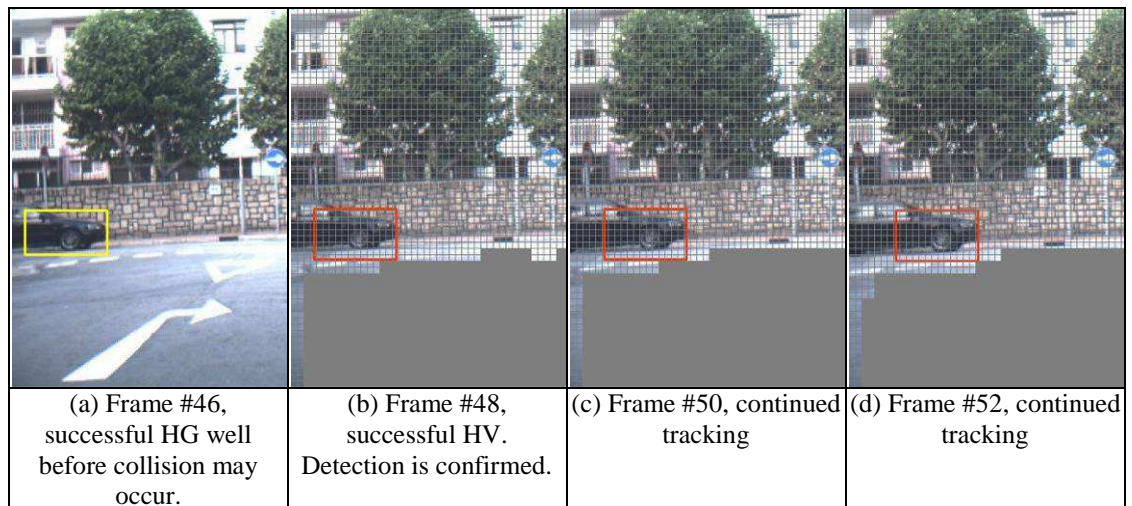


Figure 4-24: Detection Result for Seq. P.

Being similar to the detection result of Seq. P, the result for Seq. Q and Seq. R shown in Figure 4-25 and Figure 4-26 also reveals the successful detection of part of the relatively fast moving vehicle at the front.

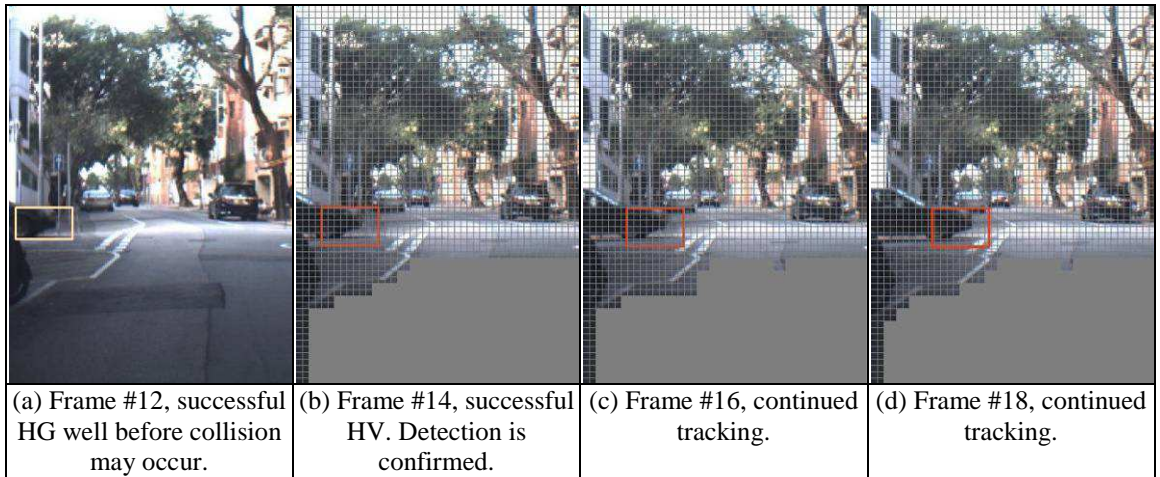


Figure 4-25: Detection Result for Seq. Q.

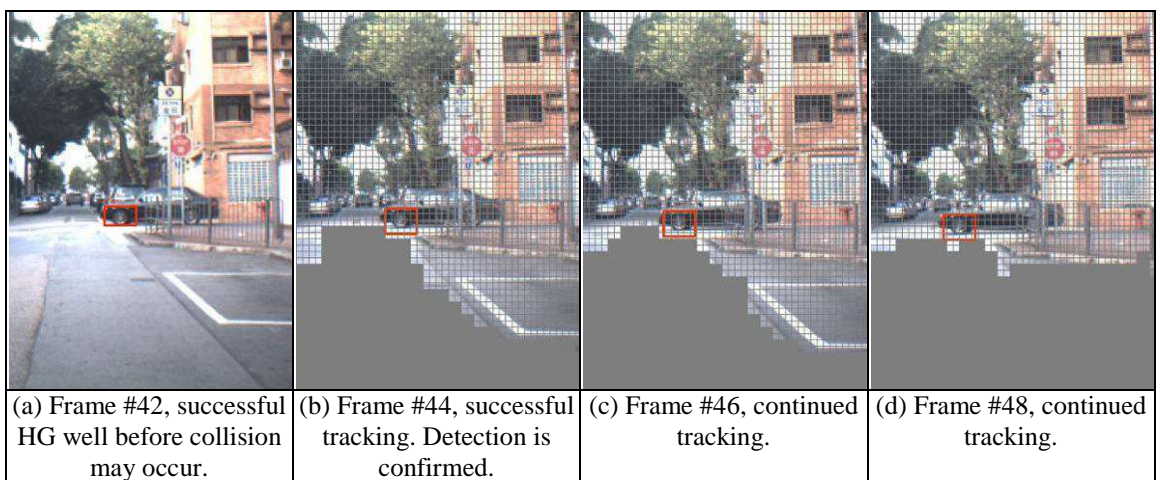


Figure 4-26: Detection Result for Seq. R.

Being different from Seq. P, Q and R with relatively fast moving object travelling from the left to the right or vice versa, Figure 4-27 shows the result for Seq. S that has a relatively fast moving object moving in opposite direction to the ego vehicle. It changed its direction and moved across the driving lane, causing danger to the ego vehicle. Figure 4-27(a) shows the successful HG using the constraints mentioned in Chapter 3.6.2. Figure 4-27(b) shows the successful HV of the relatively fast moving object after successful template matching. Figure 4-27(c) and (d) show the successful tracking for two more frames after successful HV.



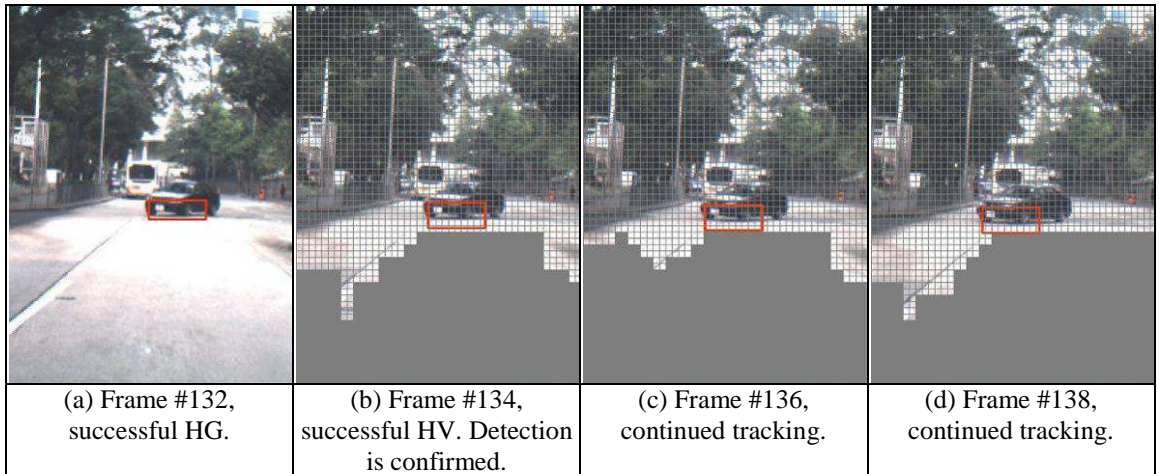


Figure 4-27: Detection Result for Seq. S.

Figure 4-28 and Figure 4-29 show the detection results for the case of a moving vehicle at the front that changes its driving lane suddenly. Both sequences show positive detection results. It is also noticed that the area of the part for template matching of the detected vehicle varies, depending on the result of initial detection based on the constraints to PPRVs.

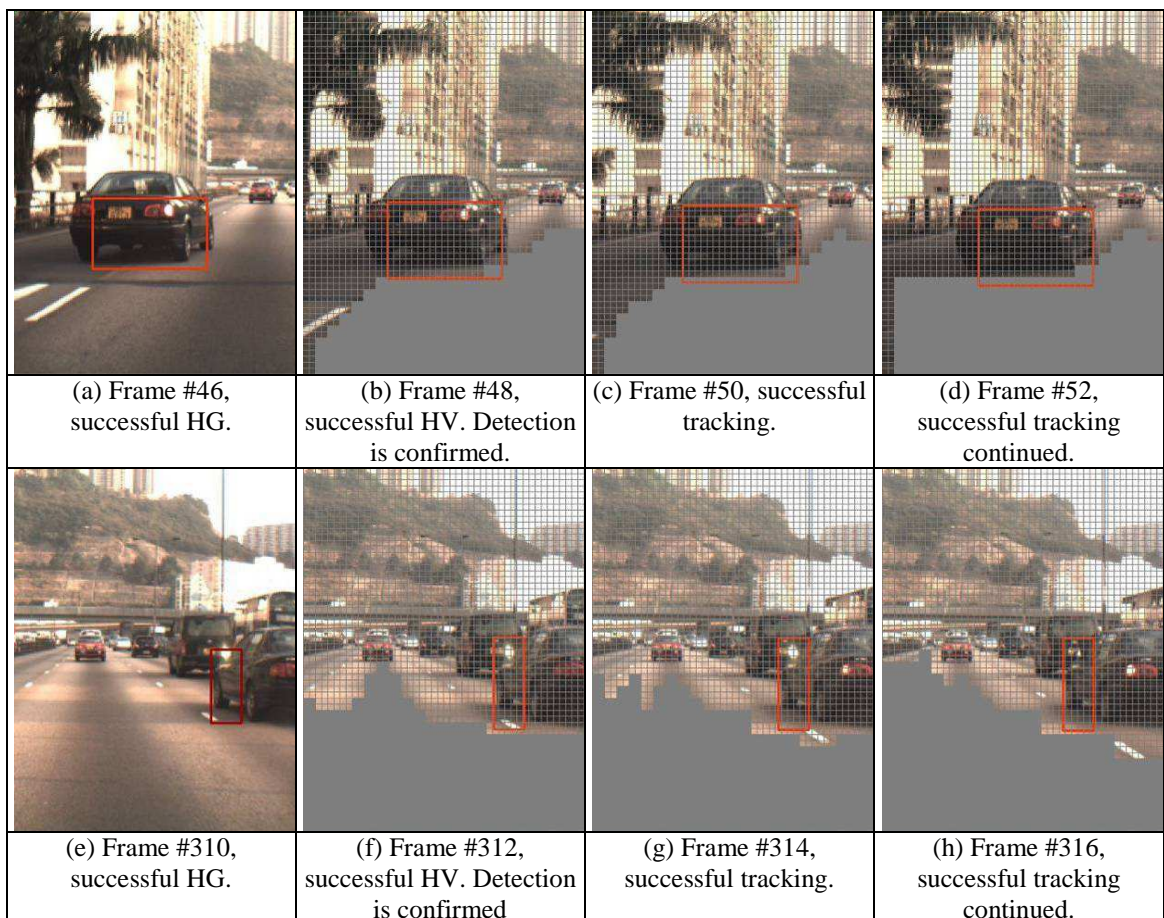


Figure 4-28: Detection Result for Seq. T.

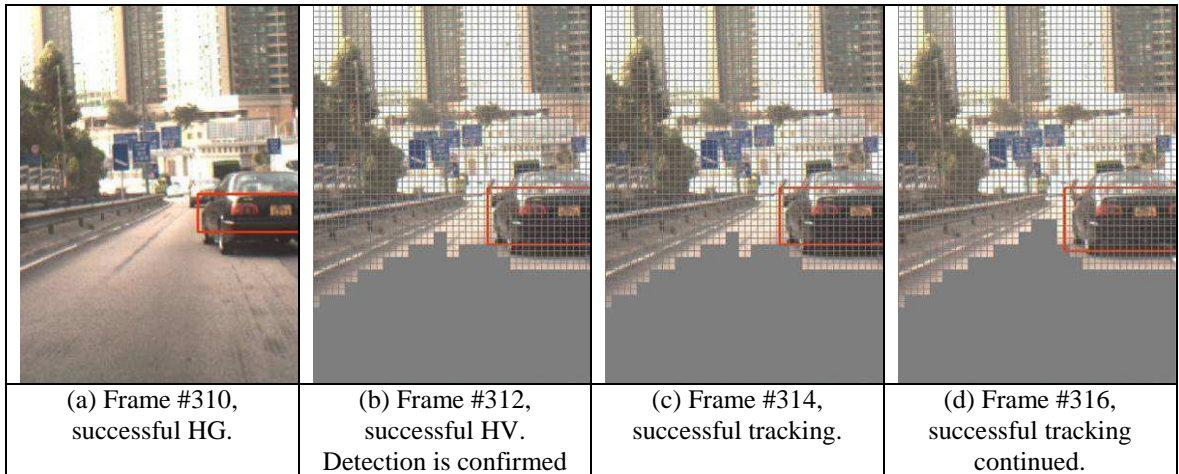


Figure 4-29: Detection Result for Seq. U.

