Citation for published version:
Feng, Y, Pickering, S, Chappell, E, Iravani, P & Brace, C 2017, Distance Estimation by Fusing Radar and Monocular Camera with Kalman Filter. in SAE Intelligent and Connected Vehicles Symposium., 2017-01-1978, SAE International, SAE Intelligent and Connected Vehicle Symposium 2017, 26/09/17.

Publication date:
2017

Document Version
Peer reviewed version

Link to publication

Distance Estimation by Fusing Radar and Monocular Camera with Kalman Filter

Author, co-author (Do NOT enter this information. It will be pulled from participant tab in MyTechZone)

Affiliation (Do NOT enter this information. It will be pulled from participant tab in MyTechZone)

Abstract

The major contribution of this paper is to propose a low-cost accurate distance estimation approach. It can potentially be used in driver modelling, accident avoidance and autonomous driving. Based on MATLAB and Python, sensory data from a Continental radar and a monocular dashcam were fused using a Kalman filter. Both sensors were mounted on a Volkswagen Sharan, performing repeated driving on a same route. The established system consists of three components, radar data processing, camera data processing and data fusion using Kalman filter. For radar data processing, raw radar measurements were directly collected from a data logger and analyzed using a Python program. Valid data were extracted and time stamped for further use. Meanwhile, a Nextbase monocular dashcam was used to record corresponding traffic scenarios. In order to measure headway distance from these videos, object depicting the leading vehicle was first located in each frame. Afterwards, the corresponding vanishing point was also detected and used to automatically compute the camera posture, which is to minimize the interference caused by camera vibration. The headway distance can hence be obtained by assuming the leading and host vehicles were in the same ground plane. After both sensory data were obtained, they were synthesized and fused using Kalman filter, to generate a better estimation of headway distance. The performances of both sensors were assessed individually and the correlation between their measurements was evaluated by replotting radar measurements on the video stream. The results of individual sensors and Kalman filter were compared to investigate the optimization performance of the data fusion approach.

This is a general guidance of headway distance estimation with a low cost radar and a monocular camera. With described general procedures, this paper can allow researchers to easily fuse radar and camera measurements to obtain optimized headway distance estimation. This paper can facilitate the development of a more realistic robotic driver that can mimic human driver behaviors.

Introduction

While the influence of driving style on fuel consumption has been revealed by many studies [1]-[3], the debate about its exact impact never comes to a halt. The major reason of this phenomenon is the unrepeatability nature of human behaviors. Therefore, developing a driver model that can mimic human behaviors and perform various driving styles could be a potential solution to the controversy. As the first step of establishing such a driver model, sufficient driving related information need to be collected to maximize its similarity to human drivers. From drivers' perception, the most important driving information is commonly identified as vehicle state and traffic scenario. While vehicle state information, such as vehicle speed, engine rpm and gear selection, can be easily logged from vehicle's ECU, traffic scenario information can only be collected using external devices. In order to simplify the complexity, only the most prominent traffic scenario information is considered, which is the headway distance between the host and leading vehicle. While various sensors can be used to measure headway distance, they have corresponding benefits and drawbacks, as shown in Table 1.

Table 1. Comparison between sensors

Sensor	Benefits	Drawbacks
Lidar	Immune to light condition	Need to mount outside
	High accuracy	Expensive price
	Max range around 200 m	Sensitive to snow, fog, rain and dust.
Radar	Immune to weather and light condition	Difficult to distinguish targets
	Max range around 200m	Hard to detect pedestrians
	Reasonable price	Narrow field of view
Camera	Cheap	Sensitive to light
	Detect colour	Computational expensive
	Recognize signs	Low precision
Ultrasonic	Lowest price	Short measurement range

Although Lidar can obtain a better performance, its price is a major crisis in practical applications. Meanwhile, radar is a more affordable option which can also provide satisfying performance. It is more compact design and robust to various weather conditions. Some 77 GHz radars can measure up to 200 m, which is important in driver modelling, especially in highway scenarios. Meanwhile, radar has been widely adopted in the automobile industry, mainly for adaptive cruise control and collision avoidance. For example, Audi "Pre sense" system, BMW "Driving Assistant Plus" system and Mazda's "Smart Brake Support". Aside from these industrial applications, there have also been some studies using radar as a range finder, mainly in autonomous driving research. For instance, Wei et al. used six radars, together with other sensors, to develop an autonomous vehicle with minimal appearance modifications [4]. Moreover, Gӧhring et al. fused a long range TRW radar and several lidars to enable autonomous following behavior on highways [5].

