A Three-Step Vehicle Detection Framework for Range Estimation Using a Single Camera

Ritesh Kanjee
Optronic Sensor Systems
Defence, Peace, Safety and Security
Council for Sci. & Ind. Research
ritesh.kanjee@icloud.com

Asheer K. Bachoo
Research Associate Tsotsos Lab
Lassonde School of Engineering
York University
abachoo@gmail.co.za

Johnson Carroll
Engineering and the Built Environment
University of Johannesburg
jcarroll@uj.ac.za

***Abstract*—This paper proposes and validates a real-time on-road vehicle detection system, which uses a single camera for the purpose of intelligent driver assistance. A three-step vehicle detection framework is presented to detect and track the target vehicle within an image. In the first step, probable vehicle locations are hypothesized using pattern recognition. The vehicle candidates are then verified in the hypothesis verification step. In this step, lane detection is used to filter vehicle candidates that are not within the lane region of interest. In the final step tracking and online learning are implemented to optimize the detection algorithm during misdetection and temporary occlusion. Good detection performance and accuracy was observed in highway driving environments with minimal shadows.**

I. Introduction

Vehicle detection for a vision-based driver assistance system requires the analysis of video images using image processing algorithms to isolate and track moving vehicles in the video sequence. Monocular vision based vehicle detection systems are particularly interesting for their low cost and for the high-fidelity information they give about the driver environment.

In this paper, we present the vehicle detection as a generalised three-step vehicle detection framework to detect, verify and track a lead vehicle within an image sequence. Various vehicle detection studies are described and relate to the framework in Section II, then the authors' own implementation is presented in the subsequent sections. The headway region is calculated from these image coordinates in conjunction with lane detection geometry. After the vehicle detection process, the lead vehicle's range is then calculated using the geometry of the lane detected in the previous step, and a Kalman filter is then used to attenuate the jitter from the range signal. Section III discusses the image processing algorithms and implementation of the three-step vehicle detection framework. Thereafter, experimental results of the implemented framework are shown in Section IV. Finally, in Section V, the results are summarized and conclusions are drawn.

II. The Three-Step Detection Framework

Most studies on vision-based vehicle detection address one or more of the following three steps, which form the basis of the following generalised vehicle detection framework:

978-1-4799-7560-0/15/$31 ©2015 IEEE

1) Hypothesis Generation (HG) - Targets that may be vehicles are identified within the image,
2) Hypothesis Verification (HV) - Potential vehicles are classified and non-vehicle objects are eliminated,
3) Algorithm Optimisation (AO) - The HG and HV steps are improved through optimisation techniques, such as tracking the relevant vehicles' coordinates from frame to frame.

Each step may consist of various combination of algorithms which ultimately achieves similar outputs (i.e. find/detect vehicle hypotheses, verify and isolate vehicle hypotheses, track/monitor vehicle hypotheses). Once the appropriate lead vehicle has been detected, the distance between vehicles is estimated so that appropriate driver assistance actions can be taken.

A. Hypothesis Generation

Sun *et al.* reviews on-road vehicle detection studies and groups them into three main Hypothesis Generation (HG) techniques: knowledge-based, stereo-based and motion-based [1]. For the purposes of this study, only the monocular methods are relevant, and the literature is notably dominated by knowledge-/feature-based methods. In particular, shadow, wheel and vertical edge information can be used to define basic features for initial object hypotheses [2]–[5]. Using shadows as an initial HG step seems problematic since the vehicle's shadow positions may change depending on sunlight inclination, as well being influenced by other shadows. However, this flaw is compensated for by combining with other HG or HV steps such as edge detection and symmetry. Han *et al.* [6] uses a method of detecting edges using Haar-like features, and proves better than the Sobel edge filter method prescribe by Duan *et al.* [7]. Other knowledge-based HG methods include gradient-based methods and Adaboost classification [8]–[10]. Sivaraman *et al.* also reviews and summarises the main monocular vision-based vehicle detection algorithms [11]. As noted in [1], appearance-based detection is widely used over motion-based detection methods; as of [11], the most prominent appearance-based algorithms used for vehicle HG are Histogram of Gradients (HOG) features, Haar-like features, Speeded-Up Robust Features (SURF) and Gabor features. Most of these algorithms are used to hypothesize strong edge and symmetry features of the vehicle [11].

B. Hypothesis Verification

Classical HV methods for vehicle detection frequently utilise symmetry (as most vehicles are symmetrical) [1], potentially in combination with other image features such as Haar-like edge features [6]. However, symmetry is reliable only under certain target vehicle orientation conditions [6]. Another classical method defines a region of interest (ROI) based on characteristics of the detection scenario. Any potential vehicles outside the ROI are eliminated as either non-vehicles or non-target vehicles.

