Vision-Based Adaptive Cruise Control Using Pattern Matching

Ritesh Kanjee
Electrical and Electronic Engineering
University of Johannesburg
reigngt09@gmail.com

Asheer K. Bachoo
Optronic Sensor Systems
Defence, Peace, Safety and Security
Council for Scientific and Industrial Research
abachoo@csir.co.za

Johnson Carroll
Electrical and Electronic Engineering
University of Johannesburg
jcarroll@uj.ac.za

***Abstract*—Adaptive Cruise Control (ACC) is a relatively new system designed to assist automobile drivers in maintaining a safe following distance. This paper proposes and validates a vision-based ACC system which uses a single camera to obtain the clearance distance between the preceding vehicle and the ACC vehicle. Pattern matching, with the aid of lane detection, is used for vehicle detection. The vehicle and range detection algorithms are validated using real-world data, and then the resulting system performance is shown to be sufficient using a simulation of a basic vehicle model.**

***Index Terms*—Vision-Based Adaptive Cruise Control, Adaptive Image Cropping, PI Controller, Autonomous Vehicle, Fiducial Detection, Pattern Matching, Single Camera.**

I. Introduction

Adaptive Cruise Control (ACC) is an automotive feature in which sensors and control systems help the driver maintain a safe driving speed and following distance relative to any other vehicle in front. ACC provides longitudinal distance control which does the following:

1) Find target vehicle.
2) Determine where the target vehicle is along the driving path.
3) Measure distance to the target vehicle.
4) Perform the necessary throttle or braking actuation to maintain a safe distance from the target vehicle.
5) If no target vehicle is present then the ACC vehicle should resume a preset cruising speed.

Previously implemented ACC systems have used various sensors to obtain distance data, including radar, laser radar (LADAR), and sonar range finding sensors [1]. For a general overview of vehicle detection methods, see [2]. Vision-based ACC uses one or more cameras with image processing to perform steps 1 to 3 described above.

In this paper, we propose an ACC system using pattern matching techniques to detect vehicles and calculate the headway distance. Section II discusses the image processing algorithms that are available for vehicle detection and range estimation. Experimental results of the implemented pattern matching algorithm are shown in Section III. Thereafter, Section IV briefly describes the vehicle dynamics and the controller design used to evaluate the system performance. Finally, in Section V, the results of the simulated system are presented and conclusions are drawn.

II. Image Processing

Although other studies have used stereo vision to determine vehicle range [3], this study restricts its attention to single-camera systems. Various image processing algorithms can be used for detecting vehicles as well as determining range with a single camera.

One initially appealing method is detecting a fiducial marker, which can be anything that is generic to most cars (e.g., number plate, back windscreen, or tail lights). Once detected, the size and orientation of the fiducial marker can be used to geometrically calculate the range. Fiducial markers provide a highly effective way of ensuring accurate target localization. However, it is difficult in practice to identify a generic fiducial marker that can be accurately detected at medium to far distances (larger than 30 meters). For example, number plates often come in different shapes and are too small to be accurately identified and measured by a camera with reasonable resolution.

As an indicative example, Sotelo et al. identifies possible vehicles by identifying horizontal edges within the detected lane which are presumed to correlate to the presence of a vehicle; the detection is then validated using colour symmetry properties of the object [4]. Though pleasingly simple, this method may not be robust enough to accurately detect vehicles in situations where other horizontal edges are present, such as sectioned pavement.

A. Pattern Matching

Pattern matching works by searching an image for a predefined template. Normalized cross-correlation is the most common way to find a template in an image [5], and is used for vehicle detection. A basic algorithm for detecting vehicle headway range involves having a set of templates of the target vehicle at various scales and orientations under various environmental conditions. Each frame in the video is compared with each template in the set; thus more templates result in longer processing and slows the detection rate. The processing time could be reduced by decreasing the Minimum Match Score (MMS), which is the minimum percentage match of the template for which a possible match can be positive (i.e. vehicle detection sensitivity). However, reducing the MMS increases the risk of false detections. The pattern matching detection algorithm returns the x, y coordinates of the vehicle within the image as well as the bounding box size enclosing the pattern. The bounding box is used as the Region Of Interest (ROI) for edge detection which determines edge point where the bottom of the vehicle meets the road. This point is mapped to the width of the lane, which is then used to approximate the longitudinal distance to the vehicle. The details are discussed in Section II-D.