Meanwhile, as an affordable external device, people's interest to purchase a dashcam is growing overwhelmingly. This is partially caused by drivers' increasing concern about traffic accidents, and their willingness to have a record of daily driving. While binocular cameras are more commonly used as range finder [6]-[8], some studies indicate that monocular cameras can also measure distance under some assumptions. For example, Stein et al. developed a single camera based ACC system that can estimate headway distance using road geometry and contact point of vehicle and road [9].

While the combination of radar and binocular camera is common in autonomous driving research, there have been very few studies that integrate long range radar with a monocular camera. Thus, this research is among the earliest to fuse long range radar and monocular camera using Kalman filter, and describe the corresponding procedures.

Methodology

The experiment equipment and general procedures are introduced in this chapter. The theoretical basis of related algorithms, such as the coordinate frame conversion between camera and radar, camera data analysis, radar data analysis and Kalman filter optimization, are included in this section.

The Equipment

The platform used in this project is a 2014 VW Sharan. It's driven repeatedly on a same route during daily driving for three consecutive months. The vehicle is equipped with an Influx Rebel CT OBD data logger to retrieve vehicle state information from ECU. Meanwhile, a long-range Continental ARS 308-2 77 GHz radar was mounted behind the bumper, which costs approximately £2,640. The raw radar measurements are also published on the data logger. Moreover, a £139 Nextbase 512G dashcam was mounted behind the windscreen to record forward traffic scenarios. This instrumented vehicle and corresponding sensors are illustrated in Figure 1.

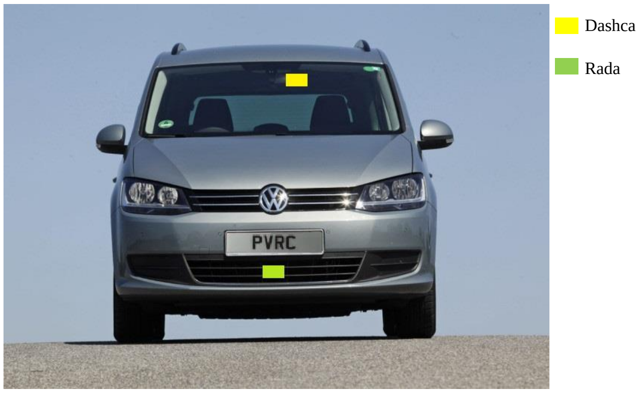

Figure 1. The instrumented vehicle

Coordinates Conversion between Camera and Radar

As indicated in Figure 1, there is a fixed mounting deviation between the radar and camera. Therefore, a transformation matrix needed to be established first to convert both sensory measurements into a mutual coordinate frame. The corresponding coordinate frames and mounting deviations were illustrated in Figure 2.

Thus, the transformation matrix T between the radar and camera can be derived as,

$$T_1 = \begin{bmatrix} -1 & 0 & 0 & 0.25 \\ 0 & -1 & 0 & -1.46 \\ 0 & 0 & 1 & 1.32 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