The test condition for the scenarios shown in Figure 4-30 and Figure 4-31 was that the ego vehicle was accelerating quickly from stand still towards the moving dummy vehicle for a direct collision. The dummy vehicle was also stationary at the beginning. It was pulled manually by a running human so it accelerated gradually to around  $6\text{ms}^{-1}$ .

The result shown in Figure 4-30 indicates successful HG, HV and tracking, although the size of the template indicated by the coloured rectangles in Figure 4-30(a) to (c) has changed significantly. The size change is due to the intended expansion of the rectangle in the algorithm to accommodate potential increase in the size of the detected object due to camera perspective change and the distance of the object from the camera. The result shows that it can alert the driver before the collision may occur, leaving enough time for the driver to react.

For the result shown in Figure 4-31, the HG and HV were successful, meaning that the dummy vehicle was successfully identified. The dummy vehicle was also detected before the collision may occur, leaving enough time for the driver to react. However, the tracking was not successful after the HV stage. This was because both the dummy vehicle and the ego vehicle were accelerating. The predicted MV amplitude and direction have not been estimated with acceleration taking into account. This leads to a



relatively large deviation in the predicted displacement and the actual displacement, and hence the unsuccessful tracking.

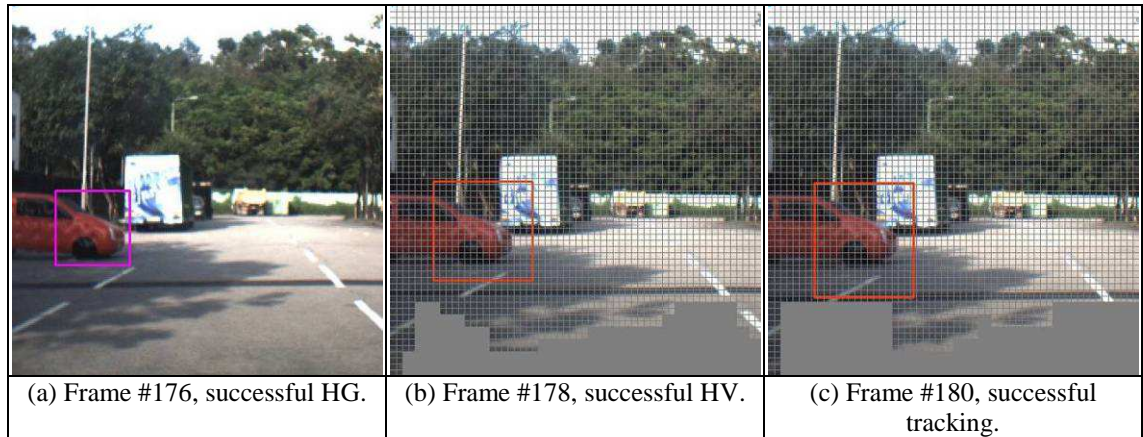


Figure 4-30: Detection Result for Seq. V

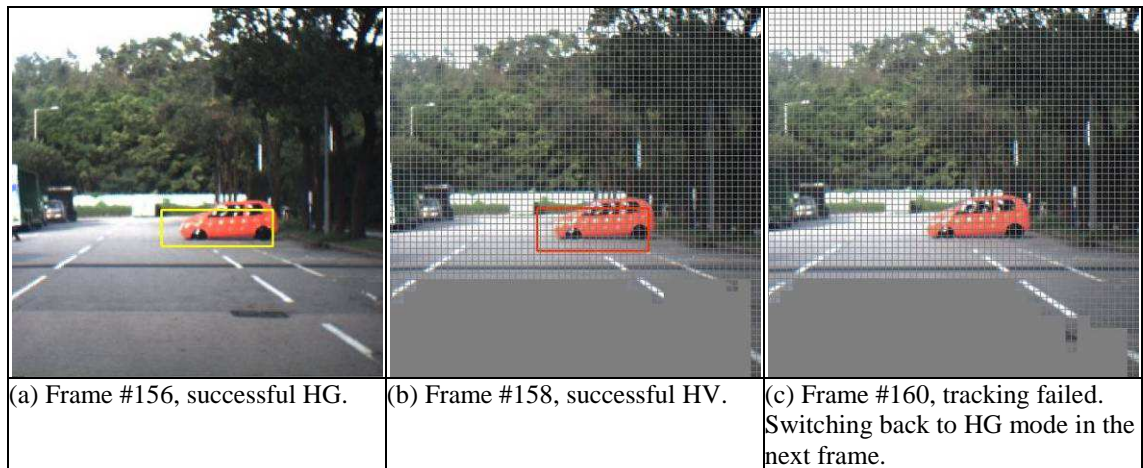


Figure 4-31: Detection Result for Seq. W. The tracking after HV shown in (c) is not successful.

#### 4.4.4 Exception of Detections

Despite the true-positive detections shown in Chapter 4.4.3, there were false-positive detections and unwanted loss-of-track after successful detection of a potential moving object.

##### Loss-of-track

Figure 4-31 shows the successful HG and HV result with failed tracking of the detected object in Figure 4-31(c). Since the ego and the dummy vehicles were moving under acceleration, the actual displacement of the object appearing on the screen deviated

beyond the allowable range of the predicted displacement. Therefore, the tracking failed. The detection process went back to HG stage to start the detection again in the next frame.

Figure 4-32 shows another lost-track case in which the initial clustered region included a large portion of static object in the scene. Figure 4-32(a) shows the undesirable clustering result in which a large area of the wall in front of the ego vehicle was included. The rich and repetitive texture of the wall resulted in many irregular MVs being evaluated by the H.264/AVC encoder. The clustering algorithm grouped the MVs wrongly due to unexpected similarities in amplitudes and directions. Subsequently, the area enclosed in the rectangle in Figure 4-32(a) was used as the template for block matching in Figure 4-32(b). Due to the repetitive pattern of the wall, the block matching algorithm returned a positive matching result as shown in Figure 4-32(b). When the vehicle moves further, the block matching algorithm returns negative result as the similarity falls below a pre-defined threshold. This ended the tracking mode and the HG mode was used in the successive frames.

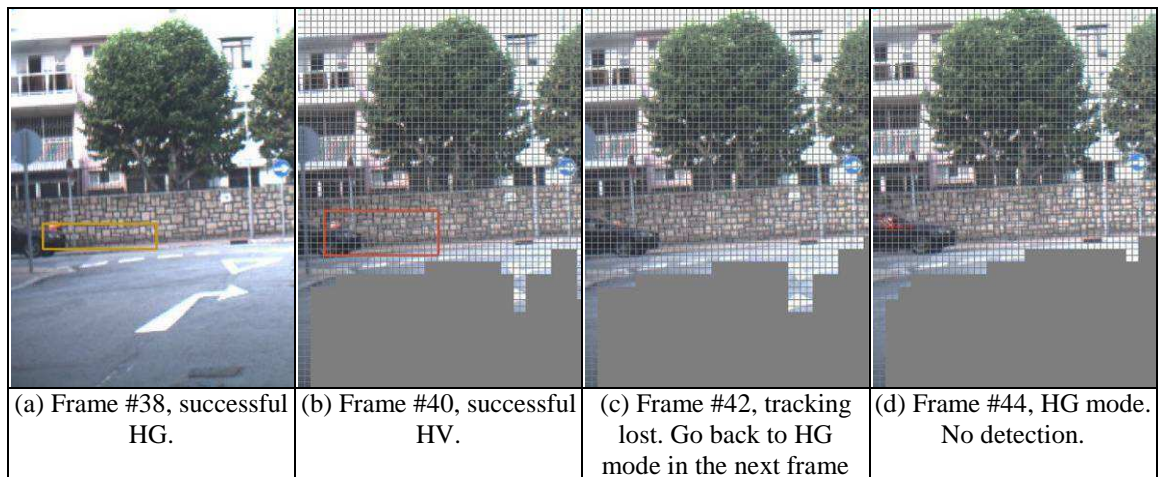


Figure 4-32: Loss-of-track after entering tracking mode.

### **False-Positive Detection**

There were cases of detecting non-moving objects on the road. Figure 4-33 (a) shows a false-positive detection result. Because of the rich texture and repetitive pattern of the wall, the H.264/AVC encoder outputs MVs violating the amplitude and direction of the estimated MVs for ego motion. The template found in the HG mode in Figure 4-33(a) was used for the HV and it was successfully verified in Figure 4-33(b). It was noticed that the resultant displacement of the matched template from Figure 4-33(a) to Figure 4-33(b) was around eight pixels horizontally and vertically, and the horizontal displacement was negative, meaning that the matched template was actually moving away from the ego vehicle. Since the displacement of the matched template has not been verified for its validity of being an object that may give rise to danger to the ego vehicle, it resulted in the false positive HG and HV. The false detection can further be eliminated if the matched template appears to be moving away from the ego vehicle.

Similarly, the false detection results shown in Figure 4-34 and Figure 4-35 were because of the erroneous MVs output from the H.264/AVC encoder. These MVs had a large deviation from the expected displacement due to ego motion. The clustering algorithm identified the region in Figure 4-34(a) and Figure 4-35(a) as potential moving objects. Because of the similarity of the region in successive frames, HV was successful with the use of the block matching algorithm where the actual displacement was within a predefined percentage from the expected displacement. The displacement of the template from Figure 4-34(a) to Figure 4-34(b), and Figure 4-35(a) to Figure 4-35(b) actually indicated that the object identified by HG was moving away from the ego vehicle. The false detection can further be eliminated if the displacement direction is taken into account.



Since the MV based moving object detection does not depend on the shape of the object, prior assumption on the size, shape and features of the object are not available. The wrongly detected object during HG cannot be rejected by methods related to the features of the detected object. Therefore, the HV stage is proposed to reject objects with inconsistent temporal motion. The false detection rate can be reduced further by re-designing the motion estimation algorithm of the H.264/AVC encoder using the ego motion information available from the built-in inertial sensor of the camera and the signal from the vehicle speed sensor.

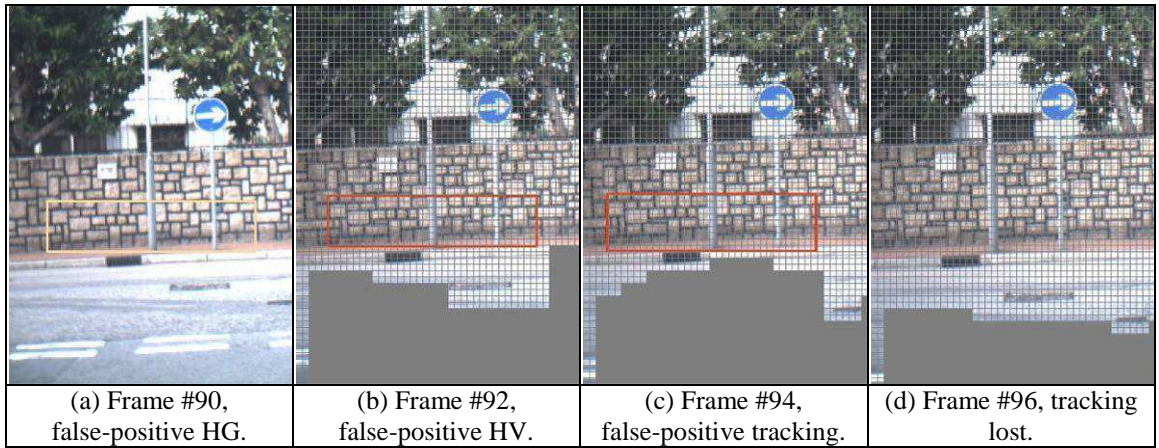


Figure 4-33: False-positive detection of the wall on the road.

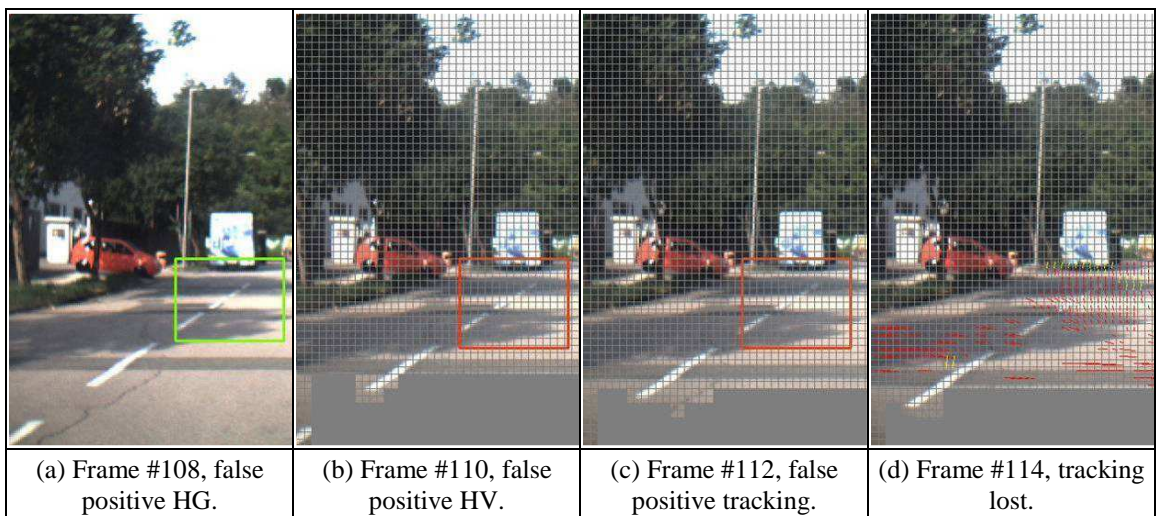


Figure 4-34: False-positive detection of the tree shadow on the road.