Learning algorithms, such as support vector machines [7], [12], [13] and Adaboost/genetic algorithms [8]–[10] can also be used to separate potential vehicles into either "vehicle" or "non-vehicle" classes. Such HV algorithms depend on pre-sorted training sets to then automatically classify potential vehicles, and are often computationally intense. In other recent studies, Histograms of Oriented Gradients (HOG) and Pyramid of Histograms of Oriented Gradients (PHOG) features are used for vehicle HV [14]–[16].

C. Algorithm Optimisation Techniques

The aim of the third step is to initially eliminate non-detections that could occur when the HG module fails, and secondly to be able to identify the same vehicle over time [8]. A successful tracking algorithm can define a new ROI for each potential vehicle according to the object's speed and recent position using knowledge-based, as well as template-based methods [7], [17], [18]. Alternately, the previously estimated vehicle's position can be stored and maintained for a fixed period before declaring that the vehicle track is lost, allowing for partial occlusions [19]. Such tracking methods can also be used simultaneously on multiple objects [7], and the characteristics of motion can be incorporated into the HV methods [20]. Such strategies can be used in conjunction with algorithms traditionally utilised for object tracking, such as the Kalman Filter (KF), extended KF, particle filtering, optical flow and feature-based tracking [11].

D. Range Estimation

Single camera vehicle range estimation is most commonly implemented by triangulation using perspective and road geometry. Stein *et al.* obtain longitudinal range by using the known camera height and the point of contact of a lead vehicle to calculate the clearance range [21]. Nakamura *et al.* propose an algorithm that uses both horizontal and vertical triangulation, reducing errors due to the vehicle pitch [22].

III. Three-Step Vehicle Detection Implemented

The presented design (see Figure 1) initiates by capturing an image from a camera. A three-step vehicle detection framework is used to detect the appropriate target vehicle in the image. The first step (HG) uses pattern matching to generate a set of candidate vehicles within the image.

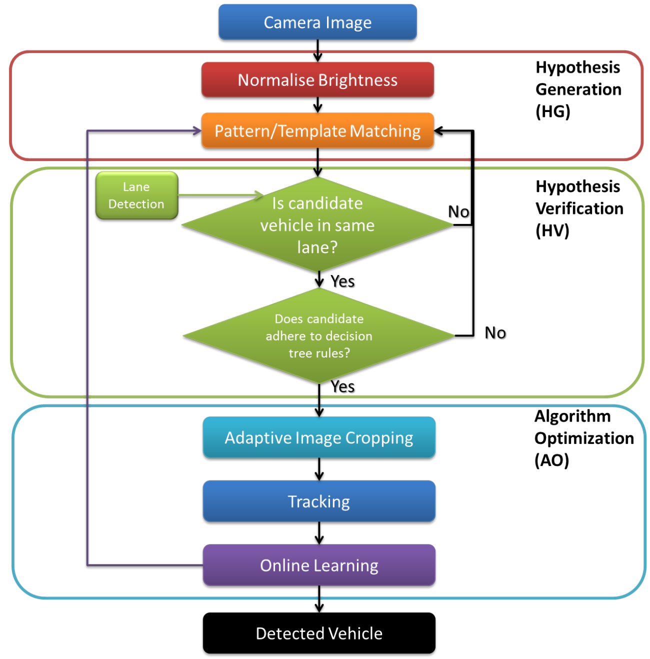

Figure 1. Flowchart of the vehicle detection algorithm design.

The second step is to identify which of these candidate objects are in fact vehicles on the road. This is accomplished by filtering the image to search for only the candidate object within the bounds of the lane markings. Vehicles or candidate objects that are not in the same lane as the ACC vehicle are discarded. This ensures that the detector targets the vehicle directly in front of it and eliminates multiple targets. Thereafter, a decision tree evaluates the candidates based on prior knowledge of the location of the vehicle candidate within the image.

The final step of the vehicle detection framework is the algorithm optimisation process. This step consists of Adaptive Image Cropping (AIC), video sequence tracking and online learning. AIC involves speeding up the performance of the algorithm by limiting the search area to the region of the detected vehicle, similar to the method described by Sotelo *et al.* [19]. A tracking method, adapted from the same study, is used to avoid losing track of the vehicle due to misdetection or partial occlusion. Online learning is used to optimise the detection in the hypothesis generation stage by cropping a positive vehicle match from the input video feed and using that cropped image as part of the template database. This method is similar to the method to Betke *et al.* [20].