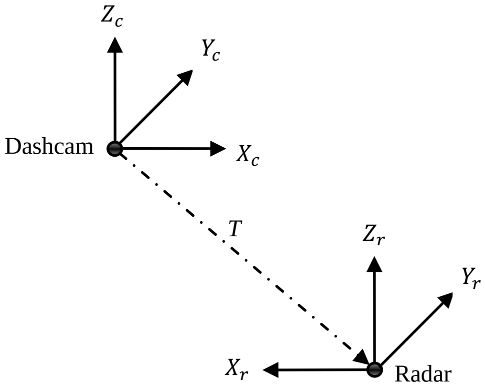

Figure 2. Coordinate frame conversion between radar and dashcam

Camera Calibration

In order to retrieve headway distance information from recorded dashcam footages, the transformation matrix between 3D world frame and 2D image frame needs to be established prior to other manipulations. This is also called inverse perspective mapping (IPM). In order to formulate this coordinate conversion, an intermediate 3D camera coordinate frame needs to be introduced. The relations between three coordinate frames are illustrated in Figure 3. The parameters for conversion between camera and image coordinate frames are intrinsic parameters, such as focal lengths and optical centers. Meanwhile, the transformation between the world and camera coordinate frames is defined by extrinsic parameters, such as the camera's height, pitch and yaw angles.

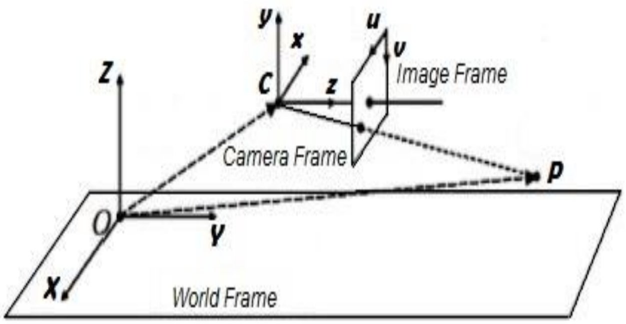

Figure 3. Relation between three coordinate frames

As the intrinsic parameters of each camera are fixed after manufacture, they should be investigated as the first step of image processing. As indicated by Zhang [10], the most popular approach to obtain these parameters is Camera Calibration. Owing to the distinct pattern of chessboards, they are commonly used to calibrate cameras. Various photos of the chessboard were taken from different angles. As the exact dimensions of the chessboard were already known, the intrinsic parameters of the camera can hence be easily obtained using MATLAB Single Camera Calibration App. The obtained intrinsic parameters were exported for following applications.

Vanishing Point Detection

After obtaining the intrinsic parameters through camera calibration, the next step is to measure the extrinsic parameters. While the camera's height can be measured easily, its pitch and yaw angles are not fixed, and can vary with the movement of the vehicle, which are mainly caused by road gradient and vibration. Therefore, these two extrinsic parameters need to be measured automatically at each time step to maximize the accuracy of distance estimation.

As the camera's pitch and yaw angles have direct influence on coordinates of the vanishing point, this relation can be used to compute both angles. Owing to perspective effect, parallel lines will converge at the vanishing point in each image. Thus, these lines need to be identified in the image to locate vanishing point coordinates. Lane markers are hence selected as they are prominent features on structured roads.

While there have been some applications that use either Hough transform or curve fitting to identify lane markers, it was found that these methods tend to be computational expensive, especially for long and high resolution videos. Thus, a novel and effective lane marker detection approach was created.

Each video frame is first converted to grayscale, and vertically divided into 48 sections to reduce computation complexity. As lane markers are white, they will have larger intensity than adjacent road surface, and hence larger values in grayscale. Therefore, there will be pulses in grayscale values, which can indicate the horizontal locations of both edges of lane markers. Thus, this pulse feature in grayscale can be used to locate lane markers. Meanwhile, in order to exclude some false positives, which can be pulses caused by shadows or other interferences, a grayscale threshold and a width threshold need to be assigned. This is because owing to the distinct feature of lane markers and road surface, the pulses representing lane markers will have relatively large amplitude and short wavelength. As light conditions have a vast influence on grayscale values, the grayscale threshold is hence needed to be adaptive to improve its robustness. Thus, Otsu's method is used to compute this threshold by minimizing the intraclass variance of black and white pixels. Moreover, in order to distinguish lane markers from other white objects, the width feature is adopted for the isolation. This is because lane markers are much thinner than other white objects. Thus, in each row of grayscale data, the pulses denoting white objects are located, and their wavelengths are measured. Afterwards, a predefined width interval is used to distinguish lane markers.