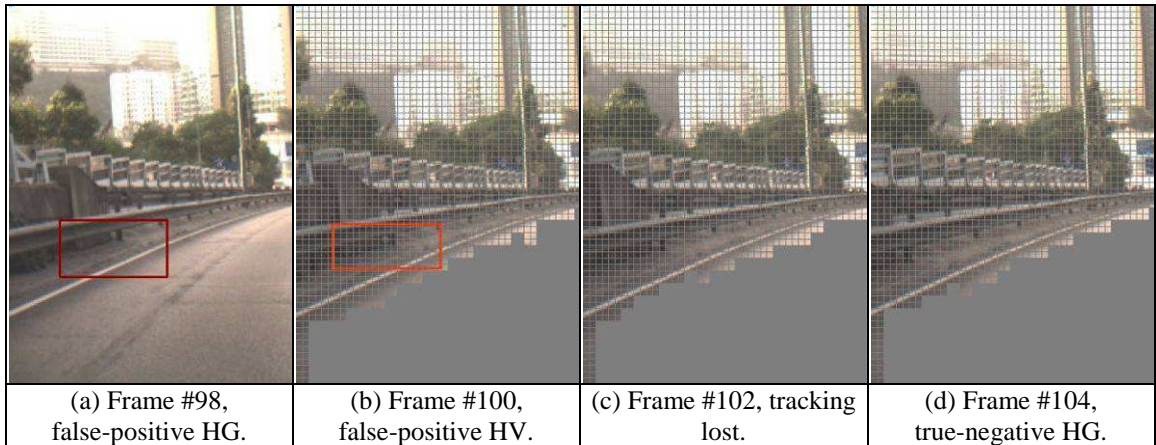


Figure 4-35: False-positive detection of the fence on the side of the road.

## 4.5 Computation Time Analysis

The computation load of the algorithm was analysed for its suitability for being used in real-time systems for practical application in automobiles.

The algorithm was developed in the same C++ language as that for the slow relative speed vehicle detection algorithm. The test platform was also the same. There are many common procedures for both relatively fast and relatively slow moving object detection algorithms. These common procedures include the capturing of image of size 640x480 each for storing to the PC for offline processing, using a JM18.4 video encoder for offline encoding the video to H.264/AVC 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 of 30fps, using the EPZS motion estimation algorithm, with intra-frame encoding in P-frame and macroblock partition smaller than 8x8 disabled.

Since the system will finally be implemented as an embedded system where the file input and output overhead for obtaining captured images and reading MVs will be eliminated, the timing analysis was focused on the computational loading of the algorithm rather than the time for input and output access.

The average time for processing the eight test sequences mentioned in Chapter 4.4.2 is shown in Table 4-14. It was found that the processing time for identifying the ROI and for HG only were having small variation. The ROI evaluation took approximately 30ms  $\pm$ 10% to complete, and the HG took approximately 4ms $\pm$ 10% to complete. However, the execution time for the HV or tracking algorithm varied from 20.1ms to 27.9ms. This was because the HV algorithm essentially consists of the block matching algorithm. It required longer time to complete when the block size was increased. For instance, the detected moving object was relatively small in Seq. T, hence the time required for block matching was relatively short. Also, the number of iterations required to find the matching block depends on the match result under the spiral search. If the matching block can be found earlier, the time required for HV and tracking becomes smaller. That is, if the estimated displacement is close to the actual displacement, the block matching algorithm used in HV and tracking can be completed earlier.

Table 4-14: Average processing time in ms for fast relative speed vehicle detection

<b>Sequence</b>	<b>Finding ROI</b>	<b>HG</b>	<b>HV or Tracking</b>
<b>Seq. P</b>	27.6	3.6	27.9
<b>Seq. Q</b>	30.7	3.8	21.3
<b>Seq. R</b>	27.8	4.0	20.1
<b>Seq. S</b>	28.6	3.8	21.1
<b>Seq. T</b>	30.5	4.3	17.4
<b>Seq. U</b>	32.2	4.0	20.3
<b>Seq. V</b>	29.5	4.0	26.4
<b>Seq. W</b>	31.1	4.2	27.3

Currently, the block matching algorithm is a simple spiral full-search, the search algorithm can be improved in the future by using more intelligent search algorithm.

Since the cycle time for vehicle detection can be finished within 66.7ms, the detection cycle is fast enough to catch up with the frame rate of 30fps for the H.264/AVC encoder with IBPBP frame structure. This means the video frame rate needs not be lowered to facilitate the detection, preserving the smoothness of the recorded video.

For a single system to have relatively fast and slow moving object detection functions, the system should be designed so that the fast and slow relative speed object detection algorithms are running alternatively, or in parallel with the use of multi-core system-on-chip.

#### **4.6 Cost Analysis**

The proposed system architecture of the MV based ADAS is shown in Figure 4-36. 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 close to the optical centre of the CMOS camera sensor. The inertial sensor is able to measure the three-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 H.264/AVC 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 two seconds. The DSP SOC is the heart of the system. It is responsible for executing all algorithms for lane detection and moving object detection.

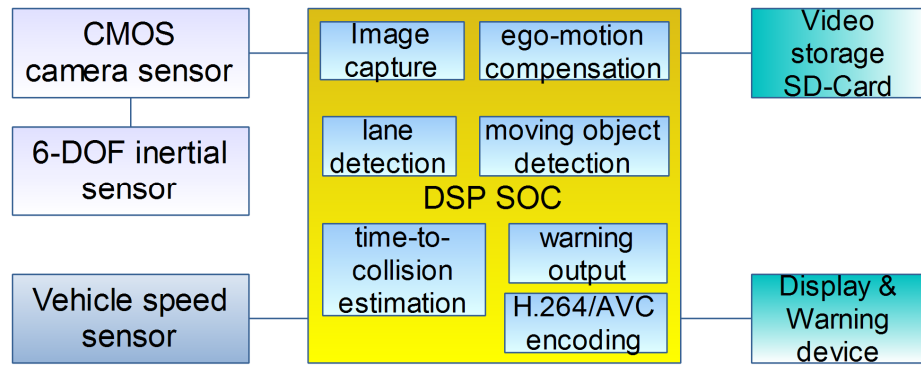


Figure 4-36: System architecture of the proposed MV based ADAS

The DSP SOC used in this project was DM3725 from Texas Instruments (TI, 2013). The SOC has an embedded ARM Cortex A8 processor core, a TMS320C64x DSP core and a dedicated hardware logic for H.264/AVC encoding. According to the official website of Texas Instruments, the reference unit price of the SOC was US\$27.20 at an order quantity of 1,000pcs (TI, 2015b). For vision based ADAS utilising optical flow technique (Giachetti and Campani et al., 1998, Klappstein and Stein et al., 2006), an additional DSP is expected to be used for the optical flow evaluation (Zhang and Gao et al., 2014). An estimate of the additional cost for the DSP is US\$26.53 (TI, 2015c). Similarly, feature-based ADAS can share-use the DSP SOC for both rear-end vehicle detection and H.264/AVC recording. However, the detection of vehicles or objects that have different shapes to the vehicles looking from the rear requires an additional DSP for the additional feature detection. Table 4-15 summarises the number of processors required to achieve the moving object detection and video recording functions using the solution proposed in this project, optical flow based methods, and feature based methods.

Table 4-15: Required processors for different ADAS solutions

<b>Function</b>	<b>Proposed solution</b>	<b>Optical flow solution</b>	<b>Feature-based solution</b>
<b>Relatively fast speed object detection</b>	1x DSP SOC DM3725	1x DSP for optical flow evaluation	1x DSP for handling additional feature recognition
<b>Relatively slow speed object detection</b>		1x DSP SOC DM3725	1x DSP SOC DM3725
<b>H.264/AVC Video Recording</b>			

Based on the number of embedded processors required for different ADAS solutions, the cost comparison of these solutions is shown in Table 4-16. The price for each component was reference to the official websites of the components, retail electronics component websites such as digikey.com, or quotations from component distributors. The comparison assumes that each embedded processor requires a set of SDRAM and NOR flash in order to function correctly. Therefore, the size of SDRAM and NOR flash memory were doubled in both the optical flow and feature based solutions. The comparison shows that the proposed solution is 50% less expensive than the optical flow or feature based solution.

Therefore, the proposed solution can achieve a lower cost than typical solutions making use of optical flow or feature based techniques. One point to note is that the research project is supported by the funding from the Innovation and Technology Commission of Hong Kong (ITC, 2015), and part of the funding of the project is supported by industry contributions. Conscientious and careful vetting processes have been gone through when the funding for the project was approved. The justifications on the cost competitiveness of the proposed solution have also been considered.

Table 4-16: Cost comparison for different ADAS solutions

Major component	Proposed solution	Optical flow solution	Feature-based solution
<b>Embedded Processor</b>	\$27.20 (1x DSP SOC)	\$53.73 (1x DSP + 1x DSP SOC)	\$53.73 (1x DSP + 1x DSP SOC)
<b>Inertial sensor</b>	\$4.02 (Invensense, 2015)	--	--
<b>Camera sensor</b>	\$20	\$20	\$20
<b>Power regulators</b>	\$5	\$5	\$5
<b>SDRAM</b>	\$17.78 (256MB x 2)	\$35.56 (256MB x 4)	\$35.56 (256MB x 4)
<b>NOR Flash</b>	\$14.0 (256MB)	\$28.0 (512MB)	\$28.0 (512MB)
<b>Passive components</b>	\$10.0	\$10.0	\$10.0
<b>Total</b>	\$98	\$152.29	\$152.29

## 4.7 Chapter Summary

This Chapter reported the test and evaluation results of the proposed algorithm for moving object detection, and the evaluation of the proposed camera calibration method.

With the techniques proposed in the algorithm to eliminate the problems with the imperfect MVs from typical H.264/AVC encoders, the detection performance is on a par to other methods found in the literature. The detection rate is higher than 90% under real and practical environment in Hong Kong. The computation time analysis shows the ability of the proposed algorithm for running in real time. The cost analysis shows the potential 50% reduction in cost with the use of the proposed algorithm and ADAS solution.

## 5 Commercialisation

A project “Development of Advanced Collision Avoidance (ITT/006/12AP)” to trial of the proposed solution has been conducted in Hong Kong with the funding support of amount HK\$800,000 from the Innovation and Technology Commission of Hong Kong. The objectives of the trial project were to collect the user experience to ADAS, and to identify problems of the hardware and software so that improvements could be done prior to mass production.

My major role was to develop the vehicle detection algorithm for this trial project for running in an embedded Digital Signal Processor (DSP). The calibration method proposed in this study was also applied to install the cameras to the wide variety of car models in this trail. The lane detection function (LDW) and the blind spot zone detection function (BSDS) that were developed by other colleagues in my organisation were also included in this trial project. The trial period was from February 2014 to August 2014. This project has invited four government departments and two non-governmental organisations (NGOs) to test the engineering prototypes. These government departments and NGOs included Fire Service Department (FSD), Water Supplies Department (WSD), Hong Kong Police Force (HKPF), Government Logistics Department (GLD), Hong Kong Society for Rehabilitation (HKSR), and the Neighbourhood Advice-action Council (NAAC). The car models that were used for testing the prototypes are shown in Figure 5-1.

This trial has provided a very good opportunity to refine the algorithm for running in real-time in a low-cost embedded Digital Signal Processor. It also has provided a chance to test the robustness of the algorithm and the embedded system for different road conditions in Hong Kong. Because of the tight schedule of the trial project and the



moving object detection algorithm was still under development, the moving object detection for trial included only the relatively slow speed moving vehicle detection algorithm in which the MV information for ROI reduction was omitted. This was because efforts were still in progress at that time to enable the MV output from the H.264/AVC encoder from the selected DSP. When the system was ready for trial, the relatively slow speed vehicle detection algorithm was able to run at 10fps in the embedded DSP. Although the processing speed could further be improved by utilising the hardware resources of the embedded DSP, such as its image pre-processing hardware for image colour conversion, image cropping, and histogram evaluation, there was not enough time for such engineering optimisation to be done. Nevertheless, the algorithm was able to work as expected due to the change of position of relatively slow speed moving vehicle is small in successive frames.

Figure 5-2 shows the components of ADAS for installation to test vehicles. These components include a camera installing to the windshield, a warning device for audible alert output and an embedded hardware prototype for running the algorithm.

The typical mounting position of the camera is shown in Figure 5-2(a). It was placed behind the rear-view mirror in the vehicle to prevent obstructing the view of the driver. The installation and calibration methods mentioned in Chapter 3.1.2 enabled our engineers to install the camera efficiently. The method also has provided a means to make sure of the installation quality and the correct estimation of the installation height of the camera. The warning device shown in Figure 5-2(b) was able to display and output audible warnings when the time-to-collision between the front vehicle and the ego vehicle were too close, or when lane departure events were detected. Figure 5-2(c) shows the embedded hardware prototype which was equipped with an embedded DSP

from Texas Instruments. Figure 5-2(d) shows the enclosure for the embedded hardware prototype, installing beneath a passenger seat in a test vehicle.



Figure 5-1: Government departments and NGOs that had participated in the Trial Project. (a) Hong Kong Police Force. (b) Water Supplies Department. (c) Hong Kong Society for Rehabilitation. (d) The Neighbourhood Advice-action Council. (e) Fire Services Department. (f) Government Logistics Department.

The drivers of those selected vehicles from different government departments and NGOs had the chance to evaluate the prototype on normal roads of their daily duties. There was no designated route assigned to the drivers, hoping to collect as many opinions from the drivers as possible. The evaluation results were collected by the Government Logistics Department and a trial evaluation form was returned after the trial period. According to the collected evaluation forms, the drivers commented that the system was useful. They also reported that there were false alarms, especially on single carriageways. This was because MV based ROI reduction technique mentioned in Chapter 3.4.1 was not used in the trial. Those vehicles driving in opposite directions to the ego vehicle were detected because of the existence of generic line features mentioned in Chapter 3.5.3. This led to undesirable excessive false detections.

The drivers also reported that the warning output could be annoying when there were warnings from both the three functions. Therefore, it is necessary to research on the warning output strategy due to the FCW, LDW and BSDS functions so that they can be harmonised to minimised unwanted distraction.

Many drivers reported that the hardware was not stable. It needed to be turned off for a while before turning it on again. This was because the embedded DSP was over-heated after prolonged operation, and the connectors used in this project were loosely due to shocks and vibrations. A heat-sink was installed to the embedded DSPs to dissipate the generated heat more efficiently. Better automotive grade connectors should also be used to make sure of the electrical and signal connectivity.