Thereafter, from the detected vehicle's image coordinates, range can be estimated using perspective and geometry. The range estimation algorithm uses a triangulation method as described previously, but the width of the lane where the vehicle meets the road is used instead of the width or height of the vehicle [20]–[22]. This helps eliminate the error due to variation in a vehicle's height or width.

A. Hypothesis Generation

1) Brightness Control: Before vehicle pattern matching is performed, the light intensity of the image has to be normalised to compensate for the brightness variation that is caused by various daylight lighting conditions [6]. A Look-Up Table (LUT) transformation was used for normalising the brightness of each image in the video sequence.

2) Vehicle Pattern Matching: Pattern matching provides a definitive and a relatively robust method of detecting a specific object in a scene, such as vehicles. Pattern or template matching works by searching for a portion of the image that resembles a predefined template corresponding to the desired vehicle. The normalised cross correlation (NCC) is the most common way to determine if a portion of the image matches the template [20], [23].

$$r = \frac{p_{T_i} \sum_{x,y} R_n(x,y) T_i(x,y) - (\sum_{x,y} R_n(x,y))(\sum_{x,y} T_i(x,y))}{\sqrt{p_{T_i} \sum_{x,y} R_n(x,y)^2 - (\sum_{x,y} R_n(x,y))^2}} \times \frac{1}{\sqrt{p_{T_i} \sum_{x,y} T_i(x,y)^2 - (\sum_{x,y} T_i(x,y))^2}}, \tag{1}$$

where p_{T_i} is number of pixels in the template image T_i that have non-zero brightness values and x, y are image template location [20]. The NCC is dimensionless, and $|r| \leq 1$. Template T_i and region R_n and are perfectly correlated if $r = 1$; in practice, a threshold is chosen to determine whether a match is positive, minimum match score (MMS).

With a pattern matching approach, a database of possible visual patterns of vehicles are created to seek similarity between a segment of the actual video frame and a database image, or template [24]. The templates of the lead vehicle are captured at various scales and orientations, under various environmental conditions. Approximately 14 to 20 templates were used. A larger database of templates may result in a robust system (depending on the quality of templates, assuming translational motion and minor amounts of rotation in some cases), but this comes at a hardware and performance cost trade-off. Each sub-image in the video frame is matched with each template, so more templates may result in longer processing. On the other hand, the number of templates can be reduced by decreasing the MMS. However, reducing the MMS increases the risk of false detections.

The pattern matching detection method returns the $(x,\ y)$ coordinates of the candidate vehicles within the image, as well as the bounding box enclosing each template. However, our implementation uses the detected vehicle's $(x,\ y)$ coordinates in conjunction with the lane width to determine range. In order to enable pattern matching without massive template databases and sufficiently fast processing, additional methods can be used to reduce false detections. For this study, false positives are filtered using a combination of lane detection and tracking. These techniques are discussed in the following sections.

B. Hypothesis Verification

1) Lane Detection: A lane detection algorithm was implemented to verify if the vehicle match is within the same lane as the following vehicle. This is particularly important if multiple vehicles are detected, and also increases execution speed for the vehicle template matching. The resultant lines that are estimated from lane detection are used to define a ROI to ignore vehicle matches that are not within the same lane as the following vehicle. The lane detection algorithm works by first implementing an edge detection filter which accentuates the lane markings and thus makes it easier to detect the lanes. Another pattern matching algorithm was used to detect lane markings, and the bounding box of the matches was used to fit straight line segments. Once the segments were detected, a running average was used to reduce jitter and the lines were extended to approximately identify the entire lane.

2) Decision Tree: A decision tree is another HV method to eliminate false positives and verify target vehicles [25]. A decision tree is basically a set of rules or assumptions about the vehicle for which tests are performed to verify the correctness of a hypothesis. In our system, template matches are discarded if they overlap the edge of the lane or the extreme bottom or top of the image, as these positions are considered unlikely for a desired target vehicle.

C. Algorithm Optimization (AO)

1) Adaptive Image Cropping (AIC): Given the hardware constraints, a method was required to increase the update frequency of the vision based range sensor. A required update frequency (capture and processing frame rate) of the distance sensor should be 10 Hz or more [26]. The combination HV/HG algorithm described so far ranged between 6 and 12 Hz [27], which was insufficient for the application of the ACC. To improve processing speed, an ROI was implemented using the last known location and size of the target. If the target is not detected within the ROI, the algorithm gradually increases the ROI until the target is detected again. After implementing this adaptive image cropping (AIC) process, the update frequency increased from 6 - 12Hz to 20 - 50Hz depending on the size of the ROI [27].