For an ACC system, implementing vehicle pattern matching directly may not be a practical solution because it would require a massive database of templates consisting of various vehicle types. A simpler method is to implement pattern matching using the rear, bottom section of the vehicle (i.e. the bumper and wheels) as a template as opposed to the entire vehicle. As this portion of a vehicle is relatively consistent for various vehicle types, successful matching requires significantly fewer templates. Unlike the studies described in [2], this study utilizes a database of actual vehicle photos that are representative of the general vehicle population. This will allow for simple addition of templates, and paves the way for intelligent addition or removal of templates.

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 project, false positives are filtered using a combination of lane detection and tracking. These techniques are discussed in the following sections.

B. 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 exclude detected vehicles and non-vehicles appropriately. 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. The ROI is then clearly defined by the detected lane, as in Figure 1.

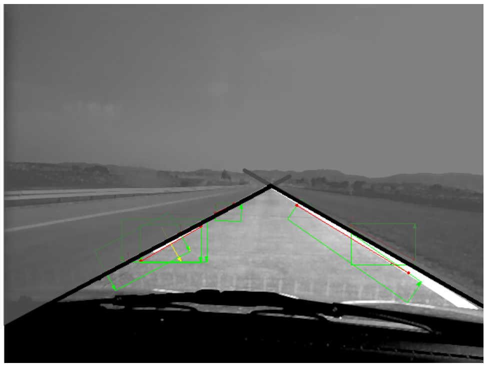

Fig. 1. Determining the region of interest using lane detection. The faint boxes indicate the pattern matches of the lane detection algorithm, and the solid black lines are the best linear fit for those matches.

C. Multiple Targets and Tracking

Restricting the pattern matching algorithm to the detected lane can significantly increase performance, and simultaneously eliminates many false detections and extraneous vehicles from consideration. When multiple vehicles are detected, the closest vehicle within the lane is selected. In the case where no lanes are found, the vehicle template that is closest to the center of the image and/or closest to the last detected target vehicle position is used.

In order to further increase detection speed and accuracy, an adaptive image cropping algorithm was developed to further restrict the ROI during reliable vehicle detection. When a vehicle is detected for several frames in succession, the algorithm restricts attention to the area immediately around the detected vehicle (see Figure 2). When the vehicle is no longer detected within that area, the search area progressively expands until the entire lane is again considered. In practice, this technique quadrupled the rate at which the system could process video frames, from 6-12Hz to 20-50Hz.

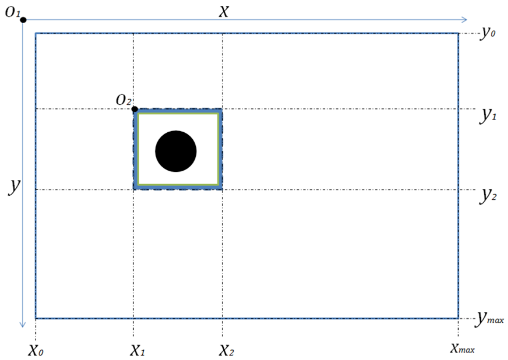

Fig. 2. Adaptive cropping uses previous vehicle location to restrict ROI.