With pixels denoting lane markers detected, they are hence classified into corresponding groups. Each group contains all pixels for a same lane marker. Afterwards, polynomial fitting is used to generate the desired parallel lines. The intersections of these lines are also calculated, and the clustering approach proposed by Rodriguez and Laio [11] is performed to classify all intersections. The members within each cluster are accumulated, and the center of the largest cluster is regarded as the vanishing point.

Transformation Matrix Formation

After the vanishing point is located, the transformation matrix between image frame and world frame can hence be calculated accordingly. Prior to forming the matrix, the remaining two extrinsic parameters need to be computed using the vanishing point. As illustrated in Figure 4, the equations for pitch and yaw angles can hence be derived as [12]

$$\theta = arctan\left[tan\partial_v\left(1-\frac{2y}{N}\right)\right] \tag{1}$$

$$\gamma = arctan\left[tan\,\partial_h\left(\frac{2x}{M}-1\right)\right] \tag{2}$$

where θ is pitch angle; γ is yaw angle; ∂_v and ∂_h are camera vertical and horizontal aperture; x and y are vanishing point coordinates; M and N are image length and width in pixels.

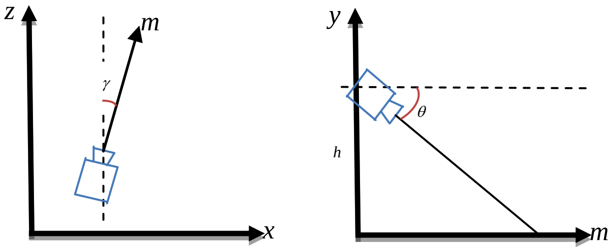

Figure 4. Camera pitch and yaw angle

With all camera parameters computed, the transformation matrix between pixel coordinates and world coordinates can hence be obtained following the relations illustrated in Figure 3.

$$T_2 = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix} \tag{3}$$

$$\begin{aligned}
&A_{11} = -h\times c2/f_x, A_{21} = h\times s2/f_x, A_{31} = A_{41} = 0,\\
&A_{12} = h\times s1\times s2/f_y, A_{22} = h\times s1\times c2/f_y,\\
&A_{32} = h\times c1/f_y, A_{42} = -c1/f_y,\\
&A_{13} = h\times c2\times p_x/f_x - h\times s1\times s2\times p_y/f_y - h\times c1\times s2,\\
&A_{23} = -h\times s2\times p_x/f_x - h\times s1\times c2\times p_y/f_y - h\times c1\times c2,\\
&A_{33} = -h\times c1\times p_y/f_y + h\times s1, A_{43} = c1\times p_y/f_y - s1.
\end{aligned}$$

where h is the camera height; f_x and f_y are camera focal lengths; p_x and p_y are camera optical center coordinates; s1 and c1 are sine and cosine values of pitch angle; s2 and c2 are sine and cosine values of yaw angle.

Vehicle Detection

With the transformation matrix formulated, the next procedure is to detect the vehicle in each image and obtain its pixel coordinates. As taillights of the leading vehicle are red and close to each other in a horizontal plane, this distinct feature can be used to detect the leading vehicle. Moreover, as shown in Figure 5, a trapezoid region in front of the host vehicle is selected as the search area to eliminate noises caused by sign plates and reduce computation complexity.

After searching in the region of interest, potential red candidates are indexed and sorted. Each pair of red objects that satisfy the vertical and horizontal tolerance are grouped and the closest pair are regarded as taillights of the leading vehicle. The centroids of this pair are extracted and averaged to represent the pixel coordinates of the leading vehicle.