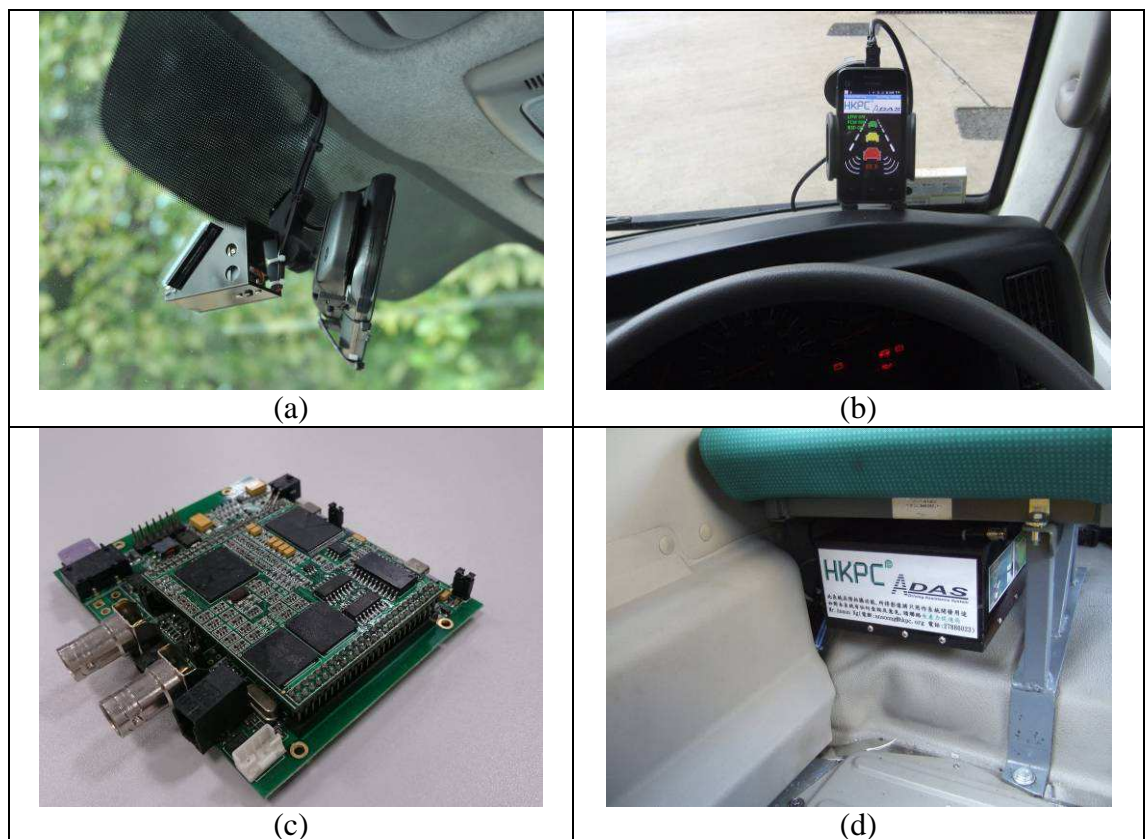


Figure 5-2: (a) Camera mounted to the windshield behind the rear-view mirror. (b) Warning device to output audible alerts to the driver. (c) Embedded DSP hardware prototype. (d) Prototype installation location.

## ***5.1 Chapter Summary***

This Chapter described the project that has been done to facilitate the commercialisation of the research results of this project. In particular, the project has tried out the developed algorithms for ADAS in different car models provided by the Hong Kong government. The tests were conducted on roads in Hong Kong. Drivers that have tried the system felt the system was useful. There was accumulated experience from this trial project. It provides useful information for continuous improvement of the system.

## 6 Publications and Patents

During the course of this research, two international conference papers and one international journal paper were either published or accepted, and one international journal was under preparation for submission. One of the two conference papers was published, and the other one was accepted. The submitted journal paper was also accepted for publication.

### Published Conference Paper

Wong, C.-C., Siu, W.-C., Barnes, S. & Jennings, P. Low relative speed moving vehicle detection using motion vectors and generic line features. *IEEE International Conference on Consumer Electronics*, 9-12 Jan. 2015 Las Vegas, pp. 208-209.

This paper summarised the interim result of the research of this study. It proposed the region of interest construction by making use of the amplitudes of MVs from a H.264/AVC encoder, and the detection of relatively slow moving objects using generic line features.

### Accepted Conference Paper

Wong, C.-C., Siu, W.-C., Barnes, S. & Jennings, P. Shared-Use Motion Vector Algorithm for Moving Objects Detection for Automobiles. *IEEE International Conference on Consumer Electronics*, 8-11 Jan. 2016 Las Vegas, forthcoming.

This paper summarised the algorithm framework proposed in this research on the shared use of MVs for moving object detection. It mentioned the techniques for dividing the detection task into the detection of relatively slow moving objects and relatively fast moving objects. The algorithm for relatively slow speed object detection and tracking, as well as that for relatively fast moving object detection and tracking were also mentioned

in details. This paper also included the test results on some challenging image sequences. The result indicated that the proposed method has detection rate above 90% which is on a par with state of the art methods proposed by other authors. The computation time was also analysed, indicating a real-time performance capability with typical cycle time of less than 66ms.

#### Accepted Journal Paper

Wong, C.-C., Siu, W.-C., Jennings, P., Barnes, S. & Fong, B. forthcoming 2015. A Smart Moving Vehicle Detection System Using Motion Vectors and Generic Line Features. *IEEE Transactions on Consumer Electronics*.

This journal paper mentioned the algorithm on the shared use of H.264/AVC MVs for relatively slow speed vehicle detection in details. It also outlined the algorithm for relatively fast speed moving object detection. Test results with image sequences containing challenging road conditions such as shadows, broken road and road-side fence were presented. It revealed the performance are on a par to algorithms proposed by other authors in terms of detection rate, and is computationally efficient for being used in a real-time system.

#### Journal Paper under Preparation

A journal paper of title “A Smart Block Based Road Region Detector for use in Vision Based Advanced Driver Assistance Systems” is under preparation. It will report the novel algorithm on block based road region detection mentioned in Chapter 3.3, as well as the improvements mentioned in Chapter 5.2.1.

## Patents

A Hong Kong Short-term patent of title “A Method and a Device for Detecting Moving Object” has been granted successfully. The application and grant numbers are 15106442.4 and 120328A respectively. In the mean time, the patent search report conducted by the State Intellectual Property Office (SIPO) of the People’s Republic of China (PRC) indicates that the invention is novel and innovative without any finding on infringement to intellectual properties. The invention therefore fulfils the criteria for patent registration to SIPO of PRC. The invention patent is under registration to the SIPO of the PRC at the time of writing this report.

### **6.1 Chapter Summary**

This Chapter mentioned two international conference papers and one international journal paper that are either published or accepted. It also mentioned an invention patent is filed in Hong Kong and is preparing to file in mainland China. One additional international journal paper is under preparation.

## 7 Future Improvements

There is room for improvements of the proposed algorithm framework for higher true-positive and lower false-positive detection rate, as well as for faster computation. The improvements will be done continuously by myself and my research team in the future.

### 7.1 Camera Calibration Method

Since the selected road surface may not be exactly on a flat level ground, the pitch angle measured by the inertial sensor inside the camera module may include the gradient of the road. Therefore, the offset due to the road gradient should be compensated so that the pitch angle of the camera with respect to the road surface is close to zero.

#### 7.1.1 Road Gradient Compensation

Figure 7-1 shows the situations where the vehicle is on a road with non-zero gradient. Figure 7-1(a) and (b) show the case with the car pointing upwards and downwards respectively. A method is proposed to remove the offset pitch angle due to the inclination of the road. The proposed method is to place the vehicle on the same road segment for two times with 180 degree opposite heading. The measured pitch angle for these two different headings can then be compensated for the offset due to the road inclination.

The pitch angle measured by the inertial sensor is  $\theta$ . Equation (7.1) and (7.2) show the measured sensor readings for the vehicle pointing uphill and downhill respectively.

Therefore, for the same road segment with the same vehicle facing in one direction, a value of  $\theta$  can be measured. After that, the same vehicle is facing another direction that is 180 degree opposite to the previous direction, another value of  $\theta$  can be measured.



The actual pitch angle of the camera can be estimated by combining Equation (7.1) and (7.2), where  $\theta_0$  is the inclination of the road,  $\theta_x$  is the camera pitch angle with respect to the road surface, and  $\theta_u$  is the pitch angle measured by the inertial sensor. Equation (7.3) and (7.4) are the results of combining Equation (7.1) and (7.2). They represent the formulae for calculating the camera pitch angle with respect to the road surface and the inclination of the road. By noticing the direction of the vehicle during calibration, the offset due to the road gradient can be eliminated. For example, the estimated road gradient is  $\theta_0=0.5$  degree, the target pitch angle  $\theta_x$  is zero, then the target reading from the inertial sensor  $\theta$  should be 0.5 degree.

$$\theta_u = \theta_x + \theta_0 \quad (7.1)$$

$$\theta_d = \theta_x - \theta_0 \quad (7.2)$$

$$\theta_x = \frac{\theta_u + \theta_d}{2} \quad (7.3)$$

$$\theta_0 = \frac{\theta_u - \theta_d}{2} \quad (7.4)$$

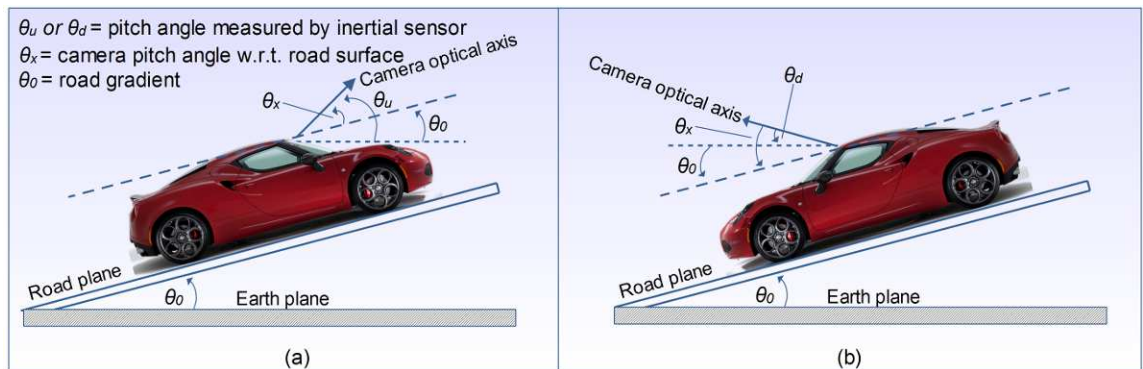


Figure 7-1: Illustration of road gradient affecting the measured pitch angle for the camera installation.

## **7.2 Road Region Detection**

### **7.2.1 Use of Temporal Information**

The computation time for road region estimation can be reduced by incorporating temporal information. Since the captured frames are continuous and the time interval between successive frames is small (typically 66ms), the road region will not be changed significantly from frame to frame.

Therefore, the road region identified in the previous frame can be reused so that the number of blocks to process for road region estimation can be reduced. By reducing the number of blocks to process, the computational time can be reduced. In this regard, some of the blocks along the boundaries of the identified road region can be invalidated according to the moving path of the ego vehicle.

The proposed improvement algorithm is illustrated in Figure 7-2 and Figure 7-3. Figure 7-2(a) shows the identified road region shaded in grey using the block based road region detection algorithm mentioned in Chapter 3.3. Figure 7-2(b) shows only the identified road region. The identified road region is reduced by one block along the road region contour. The yellow contour shows the difference between the original road region boundary and the diminished road region boundary. Figure 7-3 shows the diminished road region in Figure 7-2(b). The region growth algorithm will start from a block along the yellow contour. Since the number of blocks for processing is reduced, the road region identification algorithm can be completed in a shorter time.



Figure 7-2: Road region identified in a captured frame and the corresponding diminished road region for use in the next frame. (a) Captured frame with identified road region. The identified road region is shaded with grey colour blocks. (b) Extracted road region from (a). The boundary is diminished by 1 block along the contour. The yellow contour is the difference between the original and diminished contours.

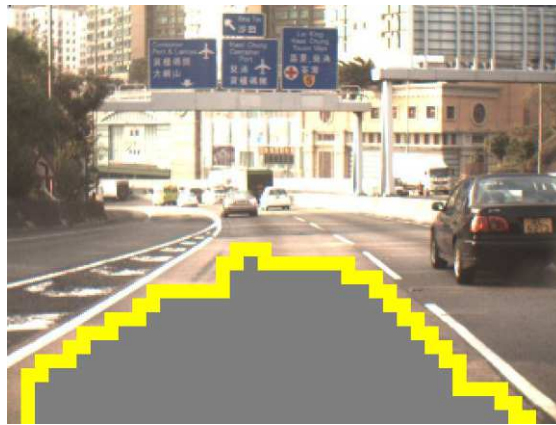


Figure 7-3: Overlaid road region found from the previous frame. The region growth algorithm will start from 1 block along the yellow contour, reducing the required computation time due to reduced number of blocks to process.

### **7.3 Segmentation Method for Relatively Slow and Fast Objects**

Currently, the proposed method splits the ROI into relatively slow and fast moving objects by means of the amplitude of the MVs only. A more intelligent segmentation method can be proposed to improve the ROI segmentation accuracy. For instance, the threshold of MV amplitude can be adaptive to the ego vehicle speed. That is, the threshold can be increased for higher speed.

The segmentation can also take other parameters into account. For instance, when there is a relatively slow speed vehicle detailed in the previous frame, the ROI in the current

frame can assume the region near the previously detected region is still the ROI for relatively slow speed moving objects.

#### ***7.4 Slow Relative Speed Moving Object Detection Algorithm***

The false-positive detection of invalid objects on the road can further be reduced by evaluating the symmetry of the area concerned.