2) Tracking: Tracking is another AO method, and the algorithm implemented relies on the very basic method of tracking by detection. Once a candidate has been verified, a tracking process is created within a video sequence. Upon verification, the previously estimated vehicle's position and bounding box coordinates are stored and maintained between 5 to 10 consecutive iterations. If in the event of misdetection or occlusion, if the vehicle was not detected within the predefined number of iterations, the vehicle is declared as lost, after which the system alerts the driver that the target is lost.

Alternative tracking methods such as the Kalman filter, particle filter or other similar adaptive filters could be used to improve the algorithm's ability to update and predict the longitudinal position of the car [28], [29]. However, as the following sections show, the performance of the combined system proved largely sufficient.

3) Online Learning: Online learning of the targets is initiated once the verified targets are being tracked in the video sequence. If the detected target shows consistency in terms of image coordinates and bounding box size through a series of frames, then a template of the currently tracked vehicle is cropped and stored in the temporary template database. This database resets if the target is lost for a prolonged period of time. The temporary database overwrites itself once its buffer overflows to avoid scanning through too many templates, which could result in slower processing. Empirical tests show that using the most recent cropped image of the target vehicle results in the detector being more robust.

D. Range Estimation

An experiment was conducted for obtaining the relationship between the lane width size, at the point where the vehicle meets the road, to the longitudinal distance. An analytical solution involved using the number of pixels between the two lane lines along with the focal length f to obtain the angle that the lane width subtends. Then, as shown in Figure 2, the distance to the vehicle can be calculated as follows:

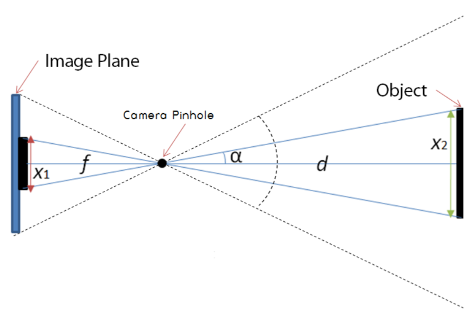

Figure 2. Distance estimation of the lane width, at the point where the vehicle meets the road, using trigonometry [27].

$$d = \frac{f x_2}{x_1}. \tag{2}$$

where:

- d is distance from camera to target
- x_1 is lane width size projected onto the camera CCD
- x_2 is approximate size of the lane width, typically between 3.4 to 3.7 meters on highways
- α is angle that the bounding box subtends
- f is the focal length of the camera

E. Range Smoothing

A Discrete Kalman Filter (DKF) was used to attenuate the jitter in range signal and to track the longitudinal position of the vehicle. Before the DKF was implemented, the hypothesis that the vision-based range signal is normally distributed was confirmed by Shapiro-Wilk test. The model derived by Shaik *et al.* is used for range estimation as in [30]. The state variables are the true range and speed of the vehicle over time. A white additive Gaussian random noise vector is assumed for covariance matrix [30].

IV. Vehicle Detection and Range Estimation Results

The performance of the detection algorithm is quantified by the following metrics [31]:

- True Positive Rate (TPR)
- False Detection Rate (FDR)
- Average False Positives per frame (FP/Frame)
- Average True Positives per frame (TP/Frame)
- False Positives per vehicle (FP/Object)

TPR is the percentage of the non-occluded vehicle in the cameras FOV that are detected. This quality measures recall and localisation. Since the vehicle detector only detects one target at a time, the TPR will either be 100 % or 0 %, depending on whether the same lane lead vehicle is detected or not. The ideal TPR should be 100 %. The TPR is calculated as follows:

$$\text{TPR} = \frac{\text{detected vehicles}}{\text{total number of vehicles}} \tag{3}$$

where total number of vehicles can either be 0 or 1. However, since dividing by 0 is undefined, the test will consist only of frames that contain a vehicle.