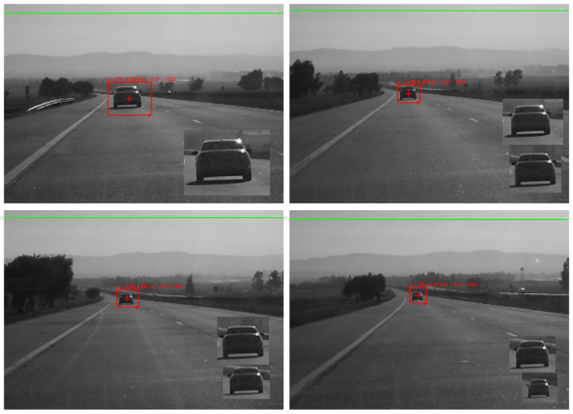

Fig. 3. Demonstration of how the lane width, at the point where the vehicle meets the road, is inversely proportional to the headway distance. This Figure also shows how multiple templates were detected.

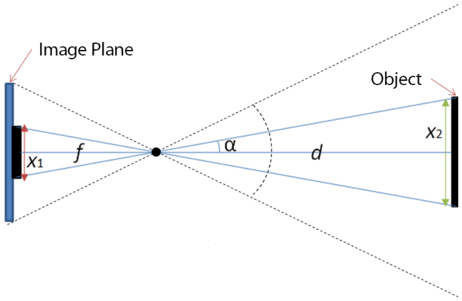

Fig. 4. Distance estimation of the lane width, at the point where the vehicle meets the road, using trigonometry

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 (see Figure 3). 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 4, the distance to the vehicle can be calculated as follows:

$$d = \frac{x_2}{2\tan(\arctan(\frac{x_1}{2f}))} \tag{1}$$

where:

- d is distance from camera to target
- x_1 is size of the template bounding box 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

Table I shows the calibrated range performance assuming that there is negligible error in the lane and vehicle detection algorithms.

TABLE I
CALIBRATED RANGE PERFORMANCE FROM THE RANGE ESTIMATION ALGORITHM.

Actual Distance with Laser (meters)	Calibrated Distance from Image (meters)	Error (meters)
3.85	3.811	-0.0055
8.194	8.263	-0.0408
13.2	13.162	-0.1596
16.1	16.225	-0.3474
24.9	25.125	0.0083
53.44	53.861	0.0552
72.04	72.039	0.0003
93.379	93.378	0.0003

TABLE II
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

III. VEHICLE DETECTION EXPERIMENTS

National Instruments (NI) LabVIEW was used to analyse video segments, extract distance information, and (as detailed in the following sections) calculate the appropriate control response [6]. A cellphone camera was used to record test data with a resolution of 700 x 480 pixels. A computer platform with a quad-core Intel Atom processor and 2GB of RAM, was used to implement the image processing algorithms while the NI CRIO 2093 embedded PC was utilized for real-time processing of the control systems and peripherals.

The vehicle detection algorithms were tested on various data sets. In the following analysis, data sets 1-17 were obtained from camera footage driving the routes shown in Figure 5. Sections of these data sets were then classified according to driving conditions, as detailed in Table II.