There has been research on symmetry feature detection algorithms (Kuehnle, 1991, Marola, 1989, Zielke and Brauckmann et al., 1992). In particular, a symmetry measurement function by comparing the summed intensity value of the greyscale image for the left and right part of a selected region is described by Broggi and Cerri et al. (2004). They combined the symmetry measurement with Adaboost algorithm as the vehicle detector (Friedman and Hastie et al., 2000). Adaboost algorithm makes use of a series of weak classifiers that were trained by a set of data, to compare with a set of new data obtained in the current image. When the new set of data is not rejected by all the weak classifiers, the set of new data is accepted. Cheon and Yoon et al. (2012) introduced the symmetry measurement method by the histogram of oriented gradients (HOG). Each region of interest was divided into four regions, namely upper left, upper right, lower left and lower right. The HOG feature was then trained by the total error rate reduction polynomial model (TER-RM) proposed by Toh and Eng (2008).

For application to a real-time system, an objective comparison will be conducted before a suitable symmetry detection algorithm is selected. The selection will be based on the computation load and the associated symmetry detection performance.

## **7.5 Fast Relative Speed Moving Object Detection Algorithm**

### **7.5.1 Predictive Displacement Estimation**

Currently, the region for template matching is limited to 16x16 pixels around the averaged amplitude and direction of MVs of the identified region which may contain relatively fast moving objects. Since the target object is moving, the average amplitude and direction of the MVs should be compensated by the predicted speed change (or acceleration) of the object. The change in speed of the detected object can be predicted by the amplitude of MVs of previous frames. With a basic kinematic model for the movement of the moving object, the prediction can be refined by the use of Kalman filter.

### **7.5.2 The Use of Bi-Prediction Motion Vector**

In the current implementation, the MVs from P-frames were used to evaluate the movements of objects in successive images. As discussed in Chapter 2, MVs from H.264/AVC can be erroneous as they were used primarily for video compression rather than precision object motion estimation. One of the very important pieces of information in B-frames is bi-prediction motion vectors of some macroblocks. The bi-prediction MVs can be used to refine the accuracy of MVs in P-frames. By taking into account the spatial and temporal consistency of object movements in successive frames, MV outliers can be detected and eliminated. In this connection, the false positive rate can further be reduced.

One point to note is that MVs are estimated using the current frame as the coordinate reference. The coordinates in the current frame are always in integer form, aligning to the macroblock boundaries. For a bi-predictive macroblock in a B-frame, each pixel in the macroblock is the average sum of the pixels in the displaced coordinates according

to the two bi-predictive MVs. The bi-predictive MVs can be of sub-pixel units of up to 1/4 precision rather than integer. Also, they are not limited to be aligned to the macroblock boundaries.

Therefore, to utilise the bi-predictive MVs in B-frames to improve the reliability of MVs in P-frames, the coordinate reference of the MVs in the B-frame has to be changed to use the current P-frame as reference. This involves the construction of a MV map by rounding or truncating the sub-pixel MVs to integer pixels. After that, the MV map can be used to compare with the MVs obtained from the current P-frame. MVs in the current P-frame can be discarded if the discrepancies are larger than certain thresholds, indicating they are temporally inconsistent to the corresponding MVs found in the previous B-frame.

### **7.5.3 The Algorithm for Template Matching**

Currently, the search algorithm for template matching is performed by simple exhaustive spiral search about the estimated displacement from the position in the previous frame. The search can be computationally inefficient especially when a match template cannot be found inside the defined search range. The speed of the search algorithm can be improved by employing smarter search algorithm. Similar fast search algorithms for H.264/AVC motion estimation can be considered, such as the Diamond Search algorithm (Tham and S. et al., 1998), UMHexagonS algorithm (Chen and Xu et al., 2006), and Image Edge Assisted Search algorithm (Chan and Siu, 2001). According to the test results of these authors, all these fast search algorithms can reduce the computation cost by at least 80%.

Also, the decision on successful template matching relies on the evaluation of sum of absolute difference (SAD) of the image region under comparison. The computation

cost for SAD evaluation can be reduced by using integral image (Schweitzer and Bell et al., 2002, Viola and Jones, 2001). The properties of Integral Image allow rapid and efficient computation. Figure 7-4 illustrates how the Integral Image can facilitate rapid computation of sums and differences of rectangular regions. Figure 7-4(a) shows a rectangle at location from  $P_0(0,0)$  to  $P(x,y)$ . The sum of pixel values above and on the left of  $P(x,y)$  is  $P_{x,y}$  as expressed in Equation (7.5).

$$P_{x,y} = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (7.5)$$

where  $i(x', y')$  is the pixel value at point  $(x', y')$ . The relationship of area A, B, C and D to the point  $P_1$ ,  $P_2$ ,  $P_3$  and  $P_4$  as shown in the Integral image in Figure 7-4(b) can be expressed as Equation (7.6).

$$P_1 = A, P_2 = A + B, P_3 = A + C, P_4 = A + B + C + D \quad (7.6)$$

By solving these equations, D can be expressed as Equation (7.7).

$$D = P_1 + P_4 - P_2 - P_3 \quad (7.7)$$

So, the area  $D$  can be evaluated very quickly by knowing the sum of pixel values at the four corners of  $D$ .

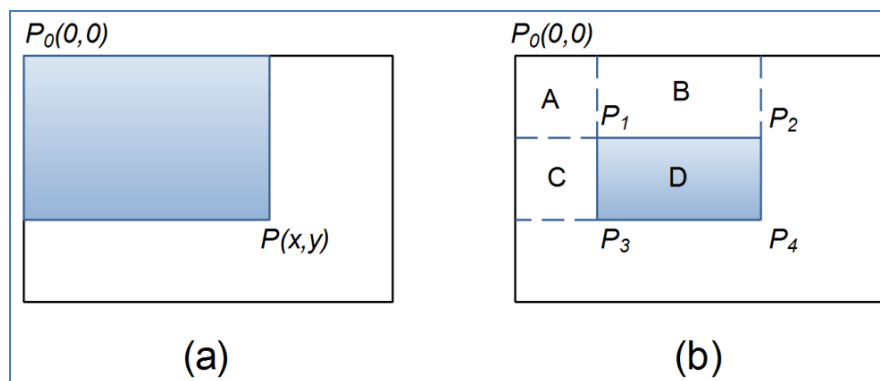


Figure 7-4: Integral image. (a) sum of pixel values above and left of  $P(x,y)$ , (b) sum of pixel values above and left of  $P_4(x,y) = A+B+C+D$

In addition to the use of search algorithm, the moving direction of the identified template can also be estimated from the MVs from the H.264/AVC encoder. During the HV mode, the template identified from HG mode can be evaluated for the moving direction of blocks inside the template by referring to the MVs from the H.264/AVC encoder. Since the MVs from H.264/AVC encoder are referring to the movement from the current frame to the previous frame, a reverse MV map needs to be constructed so that the movement of the blocks in the template from the previous frame to the current frame can be evaluated. After that, if the resultant movement of the whole template is consistent to the estimated movement, the HV can be regarded as successful. Similarly, tracking can perform in a similar way as in the HV mode.

#### **7.5.4 Reducing False-Positive Detection**

As mentioned in Chapter 4.4.4, there were cases of false-positive HG and HV for MV based moving object detection. The false-positive hypothesis can pass the HV stage because the identified displacement in the HV stage is within the allowable limit of the estimated displacement evaluated in the HG stage, even though the estimated displacement found in HG is erroneous.

The false positive detection can be reduced by evaluating the direction of movement of the template in HV stage. If the movement of the template from the HG frame to the HV frame is moving away from the ego vehicle, the HV can be treated as failed. And if the direction of movement of the template is pointing to the FOE, and the amplitude of the displacement is the same as the displacement due to ego motion, the HV can also be regarded as unsuccessful.



## **7.6 The Use of Stereo Camera**

With the use of stereo camera, a scene can be captured by two cameras simultaneously. The depth information of objects in the scene can be obtained by evaluating the pixel disparity between the two images. Disparity refers to the displacement of pixels on the same object appearing on the pair of images captured simultaneously by the two cameras.

The disparity evaluation can be done in the un-compressed image domain using different block matching techniques. The most trivial matching cost is to evaluate the difference between pixels one by one along the epipolar line. However, this trivial method requires absolute intensity constancy of images taken by the two cameras. Such absolute intensity constancy is not achievable in practice because of the manufacturing tolerances of components for the cameras such as lens, sensors and mechanical enclosure, as well as the fact that the light reflecting from objects reaching the two cameras can be different. Most stereo matching cost functions are block based so as to reduce the effect of intensity difference of the two cameras. Common matching cost functions used for finding stereo correspondence are the sum of absolute difference (SAD), sum of squared difference (SSD), normalised cross correlation (NCC) and the sampling insensitive absolute difference or known as BT algorithm (Birchfield and Tomasi, 1998).

In addition to evaluating disparity in the image domain, it can also be evaluated in the compressed domain. H.264/AVC encoders support efficient stereo camera video encoding using the Stereo High Profile. In stereo view scenarios, there are many similarities between the two camera views. Coding efficiency gain can be achieved by using one of the camera inputs as reference, and storing the difference of the input from the second camera by either P-frame or B-frame encoding. Disparity of stereo

images can be evaluated by looking into the MVs between the pair of images. Hence, the depth maps of the pair of video can be obtained. Although there are few research on depth map estimation using stereo H.264/AVC stream, Pourazad et al. (2010) showed that it is possible in a published paper.

With the availability of depth information, more MVs that potentially are outliers can be discarded. For instance, the clustering process can include the depth information as an input parameter so that only MVs of similar depth will be considered for the same cluster. Also, the selection of the final cluster for hypothesis verification can be based on the distance to the ego-camera, selecting the one that is closest to the ego-vehicle.

### ***7.7 Improving H.264/AVC Motion Estimation***

The motion estimation algorithm used in the H.264/AVC encoder involves some search and evaluation methods to find the most appropriate motion vectors in order to achieve the lowest bit rate possible.

Since the ego motion of the camera can be estimated from the built-in inertial sensor and the signal from the vehicle speed sensor, the motion estimation algorithm can be improved by making use of the information from these sensors. In particular, fast motion estimation algorithms available in the JM18.4 H.264/AVC encoder, such as MVFAST (Tourapis and Au et al., 2000), UMHS (Chen and Zhou et al., 2002) and EPZS (Tourapis and Cheong et al., 2005), use predictive information from the video stream to reduce the computation cost on block based motion search. The predictive information is obtained from the spatial and temporal movements of adjacent blocks in successive frames. The movement of a particular block cannot be obtained from adjacent blocks at the boundaries of independently moving objects.

There has been research on the use of inertial sensors for video encoding (Chen and Zhao et al., 2011, Angelino and Cicala et al., 2013, Wang and Ma et al., 2012). The experimental results by Chen and Zhao et al. (2011) showed that the computational cost for motion estimation can be reduced significantly by transforming the camera motion into the predictor for motion estimation. Since the search direction and search range can be reduced, the computational cost is reduced. According to the experiment result of Angelino and Cicala et al.'s sensor-assist motion estimation method, the computation time for motion estimation can be reduced by at least 50% comparing to the UMHS fast search method.

Further research on this direction can reduce the computational cost as well as the accuracy of the output motion vectors to represent actual object movement without sacrificing the video coding efficiency. The computational cost can be reduced because the motion search direction and range can be estimated from the inertial sensors and the vehicle speed sensor, allowing less number of tries for finding the best match. Similarly, the motion vector accuracy can be improved because the first guess for direction and amplitude of motion can be estimated from the inertial sensor and the vehicle speed sensor, reducing the chance of trapping into an irrelevant local minimum.

### ***7.8 Extension to Next Generation Video Encoder***

The proposed algorithm relies on the MV output from video encoders that conform to the H.264/AVC standard, it is likely that the encoder will be phased out sometime in the future and be replaced by encoders of newer standard to cope with the ever increasing demands for higher screen resolution and coding efficiency. The block based MV output format used in the proposed algorithm ensures the extensibility of the algorithm to work with next generation video encoders.

## 7.8.1 Comparison to H.265/HEVC Encoding Standard

Comparing with H.264/AVC standard, the emerging High Efficiency Video Coding (HEVC) standard can reduce video bit rate to up to 50% without loss of video quality (Sullivan and Ohm et al., 2012). The different arrangements in block structure for the two encoding standards are shown in Figure 7-5. As shown in the figure, the major difference between H.264/AVC and HEVC standard is that the coding block size in H.264/AVC is fixed at 16x16, while it can be 8x8 to 64x64 in HEVC. The variable block size in HEVC is known as Coding Unit (CU) that shown in the Quadtree Coding Structure in Figure 7-5. The selection of block size for motion estimation in HEVC standard is arranged in a coding tree unit (CTU) and coding tree block (CTB) structure. One frame is divided into a series of non-overlapped CTU. The size of a CTB can be chosen as 16x16, 32x32 or 64x64 samples. The CTB may contain one or multiple coding units (CUs). Similarly, each CU may contain one or multiple prediction units (PUs). The size of each PU varies from 64x64 down to 4x4 samples (Sullivan and Ohm et al., 2012).