The FDR is the percentage of detections that were not true vehicles, and so the ideal FDR should be 0 %. This percentage of erroneous detection is a measure of precision and localisation, which is defined by

$$\text{FDR} = \frac{\text{false postives}}{\text{detected vehicles + false positives}}. \tag{4}$$

FP/Frame describes how susceptible a detector is to false positives and gives an informative measure of the credibility of the system. It measures localisation, scalability and robustness. The ideal FP/Frame is 0. It is determined by

$$\text{FP/Frame} = \frac{\text{false postives}}{\text{total number of frames processed}}. \tag{5}$$

Another measure of robustness is to use the FP/Object metric. This metric describes how many false positives are observed on average. The ideal FP/Object is 0, and the equation is

$$\text{FP/Object} = \frac{\text{false postives}}{\text{true vehicles}}. \tag{6}$$

The final metric is also a measure of robustness and describes how many true vehicles are recognised on average. The average TP/Frame is defined as

$$\text{TP/Frame} = \frac{\text{true postives}}{\text{total number of frames processed}}, \tag{7}$$

and the ideal TP/Frame is 1.

The overall performance of the system is based on the performance metrics mentioned above. It gives an informative assessment of the vehicle detection module's performance [31].

Nineteen datasets, which total approximately 23686 frames, were obtained using the Samsung Galaxy S video camera. The vehicle detection algorithm was executed on these datasets and compared to the ground truth data. The break-down of driving conditions for the 17 datasets are discussed in [27].

Table I
BREAK-DOWN OF DRIVING CONDITIONS FOR DATA SETS

Set	Glare	Distance	Traffic	Road	Cut In	Shadow
1	Sun	Medium	Low	Straight	No	No
2	Sun	Medium	Low	Straight	No	Yes
3	Sun	Close	Heavy	Varied	No	Yes
4	Sun	Varied	Varied	Varied	No	Yes
5	Glare	Varied	Varied	Straight	Yes	Yes
6	Sun	Medium	Low	Straight	No	Yes
7	Glare	Varied	Heavy	Curved	No	Yes
8	Sun	Varied	Low	Varied	No	Yes
9	Sun	Varied	Low	Straight	No	No
10	Sun	Medium	Low	Varied	Yes	No
11	Cloud	Medium	Low	Curve	Yes	No
12	Cloud	Medium	Varied	Straight	Yes	Yes
13	Cloud	Varied	Medium	Straight	No	No
14	Glare	Varied	Low	Straight	No	No
15	Sun	Varied	Low	Straight	Yes	Yes
16	Cloud	Medium	Low	Varied	No	No
17	Sun	Medium	Heavy	Straight	No	Yes

The performance metrics were calculated and tabulated in Table II.

Table II
VEHICLE DETECTION ACCURACY RESULTS SUMMARY

Dataset	TPR	FDR	FP/Frame	TP/Frame	FP/Object
1	100 %	0 %	0.00	1.00	0.00
2	100 %	0 %	0.00	1.00	0.00
3	100 %	0 %	0.00	1.00	0.00
4	99.5 %	5 %	0.05	0.95	0.05
5	98.5 %	33 %	0.28	0.57	0.48
6	99.9 %	0 %	0.00	1.00	0.00
7	90.5 %	14 %	0.13	0.78	0.16
8	97.3 %	72 %	0.23	0.09	2.55
9	97.1 %	3 %	0.03	0.95	0.03
10	99.2 %	2 %	0.02	0.97	0.02
11	99.7 %	17 %	0.15	0.72	0.21
12	99.8 %	1 %	0.01	0.99	0.01
13	99.2 %	2%	0.02	0.98	0.02
14	100 %	52 %	0.48	0.44	1.10
15	97.5 %	24%	0.23	0.72	0.32
16	99.9 %	0%	0.00	1.00	0.00
17	47.3 %	0%	0.00	0.47	0.00
18	97.0 %	0%	0.00	0.99	0.00
19	96.3 %	0%	0.00	0.96	0.00
Average	95.9 %	11.8%	0.08	0.82	0.26

A. Analysis Results and Discussion

The detector yielded a relatively high TPR since the detector only searches for one vehicle in the scene. A high TPR means that the detector has good recall and localisation. The FDR for most datasets were low, which means that the detector has good precision and localisation. Dataset 8 and 14 had the worst FDR, which could be as a result of the combination of curved road and shadows. The average FP/Frame for the datasets is low, which suggest robustness, localisation and scalability. Dataset 14 had the worse FP/Frame, which could be caused by glaring sunlight on the sensor. The closer the TP/Frame is to one, the better the robustness will be. Dataset 8 had the worst TP/Frame of 0.09, which could be a result of the curved roads as well because of poor lane markings. Finally the overall FP/Object was adequately low, but with datasets 8 and 14 being the worst performing datasets. The FP/Object in datasets 5, 11 and 15 were not as low as the other dataset, but still adequate.