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

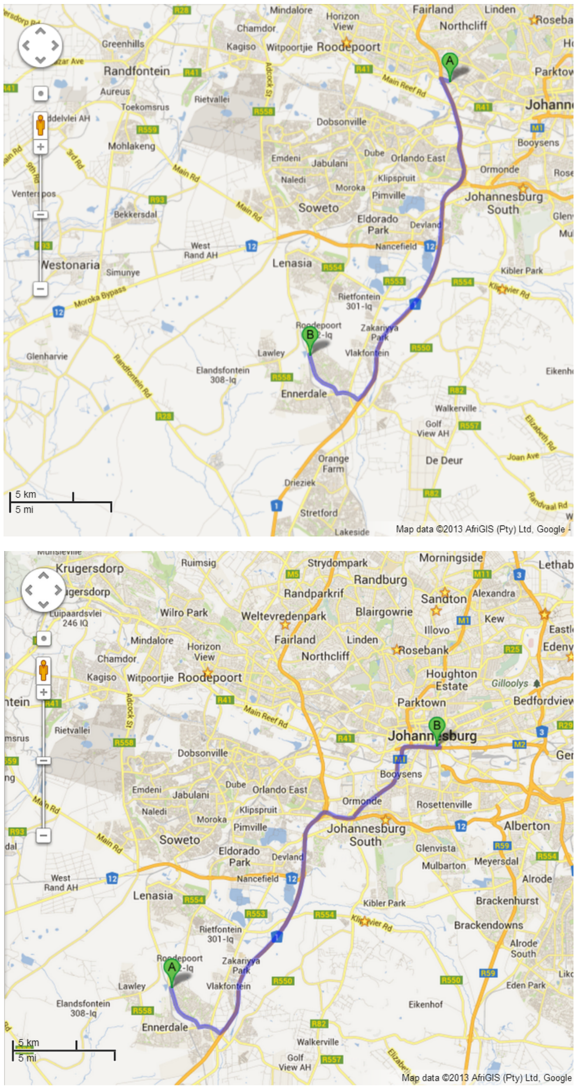

Fig. 5. The first map shows the 38.3 km route which was travelled from south to north. The second map is the 34.3 km route travelling from north to south. The dataset was gathered from the collection of videos captured on both routes. Images obtained using Google Maps.

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%

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).

TABLE IV
FRAME-BY-FRAME DETECTION ANALYSIS

	Car detected	No car detected
Car present	17557 frames (78.6 %)	1004 frames (4.5 %)
No car present	2015 frames (9 %)	1760 frames (7.9 %)

A more detailed indication of detection performance is given in the confusion matrix shown in Table IV, which shows the nature of the mis-detections for one video segment. This particular segment shows an accurate detection rate of just over 87%.

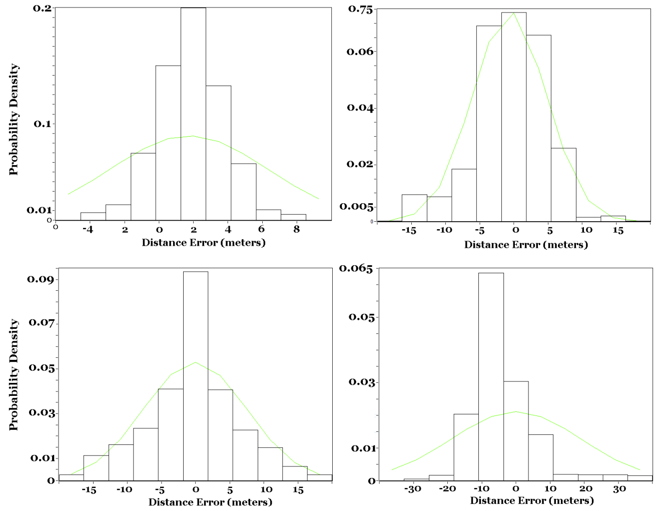

Fig. 6. Noise distribution in the detected range signals.

Figure 6 shows the distribution of errors in detected distances for several different data sets. As seen from the graphs, the error can sometimes be extreme and the variance can be dramatically different depending on road conditions. However, in the cases where the error is extreme the distance measurement is quickly fluctuating. A smoothing filter can reduce these fluctuations and produce more accurate results. In this case a Kalman Filter was used to significantly reduce the jitter of the range signal. Figure 6 also shows that the Kalman filter can, in fact be used in this situation, since the filter assumes that the noise is normally distributed.

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.

IV. SIMULATED CONTROL SYSTEMS

In order to validate the vehicle detection and range calculations described above, a simulated two-vehicle system was modelled and tested. All vehicle parameters were estimated based on actual physical characteristics and performance of the University of Johannesburg's hybrid electric vehicle.