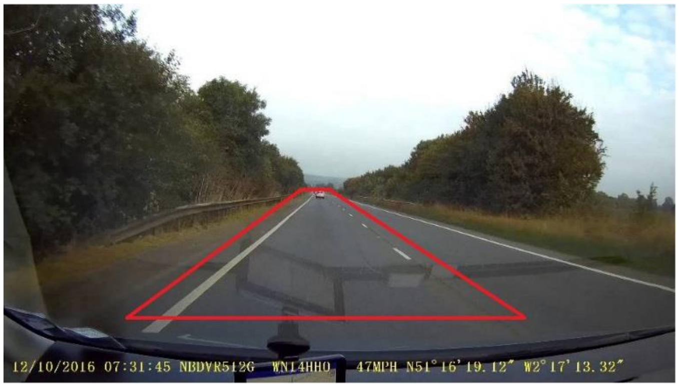
12/10/2016 07:31:45 NBDVR512G WN14HHO 47MPH N51°16'19.12" W2°17'13.32"

Figure 5. Leading vehicle detection zone

Headway Distance Computation

Assuming the transformation matrix is T and vehicle pixel coordinates are (p_u , p_v), the distance between the leading vehicle and host vehicle can hence be easily obtained. Firstly, rearrange the vehicle coordinates as homogenous coordinates(P_u , P_v , 1) ^{T},and multiply it with the transformation matrix T, can result in,

$$\begin{bmatrix} W_x \\ W_y \\ W_z \\ W \end{bmatrix} = T_2 \times \begin{bmatrix} P_u \\ P_v \\ 1 \end{bmatrix} \quad (4)$$

Therefore, the coordinates of the detected vehicle in the world frame are (W_x/W, W_y/W, W_z/W). According to the world frame shown in Figure 3, the headway distance can hence be derived as W_y/W.

Radar Data Analysis

Aside from the camera footages, the raw radar measurements are published on the Rebel data logger through a medium speed CAN bus, with a transmission rate of 500 KBits/s. While this Continental radar offers object tracking and target tracking output modes, the former is preferred as it can track the same objects in continuous measure cycles. In this mode, radar can measure up to 40 different objects in each cycle. It possesses an updating frequency of 1000 Hz, and publishes an incorporated CAN message at each time step. This generated hexadecimal CAN message contains multiple useful information, such as longitudinal distance, lateral distance, relative speed and acceleration between detected objects and host vehicle.

After downloading the radar measurements from the data logger, these CAN messages need to be decoded to facilitate further application. Therefore, a GUI based Python program was created. This program was compiled and can directly load raw radar files, detect each message type, analyze the message according to corresponding structure, plot radar data and output results as csv files.

As the radar can measure multiple objects in each cycle, post processing is hence implemented to extract useful data from the decoded files. In order to isolate data representing the leading vehicle from all radar measurements, camera footages are hence used for the detection and localization of the leading vehicle. Thus, the leading vehicle is first identified within each video frame. Afterwards, its pixel coordinates are used to multiply with the transformation matrix between the pixel and world frame to compute its world coordinates within the camera frame. The position of the leading vehicle within the radar frame can hence be obtained by multiplying the computed result with the transformation matrix between radar and camera frame. Afterwards, a small search zone is created around this position to locate corresponding radar measurements. Meanwhile, as the original frame rate of the dashcam is 30 Hz, which is much slower than the radar updating frequency, the same search zone is used until the next video frame is updated. Moreover, as the leading vehicle position should be rather consistent between every two adjacent frames, its consistency is also checked to exclude abrupt changes caused by interference.

Kalman Filter

Kalman filter is adopted to fuse measurements from both sensors to compensate the drawbacks of each individual sensor and obtain an optimized result. A typical Kalman filter contains two phases, predict and update. At each time step, a priori state and covariance estimates are first predicated using previous values and the system model. Afterwards, these priori estimates are updated according to sensory measurements. An optimized posteriori estimate of the state vector can hence be obtained through this process. The predict phase is defined as,

$$x_{k|k-1} = F_k x_{k-1|k-1} + B_k u_k \quad (5)$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \quad (6)$$

Meanwhile, equations for update phase are,

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (7)$$

$$x_{k|k} = x_{k|k-1} + K_k (z_k - H_k x_{k|k-1}) \quad (8)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (9)$$

where F_k is the state transition model, B_k is the control-input model, w_k ~ N(0 , Q_k) is the process noise, v_k ~ N(0 , R_k) is the observation noise, P_k is the error covariance matrix, H_k is the observation model.