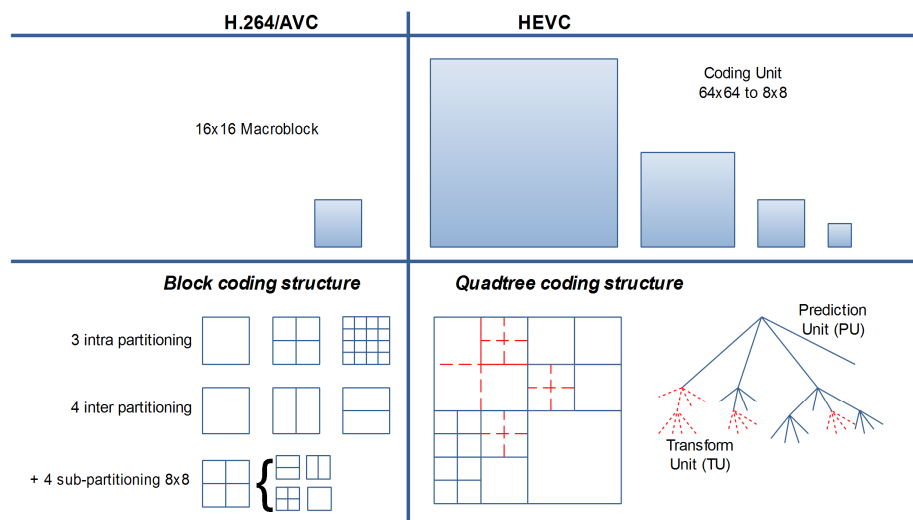


Figure 7-5: Comparison of block structure for H.264/AVC and HEVC

In addition to the motion estimation process to find the MV to relate the current frame to previous frames, there is a special mode known as SKIP mode in H.264/AVC

encoders. No motion information is present in this SKIP mode. The information for the block concerned is obtained from the co-located block in the previous frame. In contrast, MVs in HEVC are evaluated by either spatial or temporal approach. For an intercoded PU, the encoder can decide to use motion estimation mode or motion merge mode.

The motion estimation mode is the same as that in H.264/AVC encoders. It evaluates the motion vectors according to images from previous frames. Motion merge mode is newly introduced in HEVC, which is an improvement on SKIP mode. A set of previously coded neighbouring PUs is used to form a list of candidates that are either spatially or temporally close to the current PU. After deciding which candidate in the list is suitable to be used, the motion information of the selected candidate is used directly for the current PU. Therefore, the index of the selected candidate in the list is encoded; no intensive motion search algorithm is required. By encoding only the index, very few bits are required to code the difference between frames.

Bi-prediction motion estimation is employed in both H.264/AVC and HEVC. It makes use of two sets of motion data to generate two MVs from different reference images. The resultant motion compensated block is the weighted sum of the two motion compensated blocks using the two MVs. Different weight can be applied to the two motion compensated blocks in order to accommodate for different scenarios, such as reflections and sudden light intensity changes.

Although there are different motion estimation algorithms available in the free reference software for HEVC and H.264/AVC encoders, they are provided for reference only. Among the motion estimation algorithms available in the reference encoder software, Test Zone Search (TZS) algorithm is provided as a reference fast search algorithm to speed up the motion estimation process (Tourapis, 2002). Since

there is no specified motion estimation algorithm for the video standards, many motion estimation methods have been studied and proposed by many researchers to achieve different computation complexity and video quality targets.

The increased coding efficiency in HEVC is the result of increased encoder complexity. The high computational cost on mode decision and motion estimation require parallelised hardware and more computationally efficient algorithms to enable the HEVC encoders to be used for real-time applications.

### **7.8.2 Shared-Use of Motion Vector from HEVC Encoder**

Although the complexity of the HEVC encoder has considerably increased to achieve a higher coding efficiency, it can be regarded as an improvement based on the experience gained from previous coding standards, such as H.264/AVC. One of the most important observations is that both H.264/AVC and HEVC encoders work by dividing one frame into multiple non-overlapping small blocks. The motion estimation process will also output MVs with different block size to best represent the underlying motions of objects in the screen.

The block size of MV used in the proposed algorithm is 8x8 samples, the HEVC encoder can be modified to output inter-frame MV of block size 8x8, using the method proposed in Chapter 3.1.1. With the MV output from the HEVC encoder following the format proposed in this project, the algorithm can be extended to be used in newer generation video encoders. The computation cost for motion estimation in the HEVC encoder can also be enhanced by making use of the information from the inertial sensors and vehicle speed sensor. The efforts paid on enhancing H.264/AVC encoders can therefore be re-used in the new HEVC encoder.

## **7.9 Chapter Summary**

This Chapter has elaborated the potential improvements that can be made to the proposed algorithm. These include an improved camera calibration method, road region detection with temporal consistency consideration, adaptive ego vehicle speed region of interest segmentation and adding symmetry detection for slow relative speed vehicle detection. For relatively fast speed moving object detection, adding predictive algorithm to the estimation of displacement of MVs, faster template matching algorithm, and rejection of non-critical detected objects by temporal movement evaluation, can be considered to improve the detection rate and computational speed, as well as to reduce the false alarm rate. Improvements to the motion estimation algorithm in the H.264/AVC encoder, and future extension in using the MVs from the newer HEVC encoder are also considered. With continuous efforts putting to the research on ADAS, these suggested improvements will be tested and evaluated in the near future.

## 8 Future R&D Directions

The future R&D direction will be to refine the performance of the developed algorithms in terms of detection rate, false positive rate, as well as computation efficiency. Potential improvements mentioned in Chapter 7 will be realised to improve the performance.

Since HEVC is the standard for next generation high-definition video, the future R&D direction will be on better integration of the developed algorithm with HEVC. There have been studies on the use of inertial sensors to reduce the computational cost on motion estimation for video encoders (Chen and Zhao et al., 2011, Angelino and Cicala et al., 2013, Wang and Ma et al., 2012). It is expected that the use of inertial sensors and speed sensors can reduce the computation cost of HEVC for automotive applications. Since the motion prediction can be taken from these sensors, it is expected that the resulting MVs can describe the movements of objects more accurately without loss of video compression efficiency. A new System-on-Chip with HEVC encoding making use of sensor inputs can be less computationally expensive. The power consumption for video encoding can also be reduced. This design will not only be beneficial to automobiles looking for lower power consumption, but also to battery powered handheld devices such as cameras and smartphones.

Working with multiple systems in a vehicle is another direction that needs to be addressed. Currently, the system assumes there is no other system that outputs warnings or overrides controls from the driver. The complicated on-board automobile system will include a human-machine-interface that integrates all warnings, system status, and available user preferences. The strategies on warning outputs, communication and coordination among different driver assistance systems are also required to be developed.



## 9 Further Works on Product Commercialisation

The algorithms developed are porting to an embedded system for commercialisation. Since the image resolution has been increased to 1280x720 in the target system from 640x480 during the development for commercialisation, optimisation on the algorithms is required to perform the detection in real-time. The target product is a complete ADAS with lane departure warning (LDW), relative slow speed vehicle detection and relative fast speed moving object detection. The porting of LDW and relative slow speed moving object detection has been completed. While the first trial samples for the embedded hardware has been developed, second trial samples are under development with modifications to correct the problems found in the first trial samples. In parallel to the hardware development, the integration of relative fast speed moving object detection algorithm is also in progress. At the same time, there are other supporting functions, such as image quality, lens dirt detection, adverse weather detection and night time detection, are required to be included in the system to help determine whether the detection result is reliable.

In addition, the image pre-processing functions available from the embedded System-on-Chip will also be utilised to reduce the computation overhead. Imaging functions such as colour space conversion, resizing, cropping and image histogram evaluation can be done by the hardware for use by different algorithms in the system.

Since multiple algorithms are required to run in parallel, the system will utilise the multi-core System-on-Chip to run multiple threads in parallel. The optimisation on processor load of multiple cores as well as the utilisation of the memory bandwidth is also ongoing.

To further improve the reliability of the detection, a self-testing and auto-calibration algorithm will also be developed. This algorithm aims to detect if the camera is mounted correctly, and if the inertial sensors are working properly. This is because there is chance that the camera will be misaligned due to long term usage. When misalignment is detected, the system can self-calibrate itself to make sure the orientation of the camera is usable. Otherwise, a fault should be reported to ask for repair or servicing.

In the meantime, a demonstration system will also be built to show the features of the system to facilitate the commercialisation.

## 10 Conclusions

The objective of this research was to develop a low-cost Advanced Driver Assistance System by the shared-use of motion vectors from a H.264/AVC encoder. It was identified via the literature review on vision based ADAS. This project was funded by the Innovation and Technology Commission of Hong Kong via a conscientious and careful vetting process.

The challenges of using MVs from the H.264/AVC encoder included the difficulty in detecting moving object due to small and imprecise MVs on moving objects with relatively slow moving speed to the ego vehicle, and the erroneous MVs due to the fact that the motion estimation algorithm of the H.264/AVC encoder is designed for highest video compression ratio rather than for precise motion estimation of objects in the scene.

The main contributions of this research are the methods proposed for moving object detection given the limitations of MVs from H.264/AVC encoders. By separating the captured image into ROI for relatively slow and fast moving object detection respectively, the proposed algorithm has solved the problem of difficult object detection due to small MV on moving objects with relatively slow speed to the ego vehicle. Relatively slow moving objects are detected by the proposed algorithm with the use of generic line features of vehicles. It allows a wide variety of vehicles with different shape and size to be detected, such as passenger cars, minivans, trucks, coaches and buses. The resulting detection rate and false positive rate are on a par to other state-of-the-art algorithms proposed and published by other authors. Analysis also shows that the cost of the proposed system is 50% less expensive than systems utilising optical-flow or feature-based techniques.

The algorithm for fast relative speed object detection works in conjunction with the detection of moving objects that may not be covered by the slow relative speed moving object detection algorithm. The detection is based on the amplitude and direction of planar parallax residuals of MVs of the macroblocks inside the ROI. The detection phase consists of a Hypothesis Generation stage for finding the template dynamically for matching in the Hypothesis Verification stage in the next frame. Hypothesis Verification is achieved by the template matching algorithm and dynamic template re-generation in the successive frames. This algorithm solves the problem of erroneous MVs generated by the H.264/AVC encoder that will lead to wrongly detected moving object.

The road region detection algorithm serves to reduce the ROI for moving object detection, thereby reducing the computational time and the chance of having false positive detection. The threshold value for creating the binary image for initial vehicle location determination and for the Canny edge image for template matching is also determined during the process of road region detection. The innovation of the proposed road region detection algorithm is based on the use of block based rather than pixel based evaluation. This approach reduces the required computational cost to help achieve the real-time performance.

A six-degree-of-freedom inertial sensor is incorporated in the camera module. The sensor is able to measure the three dimensional angular speed, acceleration and orientation of the camera. By making use of the readings from the sensor, the transformation matrix between the camera coordinates and the World coordinates can be deduced more accurately for ego motion compensation. The sensor is also utilised in the camera calibration method proposed in this study.

As the system was targeted to operate at real-time, the computational cost was also evaluated. The result shows that the algorithm can be completed in 66ms, within the time duration of successive frames.

A new camera calibration method was also proposed in this study to facilitate the practical needs of consistent camera installation quality and the accurate estimation of camera intrinsic and extrinsic parameters.

A trail project has also been carried out for commercialisation preparation. Although a prototype with all the features proposed in this project could not be put into trail due to the time constraints, the collection of user experience and identification of areas for improvements will be useful to the future commercialisation.

A Hong Kong Short-term patent was filed successfully. The associated patent search report indicates that the invention is innovative and there is no infringement to intellectual properties. The invention is under registration to the SIPO of the PRC. Finally, there remain some areas where the algorithm can be improved in terms of computational cost, detection rate and false positive rate. These include the use of temporal information to speed up the road region detection algorithm, using a better strategy for splitting the ROI for relatively fast and slow moving objects, adding symmetry evaluation to further reduce the false alarm rate, predictive displacement estimation of relatively fast moving object during successive tracking, faster search algorithm for template matching, and improvement to the motion estimation algorithm of H.264/AVC encoder using the information from the inertial sensor and vehicle speed sensor. These improvements to the algorithm will be investigated by the author and the research team in his organisation.

The specific conclusions from this research are as follows:

- A low-cost ADAS that share-uses MVs from the H.264/AVC encoder has been developed. It matches with the objective of this research. Analysis shows that the cost of the proposed system is 50% less expensive than systems utilising other techniques
- An original and novel algorithm to address the problems of the use of MVs from the H.264/AVC encoder for moving object detection has been proposed. Test results show that the detection rate is on a par to state-of-the-art algorithms proposed by other authors. This is also the main contribution of this research.
- A six-degree-of-freedom inertial sensor was built into the camera unit to assist the ego motion estimation and focus of expansion.
- A novel camera calibration procedure was developed. It has provided a systematic approach for engineers so that the camera can be installed and calibrated consistently.
- The novelty of this research is illustrated by the successful publications of a peer review journal paper and a conference paper, a successfully filed short-term patent in Hong Kong with the patent search report mentioning the eligibility of the invention for registration towards the SIPO of the PRC.
- A trial project has been carried out to collect the user feedback and identify problems of the system prototype. The algorithm has proven to be useful and improvements to the system will be made continuously during the commercialisation of the system.