A. Modelling the Vehicle

The vehicle model used is derived from the longitudinal dynamics of the vehicle, and incorporated appropriate delays and responses for throttle actuation and motor response, as shown in Equation 2 and Figure 7. As an example of this sort of derivation and the assumptions involved, see [7].

$$ma_x = F_{traction} - b_{drag}v^2 - C_{rr}\frac{mg\cos\theta}{r} - mg\sin\theta, \qquad (2)$$

where:

- $F_{traction}$ is traction force from the wheels applied to the road,
- m is the mass of the vehicle,
- a_x is the longitudinal acceleration,
- b_{drag} is the drag constant,
- v is velocity of the vehicle,
- C_{rr} is the coefficient of rolling friction,
- g is the gravitation acceleration,
- r is the radius of the tire,
- θ is the incline angle of the slope.

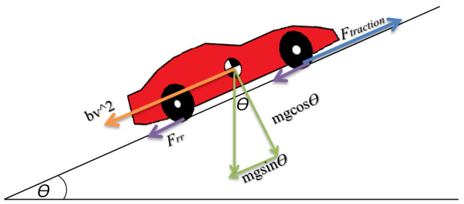

Fig. 7. Forces acting on a vehicle

B. Modelling the Power Train

As shown in Figure 8, the throttle input generated by the control system does not directly induce a traction force on the car tires. In an electric vehicle, the throttle position is processed by the engine controller and is mapped proportionally to the motor's angular velocity. For our simulation, this mapping was obtained from the manufacturer's datasheet of the electric motor controller being used in the hybrid car. The manufacturer also supplies a speed-torque curve. An approximate equation was obtained using a linear curve fitting of that curve between the starting torque and the breakdown torque of the motor. The torque and speed of the engine are scaled proportionally by the gear ratio and differential ratio resulting in the axle torque and speed and, thus, are transmitted to the tire of the car to produce $F_{traction}$ in (2).

A LabVIEW block diagram of the vehicle plant was implemented directly from these derivations, and the ACC detection system, including the controller, were tested against that model.

C. Adaptive Cruise Controller

Headway time is the time interval between the ACC vehicle and the target. The time-gap is related to the clearance and vehicle speed as shown in Equation 3 [8].

$$t_{gap} = \frac{d}{v_{ACC}} \qquad (3)$$

where:

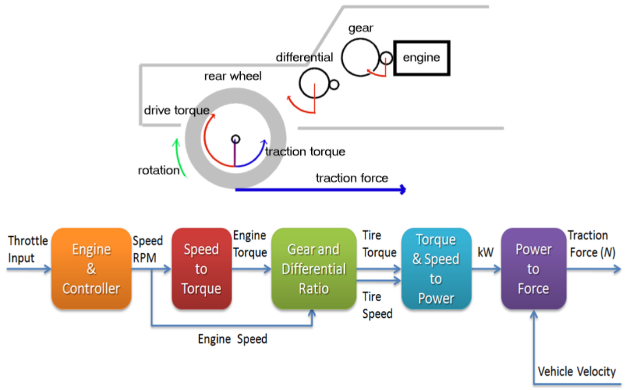

Fig. 8. Model of a power train and block diagram [6]

- d is the clearance distance between the ACC vehicle and the target vehicle
- v_{ACC} is the longitudinal velocity of the ACC vehicle
- t_{gap} is the headway time-gap

Using the range obtained from the vision-based sensor, the headway time gap can be calculated.

The time-gap variable is compared to a preset time-gap similar to the way velocity was used in the cruise control scenario. A small time gap gives the driver or system less time to react if the preceding vehicle has to slow down very rapidly, increasing the chance of an accident. There is a great deal of conflicting information regarding what an appropriate time-gap should be. Different authorities recommend or mandate following times of as little as 0.9 seconds [9] up to 7 or more seconds for some vehicles and conditions [10]. For ideal conditions, the 3 second rule has been recommended for ACC systems [9], and a 3 second time gap is used for this study. Once the appropriate headway time is chosen, a time-gap error is obtained by subtracting the actual time gap from the reference time gap. As shown in Figure 9, this error signal is fed into a Proportion Integral (PI) controller which controls the vehicle throttle signal.