Table III
DETECTION RATES IN VARIOUS DRIVING SCENARIOS

Scenario Description	Detection Rate
Low Traffic (Single Car)	80-95%
Medium Traffic	70-85%
Heavy Traffic	40-65%
Single Car, angled sun	81-96%
Single Car, overhead sun	80-95%
Single car, heavy glare	56-70%
Straight Road, low traffic	85-96%
Curved roads, low traffic	40-70%

Datasets 17 to 19 are standard (benchmark) datasets. They are also used by [31]. Dataset 1, used by Sivaraman *et al.*, is the same as dataset 17 and their dataset 3 is the same as dataset 18. The TPR in dataset 17 was very poor (47.3 %) due to heavy shadows and dense traffic conditions compared to the 95.0 % that was achieved by Sivaraman *et al.*'s Active-Learning-based Vehicle-Recognition and Tracking (ALVeRT) framework. The TP/Frame was also low at 0.47. The results may appear to be better than the results displayed in [31] in some cases. However, they implemented detection of multiple vehicles as opposed to the single vehicle detection in this case.

The vehicle detection algorithms were tested on various datasets. In the following analysis, data sets 1-17 were obtained from camera footage driving the routes. Sections of these datasets were then classified according to driving conditions, as detailed in Table I.

Using various scenarios indicative of each type of driving conditions, lead vehicles were manually identified in video segments. This ground truth was then compared with the pattern matching algorithm. Table III shows the preliminary results that were obtained using a set of 20 templates of the lower section of vehicles, which were extracted from training sets. These results utilize lane detection to define a ROI, but do not include the additional ROI restriction from tracking. Note that the detection rate is not good when there is heavy traffic, heavy glare, and curved roads. The inaccurate detection under heavy traffic conditions can be dramatically improved through the tracking algorithm described above, which would eliminate many detections of vehicles that are not the primary vehicle of interest. Curved roads produce poor results in large part because the currently implemented lane detection assumes straight lanes; this is an area for further improvement. And finally, glare is a problem for all vision-based systems, and would inevitably affect any vision-based ACC system (including a human driver).

B. Range Estimation Results

Table IV
CALIBRATED RESULTS FROM THE RANGE ESTIMATION ALGORITHM

Actual Distance with Laser (m)	Calibrated Distance from Image (m)	Error (metres)
3.80	3.811	-0.011
8.21	8.263	-0.053
13.19	13.162	0.028
16.12	16.225	-0.105
24.95	25.125	0.175
53.48	53.861	-0.175
72.08	72.039	0.0041
93.40	93.378	0.0022

Range estimation results (Table IV) show an errors lower than 0.2 meters, assuming negligible error in vehicle and lane estimation.

C. Kalman Filter Results

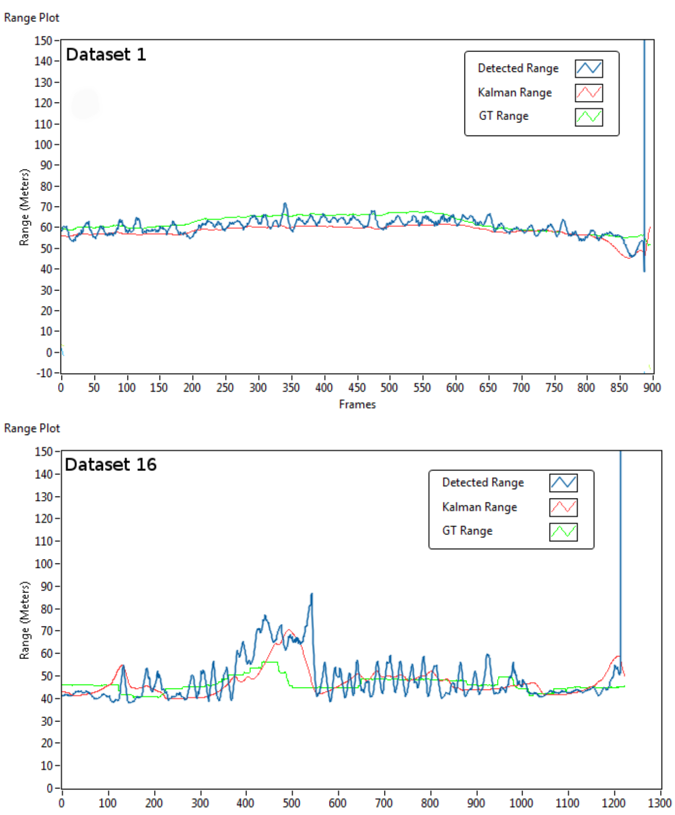

Figure 3. The original range results and the estimated range results using the Kalman filter when the lead vehicle is present in the scene. The GT range represents the ground truth range.