## 11 References

- Alvarez, J. M. A., & Lopez, A. M., "Road Detection Based on Illuminant Invariance," *IEEE Trans. on Intell. Transp. Syst.*, vol. 12, no. 1, pp. 184-193, 2011.
- Angelino, C. V., Cicala, L., De Mizio, M., Leoncini, P., Baccaglioni, E., Gavelli, M., Raimondo, N. & Scopigno, R., "Sensor aided H.264 Video Encoder for UAV applications," in *Picture Coding Symposium*, 2013, pp. 173-176.
- Apostoloff, N. & Zelinsky, A., "Robust vision based lane tracking using multiple cues and particle filtering," in *Proc. IEEE Intell. Veh. Symp.*, 2003, pp. 558-563.
- Baehring, D., Simon, S., Niehsen, W. & Stiller, C., "Detection of close cut-in and overtaking vehicles for driver assistance based on planar parallax," in *IEEE Proc. Intell. Veh. Symp.*, 2005, pp. 290-295.
- Bay, H., Tuytelaars, T. & Van Gool, L., "Surf: Speeded up robust features," *Computer vision—ECCV 2006*, pp. 404-417: Springer, 2006.
- Bendix. 2015. *AutoVue Lane Departure Warning System by Bendix CVS* [Online]. Available: <http://www.bendix.com/en/products/autovue/AutoVue.jsp> [Accessed 29 Jul. 2015].
- Bertozzi, M. & Broggi, A., "GOLD: a parallel real-time stereo vision system for generic obstacle and lane detection," *IEEE Trans.: Image Processing*, vol. 7, no. 1, pp. 62-81, 1998.
- Birchfield, S. & Tomasi, C., "A pixel dissimilarity measure that is insensitive to image sampling," *Pattern Analysis and Machine Intelligence: IEEE Transactions*, vol. 20, no. 4, pp. 401-406, 1998.
- Broggi, A., Cerri, P. & Antonello, P. C., "Multi-resolution vehicle detection using artificial vision," in *Proc. IEEE Intell. Veh. Symp.*, Parma, Italy, 2004, pp. 310-314.
- Brown, M. Z., Burschka, D. & Hager, G. D., "Advances in computational stereo," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 25, no. 8, pp. 993-1008, 2003.
- Bruhn, A., Weickert, J. & Schnörr, C., "Lucas/Kanade meets Horn/Schunck: Combining local and global optic flow methods," *Intl. Journal of Computer Vision*, vol. 61, no. 3, pp. 211-231, 2005.
- CarNewsChina.com. 2015. *FAW Besturn B30 will launch in China in November* [Online]. Available: <http://www.carnewschina.com/2015/07/22/faw-besturn-b30-will-launch-in-china-in-november/> [Accessed 28 Jul. 2015].
- Chan, Y. L. & Siu, W. C., "An efficient search strategy for block motion estimation using image features," *IEEE Trans. on Image Process.*, vol. 10, no. 8, pp. 1223-1238, 2001.
- Chang, W.-C. & Cho, C.-W., "Online Boosting for Vehicle Detection," *IEEE Trans. on Systems, Man, and Cybernetics*, vol. 40, no. 3, pp. 892-902, 2010.

- Chapuis, R., Aufrere, R. & Chausse, F., "Accurate road following and reconstruction by computer vision," *IEEE Trans. on Intell. Transp. Syst.*, vol. 3, no. 4, pp. 261-270, 2002.
- Chen, D.-Y., Chen, G.-R. & Wang, Y.-W., "Real-time dynamic vehicle detection on resourcelimited mobile platform," *Computer Vision, IET*, vol. 7, no. 2, pp. 81-89, 2013.
- Chen, M., Jochem, T. & Pomerleau, D., "AURORA: a vision-based roadway departure warning system," in Proc. IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems 95. 'Human Robot Interaction and Cooperative Robots', 1995, pp. 243-248.
- Chen, X., Zhao, Z., Rahmati, A., Wang, Y. & Zhong, L., "Sensor-Assisted Video Encoding for Mobile Devices in Real-World Environments," *IEEE Trans. on Circuits and Syst. for Vid. Tech.*, vol. 21, no. 3, pp. 335-349, 2011.
- Chen, Z., Xu, J., He, Y. & Zheng, J., "Fast integer-pel and fractional-pel motion estimation for H.264/AVC," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 264-290, 2006.
- Chen, Z., Zhou, P. & He, Y., "Fast integer pel and fractional pel motion estimation for JVT," *JVT-F017*, pp. 5-13, 2002.
- Cheng, H.-Y., Jeng, B.-S., Tseng, P.-T. & Fan, K. C., "Lane Detection With Moving Vehicles in the Traffic Scenes," *IEEE Trans. on Intell. Transp. Syst.*, vol. 7, no. 4, pp. 571-582, 2006.
- Cheon, M., Lee, W., Yoon, C. & Park, M., "Vision-Based Vehicle Detection System With Consideration of the Detecting Location," *IEEE Trans. on Intell. Transp. Syst.*, vol. 13, no. 3, pp. 1243-1252, 2012.
- Chi, Y. M., Tran, T. D. & Etienne-Cummings, R., "Optical Flow Approximation of Sub-Pixel Accurate Block Matching for Video Coding," in Acoustics, Speech and Signal Processing, IEEE International Conference (ICASSP 2007), 2007, pp. I-1017-I-1020.
- Chiu, M. Y. & Siu, W. C., "Computationally-scalable motion estimation algorithm for H.264/AVC video coding," *IEEE Trans. on Consumer Electron.*, vol. 56, no. 2, pp. 895-903, 2010.
- Civera, J., Davison, A. J., Martínez Montiel, J. M., Davison, A. & Martínez Montiel, J., "Self-calibration Structure from Motion using the Extended Kalman Filter," Springer Tracts in Advanced Robotics, pp. 111-122: Springer Berlin / Heidelberg, 2012.
- Clauss, M., Bayerl, P. & Neumann, H., "Segmentation of independently moving objects using a maximum-likelihood principle," *Autonome Mobile Systeme 2005*, pp. 81-87: Springer, 2006.
- Cortes, C. & Vapnik, V., "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.
- Davis, C. Q., Karul, Z. Z. & Freeman, D. M., "Equivalence of subpixel motion estimators based on optical flow and block matching," in Proc. IEEE Intl. Symp. on Computer Vision, 1995, pp. 7-12.



- del-Blanco, C. R., Garcia, N., Salgado, L. & Jaureguizar, F., "Object Tracking from Unstabilized Platforms by Particle Filtering with Embedded Camera Ego Motion," in Sixth IEEE Intl. Conf. on Advanced Video and Signal Based Surveillance, 2009, pp. 400-405.
- Derpanis, K. G., "Characterizing image motion," York University, 2006.
- Du, Y. & Papanikolopoulos, N. P., "Real-time vehicle following through a novel symmetry-based approach," in Proc.: IEEE Intl. Conf. on Robotics and Automation, 1997, pp. 3160-3165.
- El Ansari, M., Stéphane, M. & Abdelaziz, B., "Temporal consistent real-time stereo for intelligent vehicles," *Pattern Recognition Letters*, vol. 31, no. 11, pp. 1226-1238, 2010.
- Finlayson, G. D., Hordley, S. D., Cheng, L. & Drew, M. S., "On the removal of shadows from images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 1, pp. 59-68, 2006.
- Freescale. 2015. *i.MX 6 VPU Application Programming Interface Linux Reference Manual* [Online]. Freescale. Available: [https://www.freescale.com/webapp/Download?colCode=L3.10.53\\_1.1.0\\_LINUX\\_DOCS&location=null&fasp=1&WT\\_TYPE=Supporting%20Information&WT\\_VENDOR=FREESCALE&WT\\_FILE\\_FORMAT=gz&WT\\_ASSET=Documentation&fileExt=.gz&Parent\\_nodeId=1337694700967726419044&Parent\\_pageType=product](https://www.freescale.com/webapp/Download?colCode=L3.10.53_1.1.0_LINUX_DOCS&location=null&fasp=1&WT_TYPE=Supporting%20Information&WT_VENDOR=FREESCALE&WT_FILE_FORMAT=gz&WT_ASSET=Documentation&fileExt=.gz&Parent_nodeId=1337694700967726419044&Parent_pageType=product) [Accessed 28-Mar 2015].
- Freund, Y. & Schapire, R. E., "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.
- Friedman, J., Hastie, T. & Tibshirani, R., "Additive logistic regression: a statistical view of boosting," pp. 337-407, Apr., 2000.
- Giachetti, A., Campani, M. & Torre, V., "The use of optical flow for road navigation," *Proc.: IEEE Intl. Conf. on Robotics and Automation*, vol. 14, no. 1, pp. 34-48, 1998.
- Harris, C. & Stephens, M., "A combined corner and edge detector," in Alvey Vision Conference 1988, pp. 50.
- Hartley, R. & Zisserman, A., *Multiple view geometry in computer vision*: Cambridge University Press, 2003.
- Haselhoff, A. & Kummert, A., "A vehicle detection system based on Haar and Triangle features," in Proc. IEEE Intell. Veh. Symp., 2009, pp. 261-266.
- He, Y., Wang, H. & Zhang, B., "Color-based road detection in urban traffic scenes," *IEEE Trans. on Intell. Transp. Syst.*, vol. 5, no. 4, pp. 309-318, 2004.
- Heikkila, J., "Geometric camera calibration using circular control points," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1066-1077, 2000.
- Heinly, J., Dunn, E. & Frahm, J.-M., "Comparative evaluation of binary features," *Computer Vision–ECCV 2012*, pp. 759-773: Springer, 2012.
- Horn, B. K. & Weldon Jr, E., "Direct methods for recovering motion," *Intl. Journal of Computer Vision*, vol. 2, no. 1, pp. 51-76, 1988.

- Horn, B. K. P. & Schunck, B. G., "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185-203, 1981.
- Huang, C.-L. & Chen, Y.-T., "Motion estimation method using a 3D steerable filter," *Image and Vision Computing*, vol. 13, no. 1, pp. 21-32, 1995.
- Ieng, S., Tarel, J. P. & Labayrade, R., "On the design of a single lane-markings detectors regardless the on-board camera's position," in Proc. IEEE Intell. Veh. Symp., 2003, pp. 564-569.
- Ikonomakis, N., Plataniotis, K. N. & Venetsanopoulos, A. N., "Color image segmentation for multimedia applications," *Journal of Intelligent and Robotic Systems*, vol. 28, no. 1-2, pp. 5-20, 2000.
- Invensense. 2015. *InvenSense 6-Axis Gyro+Accel: MPU-6050 US\$4.02* [Online]. Available: <http://store.invensense.com/ProductDetail/MPU-6050-InvenSense-Inc/422200/pid=1135> [Accessed 15 Sep. 2015].
- ITC. 2015. *Innovation and Technology Commission of Hong Kong* [Online]. Available: <http://www.itf.gov.hk/1-eng/WhatsNew.asp?textmode=0> [Accessed 14 Sep. 2015].
- Jazayeri, A., Hongyuan, C., Jiang Yu, Z. & Tuceryan, M., "Vehicle Detection and Tracking in Car Video Based on Motion Model," *IEEE Trans. on Intell. Transp. Syst.*, vol. 12, no. 2, pp. 583-595, 2011.
- Jung, B. & Sukhatme, G. S., "Detecting moving objects using a single camera on a mobile robot in an outdoor environment," in, 2004, pp. 980-987.
- JVT. 2012. *Joint Video Team - H.264/AVC Reference Software version JM18.4* [Online]. Available: [http://iphome.hhi.de/suehring/tml/download/old\\_jm/](http://iphome.hhi.de/suehring/tml/download/old_jm/) [Accessed 31 Dec. 2014].
- Ke, Q. & Kanade, T., "Transforming camera geometry to a virtual downward-looking camera: robust ego-motion estimation and ground-layer detection," in Proc.: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition, 2003, pp. 390-397.
- Klappstein, J., Stein, F. & Franke, U., "Monocular Motion Detection Using Spatial Constraints in a Unified Manner," in Proc. IEEE Intell. Veh. Symp., 2006, pp. 261-267.
- Kluge, K. & Lakshmanan, S., "A deformable-template approach to lane detection," in Proc. IEEE Intell. Veh. Symp., 1995, pp. 54-59.
- Kosecka, J., Blasi, R., Taylor, C. J. & Malik, J., "A comparative study of vision-based lateral control strategies for autonomous highway driving," in Proc.: IEEE Intl. Conf. on Robotics and Automation, 1998, pp. 1903-1908.
- Kuehnle, A., "Symmetry-based recognition of vehicle rears," *Pattern Recognition Letters*, vol. 12, no. 4, pp. 249-258, 1991.
- Kuo, Y. C., Pai, N. S. & Li, Y. F., "Vision-based vehicle detection for a driver assistance system," *Comput. & Math. with Applications*, vol. 61, no. 8, pp. 2096-2100, 2011.
- Langer, M. & Mann, R., "Optical Snow," *Intl. Journal of Computer Vision*, vol. 55, no. 1, pp. 55-71, 2003/10/01, 2003.

- Lee, J. & Crane, C. D., "Road Following in an Unstructured Desert Environment Based on the EM(Expectation-Maximization) Algorithm," in SICE-ICASE International Joint Conference, 2006, pp. 2969-2974.
- Li, Q., Zheng, N. & Cheng, H., "Springrobot: a prototype autonomous vehicle and its algorithms for lane detection," *IEEE Trans. on Intell. Transp. Syst.*, vol. 5, no. 4, pp. 300-308, 2004.
- Li, X., Fang, X., Wang, C. & Zhang, W., "Lane Detection and Tracking Using a Parallel-snake Approach," *Journal of Intelligent & Robotic Systems*, vol. 77, no. 3-4, pp. 597-609, 2015/03/01, 2015.
- Lim, K. H., Seng, K. P., Ang, L.-M. & Chin, S. W., "Lane Detection and Kalman-Based Linear-Parabolic Lane Tracking," in IEEE Intl. Conf. on Intell. Human-Machine Systems and Cybernetics, 2009, pp. 351-354.
- Liu, G., Worgotter, F. & Markelic, I., "Stochastic Lane Shape Estimation Using Local Image Descriptors," *IEEE Trans. on Intell. Transp. Syst.*, vol. 14, no. 1, pp. 13-21, 2013.
- Liu, H., Hong, T. H., Herman, M., Camus, T. & Chellappa, R., "Accuracy vs efficiency trade-offs in optical flow algorithms," *Computer Vision and Image Understanding*, vol. 72, no. 3, pp. 271-286, 1998.
- Liu, T., Zheng, N., Zhao, L. & Cheng, H., "Learning based symmetric features selection for vehicle detection," in Proc. IEEE Intell. Veh. Symp., 2005, pp. 124-129.
- Longuet-Higgins, H. C., "The Reconstruction of a Plane Surface from Two Perspective Projections," in Proc. of the Royal Society of London. Series B, Biological Sciences, 1986, pp. 399-410.
- Lowe, D. G., "Distinctive image features from scale-invariant keypoints," *Intl. Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.
- Lu, M., Wevers, K. & Wang, J., "Implementation Road Map for In-Vehicle Safety Systems in China," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2148, pp. 116-123, 2010.
- Lucas, B. D. & Kanade, T., "An iterative image registration technique with an application to stereo vision," in Proc.: Intl. Joint Conf. on Artificial Intelligence, 1981, pp. 674-679.
- Lucchese, L., "Geometric calibration of digital cameras through multi-view rectification," *Image and Vision Computing*, vol. 23, no. 5, pp. 517-539, 2005.
- Manuel Ibarra Arenado, Juan Maria Perez Oria, Carlos Torre-Ferreiro & Renteria, L. A., "Monovision-based vehicle detection, distance and relative speed measurement in urban traffic," *Intelligent Transport Systems, IET*, vol. 8, no. 8, pp. 655-664, 2014.
- Mao, L., Xie, M., Huang, Y. & Zhang, Y., "Preceding vehicle detection using Histograms of Oriented Gradients," in IEEE Intl. Conf. on Comm., Circuits and Syst., 2010, pp. 354-358.
- Marola, G., "Using symmetry for detecting and locating objects in a picture," *Computer Vision, Graphics, and Image Processing*, vol. 46, no. 2, pp. 179-195, 1989.