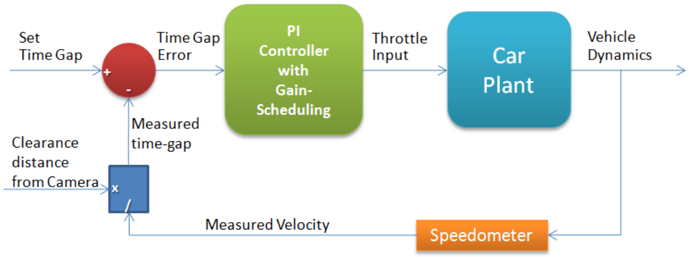

Fig. 9. Adaptive Cruise Controller Block Diagram

Near the correct headway time the PI response should be slow and smooth, but as the headway time deviates from the nominal value or region the controller must respond more dramatically. Gain scheduling is used in conjunction with this PI control because a single gain may not provide the necessary response in certain scenarios. The points at which the gains change must be based on the sensor accuracy, data rate, actuation delay, and capabilities of the chase vehicle itself. With a nominal time gap of 3 seconds, then the points are scheduled as follows:

- 0-1 seconds: emergency braking;
- 1-2 seconds: aggressive braking;
- 2-4 seconds: slower, smooth control actions;
- 4-6 seconds: more aggressive control; and
- 6-10 seconds: heavy acceleration up to speed limit.

For a time gap larger than 6 seconds, the ACC vehicle switches back into cruise control mode. If chosen appropriately, the chase vehicle will be able to consistently follow the lead vehicle through speed changes, but brake quickly enough to avoid collisions in an emergency stop [11]. However, the emergency brake region should also alert the driver to manually intervene for safety reasons.

The gains chosen here were selected based on acceptable qualitative performance using the approximate vehicle model described above. In actual implementation, the gains should be selected based on vehicle performance and driver comfort.

V. System Simulations

A simulation was used to test the cruise control algorithm. The PI gains were tweaked to obtain an approximate response in terms of rise time, overshoot and settling time. A generic design criteria was used for the response of the vehicle are as follows [12].

The angle of incline was varied and the actual velocity of the vehicle converged towards the reference velocity as expected.

The ACC control systems algorithm was tested within the simulation. The clearance distance was simulated by modelling two virtual cars and obtaining the distance between them. By manually controlling the throttle of the lead car, the longitudinal control algorithm was tested for various scenarios.

The first plot in Figure 10 shows the position with respect to time of both the ACC vehicle and target vehicle. In this experiment, the ACC vehicle is driving at high speed but approaches a target vehicle driving at a much lower, but constant speed. The lower plot shows the following time and the ACC vehicles response with the gain schedule PI-controller. As both plots indicate, the ACC car successfully slows, then re-establishes the correct following time.

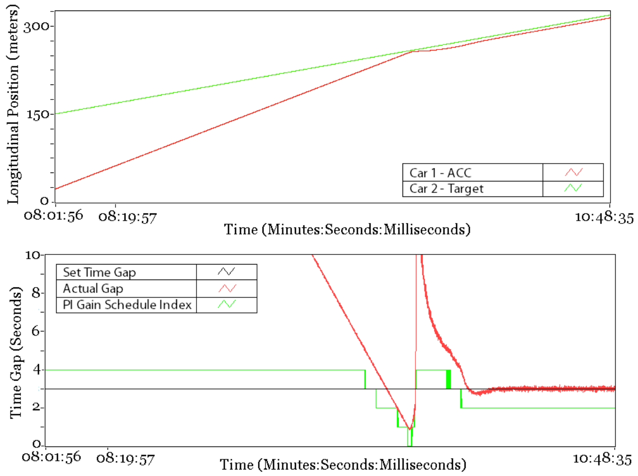

Fig. 10. Simulation results show that the ACC vehicle capably handles a slow moving target vehicle.