Figure 3 shows how the Kalman filter effectively smooths the noise and reduces the jitter of the signal when a vehicle is present in the image. The error spike in the original signal is due to the initialisation of the vehicle detection algorithm. The error covariance matrix R is the averaged variance of the last 100 points. The graphs also show how close the results are to the manually tagged ground truth range. The frame rate of the vision-based sensor ranged between 20 to 50 frames per second (fps) depending on the performance of the AIC algorithm, discussed in Section III.

These results demonstrate that the half-vehicle pattern matching and lane detection can successfully identify a lead vehicle and can reasonably estimate the headway gap under suitable conditions. However, the results also identify several situations in which the vision-based detection algorithm struggles. These particularly include situations with substantial glare from the road and other vehicles, as well as heavy traffic conditions and curved roads. While some variations on the algorithm can improve performance under some of these conditions, it is clear that any vision-based vehicle detection system will need further development to provide the robustness desired for an ACC system.

The current algorithm is not yet practical for common roadway traffic on roads worldwide. The system would therefore have to be more robust to cater for driving conditions influencing detection performance. Algorithms such as the one used by Khammari *et al.* can be used to provide good performance under complex urban environments with varying lighting conditions [8].

V. CONCLUSION

This study demonstrates that a monocular vision-based range finding system is feasible for adaptive cruise control systems, and that pattern matching is a reasonable method to explore for future system development. Though the pattern matching algorithm does not work as well in all driving scenarios, the nature of these scenarios makes them troublesome for most vision-based detection methods. This study also emphasises the utility of the three stages of the vehicle detection framework presented, which combined with a suitable range detection algorithm can be effective with a variety of underlying algorithms.

Future work can improve on multiple areas of this project. As with any such system, increased performance of the computing platform or of the nature of the algorithms would allow for more robust performance. Of particular interest for exploration are the incorporation of machine learning for visual object detection and more advanced tracking algorithms. Existing machine learning techniques, such as the Viola-Jones framework may prove ideal for fast and efficient vehicle detection. Complementing these detection methods, advanced tracking algorithms can assist in maintaining appropriate speed when detection of the appropriate target vehicle is made difficult by situational or environmental effects.

REFERENCES

[1] Z. Sun, G. Bebis, and R. Miller, “On-road vehicle detection: A review,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 5, pp. 694–711, 2006.

[2] N. Matthews, P. An, D. Charnley, and C. Harris, “Vehicle detection and recognition in greyscale imagery,” *Control Engineering Practice*, vol. 4, no. 4, pp. 473–479, 1996.

[3] T. Zielke, M. Brauckmann, and W. von Seelen, "Intensity and edge-based symmetry detection applied to car-following," in *Computer Vision — ECCV'92*, ser. Lecture Notes in Computer Science, G. Sandini, Ed. Springer Berlin Heidelberg, 1992, vol. 588, pp. 865–873. [Online]. Available: http://dx.doi.org/10.1007/3-540-55426-2_100

[4] T. Zielke, M. Brauckmann, and W. Vonseelen, "Intensity and edge-based symmetry detection with an application to car-following," *CVGIP: Image Understanding*, vol. 58, no. 2, pp. 177–190, 1993.

[5] Z. Sun, G. Bebis, and R. Miller, "On-road vehicle detection using Gabor filters and support vector machines," in *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, vol. 2. IEEE, 2002, pp. 1019–1022.

[6] S. Han, Y. Han, and H. Hahn, "Vehicle detection method using Haar-like feature on real time system," *World Academy of Science, Engineering and Techonology*, vol. 59, 2009.

[7] B. Duan, W. Liu, P. Fu, C. Yang, X. Wen, and H. Yuan, "Real-time on-road vehicle and motorcycle detection using a single camera," in *Industrial Technology, 2009. ICIT 2009. IEEE International Conference on*, Feb 2009, pp. 1–6.

[8] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle detection combining gradient analysis and adaboost classification," in *Intelligent Transportation Systems, 2005. Proceedings. 2005 IEEE*, Sept 2005, pp. 66–71.

[9] J. Kim, H. Lee, and J. Lee, "Improved Adaboost learning for vehicle detection," in *The 3rd International Conference on Machine Vision*, 2010, pp. 347–351.

[10] B. Stanciulescu, A. Breheret, and F. Moutarde, "Introducing new adaboost features for real-time vehicle detection," *arXiv preprint arXiv:0910.1293*, 2009.

[11] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 14, no. 4, pp. 1773–1795, 2013.