- McCall, J. C. & Trivedi, M. M., "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Trans. on Intell. Transp. Syst.*, vol. 7, no. 1, pp. 20-37, 2006.
- Mitsunaga, T. & Nayar, S. K., "Radiometric self calibration," in IEEE Comput. Soc. Conf. on Comput. Vision and Pattern Recognition, 1999, pp. 380.
- Mobileye. 2015. *Mobileye C2-270 Advanced Driver Assistance System* [Online]. Available: <http://www.mobileye.com/products/mobileye-c2-series/mobileye-c2-270/> [Accessed 27 Aug. 2015].
- Muad, A. M., Hussain, A., Samad, S. A., Mustaffa, M. M. & Majlis, B. Y., "Implementation of inverse perspective mapping algorithm for the development of an automatic lane tracking system," in IEEE Region 10 Conference (TENCON), 2004, pp. 207-210.
- Nedevschi, S., Golban, C. & Mitran, C., "Improving accuracy for Ego vehicle motion estimation using epipolar geometry," in IEEE Intl. Conf. on Intell. Transportation Systems, 2009, pp. 1-7.
- Pauwels, K. & Van Hulle, M. M., "Segmenting independently moving objects from egomotion flow fields," *Isle of Skye, Scotland*, 2004.
- Pernek, A. & Hajder, L., "Perspective Reconstruction and Camera Auto-Calibration as Rectangular Polynomial Eigenvalue Problem," in Intl. Conf. on Pattern Recognition, 2010, pp. 49-52.
- Pourazad, M. T., Nasiopoulos, P. & Ward, R. K., "Generating the Depth Map from the Motion Information of H.264-Encoded 2D Video Sequence," *EURASIP Journal on Image and Video Processing*, vol. 2010, no. 1, pp. 1-13, 2010.
- Ramstrom, O. & Christensen, H., "A method for following unmarked roads," in Proc. IEEE Intell. Veh. Symp., 2005, pp. 650-655.
- Rosten, E. & Drummond, T., "Machine learning for high-speed corner detection," *Computer Vision—ECCV 2006*, pp. 430-443: Springer, 2006.
- Rotaru, C., Graf, T. & Zhang, J., "Color image segmentation in HSI space for automotive applications," *Journal of Real-Time Image Processing*, vol. 3, no. 4, pp. 311-322, 2008.
- Rublee, E., Rabaud, V., Konolige, K. & Bradski, G., "ORB: An efficient alternative to SIFT or SURF," in IEEE Intl. Conf. on Computer Vision, 2011, pp. 2564-2571.
- Scaramuzza, D. & Siegwart, R., "Appearance-Guided Monocular Omnidirectional Visual Odometry for Outdoor Ground Vehicles," *IEEE Trans on Robotics*, vol. 24, no. 5, pp. 1015-1026, 2008.
- Schweitzer, H., Bell, J. & Wu, F., "Very fast template matching," *Computer Vision—ECCV 2002*, pp. 358-372: Springer, 2002.
- Sehestedt, S., Kodagoda, S., Alempijevic, A. & Dissanayake, G., "Efficient lane detection and tracking in urban environments," in European Conference on Mobile Robots, 2007, pp. 126-131.
- Shi, J. & Tomasi, C., "Good features to track," in Proc.: IEEE Comput. Soc. Conf. on Computer Vision and Pattern Recognition, 1994, pp. 593-600.

- Sirisantisamrid, K., Tirasesth, K. & Matsuura, T., "A technique of camera calibration using single view," in Intl. Conf. on Control, Automation and Systems (ICCAS), 2011, pp. 1486-1490.
- Sivaraman, S. & Trivedi, M. M., "A General Active-Learning Framework for On-Road Vehicle Recognition and Tracking," *IEEE Trans. on Intell. Transp. Syst.*, vol. 11, no. 2, pp. 267-276, 2010.
- Sivaraman, S. & Trivedi, M. M., "Looking at Vehicles on the Road: A Survey of Vision-Based Vehicle Detection, Tracking, and Behavior Analysis," *IEEE Trans. on Intell. Transp. Syst.*, vol. 14, no. 4, pp. 1773-1795, 2013.
- Sotelo, M. A., Rodriguez, F. J. & Magdalena, L., "VIRTUOUS: vision-based road transportation for unmanned operation on urban-like scenarios," *IEEE Trans. on Intell. Transp. Syst.*, vol. 5, no. 2, pp. 69-83, 2004.
- Stein, G. P., Mano, O. & Shashua, A., "A robust method for computing vehicle ego-motion," in IEEE Proc. Intell. Veh. Symp., 2000, pp. 362-368.
- Su, Y., Sun, M.-T. & Hsu, V., "Global motion estimation from coarsely sampled motion vector field and the applications," *IEEE Trans. on Circuits and Syst. for Vid. Tech.*, vol. 15, no. 2, pp. 232-242, 2005.
- Sullivan, G. J., Ohm, J., Woo-Jin, H. & Wiegand, T., "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. on Circuits and Syst. for Vid. Tech.*, vol. 22, no. 12, pp. 1649-1668, 2012.
- Sun, T.-Y., Tsai, S.-J. & Chan, V., "HSI color model based lane-marking detection," in IEEE Intell. Transportation Systems Conf., 2006, pp. 1168-1172.
- Sun, Z., Bebis, G. & Miller, R., "Monocular precrash vehicle detection: features and classifiers," *IEEE Trans. on Image Process.*, vol. 15, no. 7, pp. 2019-2034, 2006.
- Tan, C., Hong, T., Chang, T. & Shneier, M., "Color model-based real-time learning for road following," in IEEE Intell. Transportation Systems Conf., 2006, pp. 939-944.
- Tham, J. Y., S., R., M., R. & A., K. A., "A novel unrestricted center-biased diamond search algorithm for block motion estimation," *IEEE Trans. on Circuits and Syst. for Vid. Tech.*, vol. 8, no. 4, pp. 369-377, 1998.
- TI. 2013. *DM3730 / DM3725 Digital Media Processors (Rev. D)* [Online]. Available: <http://www.ti.com/product/dm3730> [Accessed 6-Jun. 2013].
- TI. 2015a. *Application Parameter Settings for TMS320DM365 H.264 Encoder* [Online]. Available: <http://www.ti.com/dsp/docs/litabsmultiplefilelist.tsp?sectionId=3&tabId=409&literatureNumber=spraba9&docCategoryId=1&familyId=1300&keyMatch=h.264%20encoder&tisearch=Search-EN-Everything> [Accessed 28-Mar 2015].
- TI. 2015b. *DM3725 Digital Media Processor Sample Buy* [Online]. Available: <http://www.ti.com/product/DM3725/samplebuy> [Accessed 14 Sep. 2015].
- TI, "TMS320DM6437ZWTQ6 Digital Media Processor Sample and Buy US\$26.53," 2015c.
- Toh, K.-A. & Eng, H.-L., "Between Classification-Error Approximation and Weighted Least-Squares Learning," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 30, no. 4, pp. 658-669, 2008.

- Toulminet, G., Bertozzi, M., Mousset, S., Benschraier, A. & Broggi, A., "Vehicle detection by means of stereo vision-based obstacles features extraction and monocular pattern analysis," *IEEE Trans. on Image Process.*, vol. 15, no. 8, pp. 2364-2375, 2006.
- Tourapis, A. M., "Enhanced predictive zonal search for single and multiple frame motion estimation," in *Electronic Imaging 2002*, 2002, pp. 1069-1079.
- Tourapis, A. M., Au, O. C. & Liou, M. L., "Predictive motion vector field adaptive search technique (PMVFAST): enhancing block-based motion estimation," in *Photonics West 2001-Electronic Imaging*, 2000, pp. 883-892.
- Tourapis, A. M., Cheong, H.-Y. & Topiwala, P., "Fast ME in the JM reference software," in *JVT Document JVT-P026*, 16th Meeting, 2005, pp. 24-29.
- Trucco, E. & Verri, A., *Introductory techniques for 3-D computer vision*: Prentice Hall, 1998.
- Tsai, R., "A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Journal of Robotics and Automation*, vol. 3, no. 4, pp. 323-344, 1987.
- Tzomakas, C. & Seelen, W. v., "Vehicle Detection in Traffic Scenes Using Shadows," *Ir-Ini, Institut Fur Nueroinformatik, Ruhr-Universitat.*, 1998.
- Vaudrey, T., Rabe, C., Klette, R. & Milburn, J., "Differences between stereo and motion behaviour on synthetic and real-world stereo sequences," in *IEEE Intl. Conf. on Image and Vis. Comput.*, New Zealand, 2008, pp. 1-6.
- Viola, P. & Jones, M., "Rapid object detection using a boosted cascade of simple features," in *Proc.: Computer Vision and Pattern Recognition*, IEEE Computer Society Conference, 2001, pp. 511-518
- Wang, C.-C. R. & Lien, J. J. J., "Automatic Vehicle Detection Using Local Features - A Statistical Approach," *IEEE Trans. on Intell. Transp. Syst.*, vol. 9, no. 1, pp. 83-96, 2008.
- Wang, G., Ma, H., Seo, B. & Zimmermann, R., "Sensor-assisted camera motion analysis and motion estimation improvement for H. 264/AVC video encoding," in *Proc. of the 22nd Intl. workshop on Network and Operating System Support for Digital Audio and Video*, 2012, pp. 89-94.
- Wang, Q., Zhang, Q. & Rovira-Mas, F., "Auto-Calibration Method to Determine Camera Pose for Stereovision-Based Off-Road Vehicle Navigation," *Environment Control in Biology*, vol. 48, no. 2, pp. 59-72, 2010.
- Wang, Y., Bai, L. & Fairhurst, M., "Robust Road Modeling and Tracking Using Condensation," *IEEE Trans. on Intell. Transp. Syst.*, vol. 9, no. 4, pp. 570-579, 2008.
- Wang, Y., Teoh, E. K. & Shen, D., "Lane detection and tracking using B-Snake," *Image and Vision Computing*, vol. 22, no. 4, pp. 269-280, 4/1/, 2004.
- Wang, Z. L., Cai, B. G., Du, X. L., Ou, S. & Zhao, J., "A robust multistage ego-motion estimation," in *5th Intl. Congress on Image and Signal Processing*, 2012, pp. 138-142.

- Weickert, J. & Schnörr, C., "A theoretical framework for convex regularizers in PDE-based computation of image motion," *Intl. Journal of Computer Vision*, vol. 45, no. 3, pp. 245-264, 2001.
- Wen, X., Shao, L., Fang, W. & Xue, Y., "Efficient Feature Selection and Classification for Vehicle Detection," *IEEE Trans. on Circuits and Syst. for Vid. Tech.*, vol. 25, no. 3, pp. 508-517, 2015.
- Wiegand, T., Sullivan, G. J., Bjontegaard, G. & Luthra, A., "Overview of the H.264/AVC video coding standard," *IEEE Trans. on Circuits and Syst. for Vid. Tech.*, vol. 13, no. 7, pp. 560-576, 2003.
- Xiaqiong, Y., Xiangning, C. & Heng, Z., "Accurate Motion Detection in Dynamic Scenes Based on Ego-Motion Estimation and Optical Flow Segmentation Combined Method," in *Symp. on Photonics and Optoelectronics*, 2011, pp. 1-4.
- Yim, Y. U. & Se-young, O., "Three-feature based automatic lane detection algorithm (TFALDA) for autonomous driving," *IEEE Trans. on Intell. Transp. Syst.*, vol. 4, no. 4, pp. 219-225, 2003.
- Yong, X., Zhang, L., Song, Z., Hu, Y., Zheng, L. & Zhang, J., "Real-time vehicle detection based on Haar features and Pairwise Geometrical Histograms," in *IEEE Intl. Conf. on Information and Automation*, 2011, pp. 390-395.
- Yuan, Q., Thangali, A., Ablavsky, V. & Sclaroff, S., "Learning a Family of Detectors via Multiplicative Kernels," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 3, pp. 514-530, 2011.
- Zeng, H., Cai, C. & Ma, K.-K., "Fast Mode Decision for H.264/AVC Based on Macroblock Motion Activity," *IEEE Trans. on Circuits and Syst. for Vid. Tech.*, vol. 19, no. 4, pp. 491-499, 2009.
- Zhang, F., Gao, Y. & Bakos, J. D., "Lucas-Kanade Optical Flow estimation on the TI C66x digital signal processor," in *IEEE High Perf. Extreme Comput. Conf. (HPEC)*, 2014, pp. 1-6.
- Zhang, J.-j. & Zhang, G.-q., "The New Situation of Road Safety in China," in *Intl. Conf. on Logistics Engineering and Intelligent Transportation Systems (LEITS)*, 2010, pp. 1-5.
- Zhang, Z., "A flexible new technique for camera calibration," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, 2000.
- Zhang, Z., "Camera calibration with one-dimensional objects," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, pp. 892-899, 2004.
- Zielke, T., Brauckmann, M. & von Seelen, W., "Intensity and edge-based symmetry detection applied to car-following — ECCV'92," *Lecture Notes in Computer Science Sandini, G., ed.*, pp. 865-873: Springer Berlin / Heidelberg, 1992.