Again, the first plot in Figure 11 shows the position with respect to time of both the ACC vehicle and target vehicle. In this experiment, the ACC vehicle is driving at constant speed following a target vehicle when another target vehicle cuts between the two. As before, the lower plot shows the following time and the PI-controller gain schedule. As both plots indicate, the ACC car successfully slows, then re-establishes the correct following time behind the second target vehicle.

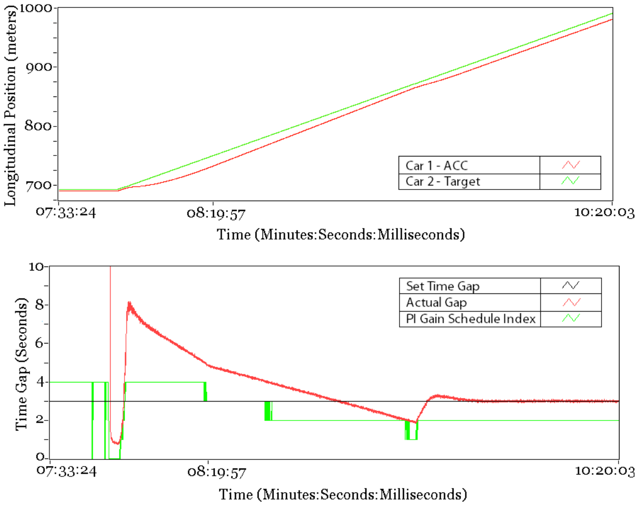

Fig. 11. Simulation results show that the ACC vehicle capably handles being cut off.

VI. Conclusion

This study demonstrates that a vision-based range finding system can be used 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.

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 [13] may prove ideal for fast and efficient vehicle detection. Complementing these detection methods, advanced tracking algorithms can assist in maintaining appropriate ACC behaviour when detection of the appropriate target vehicle is made difficult by situational or environmental effects.

References

[1] *ITS Handbook*. Tokyo: Highway Industry Development Organization, 2000, ch. 4. [Online]. Available: http://www.mlit.go.jp/road/ITS/2000HBook/chapter4/4-21e.html

[2] 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.

[3] M. Bertozzi, A. Broggi, A. Fascioli, and S. Nichele, “Stereo vision-based vehicle detection,” in *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*, 2000, pp. 39–44.

[4] M. Sotelo, D. Fernandez, J. Naranjo, C. Gonzalez, R. Garcia, 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, 2004, pp. 64–69 vol.1.

[5] 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.

[6] *IMAQ, Image Processing Manual*. Austin, Texas: National Instruments.

[7] M. Monster. (2003, November) Car physics for games. [Online]. Available: http://www.asawicki.info/Mirror/CarPhysicsforGames/CarPhysicsforGames.html

[8] MIT. (2005) Adaptive cruise control system overview. Fifth Meeting of the U.S. Software System Safety Working Group. [Online]. Available: http://sunnyday.mit.edu/safety-club/workshop5/Adaptive_Cruise_Control_Sys_Overview.pdf

[9] B. Filzek and B. Breuer. Distance behaviour on motorways with regards to active safety a comparison between adaptive-cruise-control (acc) and driver. [Online]. Available: http://www-nrd.nhtsa.dot.gov/pdf/nrd-01/esv/esv17/proceed/00066.pdf

[10] California commercial driver handbook. [Online]. Available: http://www.dmv.ca.gov/pubs/cdl_htm/sec2.htm

[11] J. Carroll, “Vehicle following with minimal memory,” in *AFRICON, 2011*, Sept. 2011, pp. 1 –5.

[12] B. Messner, R. Hill, and J. Taylor. (2012, October) Cruise control: Pid controller design. [Online]. Available: http://ctms.engin.umich.edu/CTMS/index.php?example=CruiseControl&section=ControlPID

[13] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, 2001, pp. I–511–I–518 vol.1.