[12] Z. Chen, N. Pears, M. Freeman, and J. Austin, "Road vehicle classification using support vector machines," in *Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on*, vol. 4, Nov 2009, pp. 214–218.

[13] F. Han, Y. Shan, R. Cekander, H. S. Sawhney, and R. Kumar, "A two-stage approach to people and vehicle detection with hog-based svm," in *Performance Metrics for Intelligent Systems 2006 Workshop*, 2006, pp. 133–140.

[14] N. Khairdoost, S. A. Monadjemi, and K. Jamshidi, "Front and rear vehicle detection using hypothesis generation and verification," *Signal & Image Processing: An International Journal (SIPIJ) Vol*, vol. 4.

[15] M. Cheon, W. Lee, C. Yoon, and M. Park, "Vision-based vehicle detection system with consideration of the detecting location," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 13, no. 3, pp. 1243–1252, 2012.

[16] L. Mao, M. Xie, Y. Huang, and Y. Zhang, "Preceding vehicle detection using histograms of oriented gradients," in *Communications, Circuits and Systems (ICCCAS), 2010 International Conference on*, July 2010, pp. 354–358.

[17] C. Smith, C. Richards, S. Brandt, and N. Papanikolopoulos, "Visual tracking for intelligent vehicle-highway systems," *Vehicular Technology, IEEE Transactions on*, vol. 45, no. 4, pp. 744–759, Nov 1996.

[18] X. Mei and H. Ling, "Robust visual tracking and vehicle classification via sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 11, pp. 2259–2272, Nov 2011.

[19] M. A. Sotelo, D. Fernandez, J. E. Naranjo, C. Gonzalez, R. García, T. de Pedro, and J. Reviejo, "Vision-based adaptive cruise control for intelligent road vehicles," in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1. IEEE, 2004, pp. 64–69.

[20] M. Betke, E. Haritaoglu, and L. S. Davis, "Real-time multiple vehicle detection and tracking from a moving vehicle," *Machine Vision and Applications*, vol. 12, no. 2, pp. 69–83, 2000.

[21] G. P. Stein, O. Mano, and A. Shashua, "Vision-based ACC with a single camera: bounds on range and range rate accuracy," in *Intelligent vehicles symposium, 2003. Proceedings. IEEE*. IEEE, 2003, pp. 120–125.

[22] K. Nakamura, K. Ishigaki, T. Ogata, and S. Muramatsu, "Real-time monocular ranging by bayesian triangulation," in *Intelligent Vehicles Symposium (IV)*, June 2013, pp. 1368–1373.

[23] D. Nair, R. Rajagopal, and L. Wenzel, "Pattern matching based on a generalized fourier transform," *Advanced Signal Processing Algorithms, Architectures, and Implementations*, vol. 4116, 2000.

[24] R. K. Nath and S. K. Deb, "On road vehicle object detection and tracking using template," *Indian Journal of Computer Science and Engineering*, vol. 1, no. 2, pp. 98–107, 2010.

[25] Y. Ying and Z. Yuhui, "Technique of measuring leading vehicle distance based on digital image processing theory," in *Intelligent Computation Technology and Automation (ICICTA), 2010 International Conference on*, vol. 3. IEEE, 2010, pp. 674–677.

[26] R. Abou-Jaoude, "ACC radar sensor technology, test requirements, and test solutions," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 4, no. 3, pp. 1–122, 2003.

[27] R. Kanjee, A. K. Bachoo, and J. Carroll, "Vision-based adaptive cruise control using pattern matching," in *Robotics and Mechatronics Conference (RobMech), 2013 6th*. IEEE, 2013, pp. 93–98.

[28] N. Funk, "A study of the kalman filter applied to visual tracking," *University of Alberta, Project for CMPUT*, vol. 652, p. 6, 2003.

[29] K. Okuma, A. Taleghani, N. De Freitas, J. J. Little, and D. G. Lowe, "A boosted particle filter: Multitarget detection and tracking," in *Computer Vision-ECCV 2004*. Springer, 2004, pp. 28–39.

[30] M. Shaik, O. Das, L. Zhao, and Z. Liao, "Inter-vehicle range smoothing for NLOS condition in the persistence of gps outages," in *Industrial Electronics and Applications, 2009. ICIEA 2009. 4th IEEE Conference on*, May 2009, pp. 3904–3909.

[31] S. Sivaraman and M. M. Trivedi, "A general active-learning framework for on-road vehicle recognition and tracking," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 11, no. 2, pp. 267–276, 2010.