Meanwhile, as the camera can only measure headway distance, and radar measurement contains both distance and relative speed, the state vector of the proposed filter is hence defined as $x_k = [dis_k , vel_k]^T$, and the observation vectors of camera and radar are defined as $z_k = [dis\_camera_k]$, and $z_k = [dis\_radar_k , vel\_radar_k]^T$ respectively, The state vector and the corresponding error covariance matrixs of the filter is updated when receiving new sensory measurements.

Results

The results are presented in four sections, discussing the performance of each individual sensor, the correlation between them, and optimization with Kalman filter respectively.

Radar

An experiment was implemented to assess the performance and accuracy of the radar. In this scenario, the host vehicle remained static and distances to all targets were manually measured for comparison. The performance of detecting multiple random objects was examined to justify the radar. The test scenario and detected objects were illustrated in Figure 6.

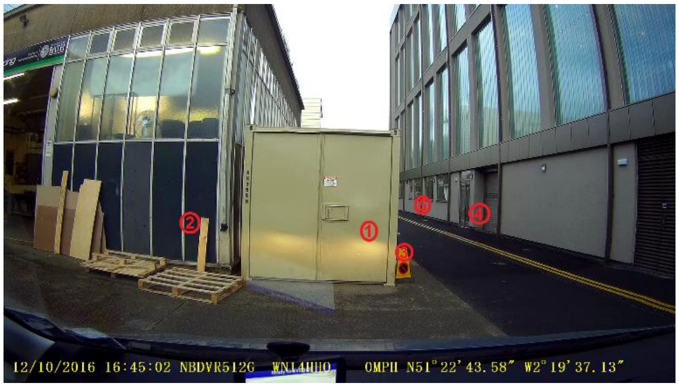
②
①
⑤
④
③
12/10/2016 16:45:02 NBDVR512G WN14HHO OMPH N51°22'43.58" W2°19'37.13"

Figure 6. Radar validation

Five objects were detected by the radar in this scenario. The obtained radar measurements were listed in Table 2.

Table 2. Radar measurements

Object ID	1	2	3	4	5
Lateral Offset (m)	0	2	-1	-4	-4
Longitudinal Distance (m)	5	5	5	18	22
Measured Distance (m)	4.95	4.95	5.38	17.88	22.32

It can be noted that the radar measurements tend to be rounded to integers. While the longitudinal distances between the radar and manual measurements were not exactly the same, the largest difference was 0.38m. This variance was satisfying as human driver's estimation about the headway distance will be less accurate than this radar. Thus, the radar data can be regarded as absolute measurements and used to improve accuracy of Kalman filter.

Camera

While the radar measurements can be directly retrieved, the distance from dashcam requires more computations. Therefore, the validation of camera measurements is divided into four steps, road recognition, vanishing point detection, vehicle detection, and distance computation.

Road Recognition

As the basis of distance estimation with a dashcam, the lane marker detection is crucial to the accuracy of the algorithm. Therefore, the performance of this proposed detection approach needs to be fully assessed. As light condition has a significant influence on the image's grayscale values, various scenarios should be examined to validate the proposed algorithm. Therefore, a straight segment of A36 between Standerwick and Woolverton was selected, and images along the route were taken at different time and dates to maximize the likelihoods of light conditions.

As shown in Figure 7, six images of various time and weather conditions are recorded. Figure 7.a, 7.b and 7.c illustrate normal weather in the morning, afternoon and evening, while 7.d, 7.e, 7.f show foggy, sunny and rainy respectively. It can be noted that only two lane markers lighted by headlights were detected at evening, and all three lane markers were detected in other figures. Meanwhile, although the differences between most figures are not prominent to visualize, the grayscale values vary among them. As listed in Table 3, the largest variance is 30.92% between morning and evening. As these figures generally contain all possible weather conditions, it can be noted that the proposed algorithm can detect parallel lane markers accurately in most scenarios. Meanwhile, it is interesting to find that the average grayscale values of different weathers almost remain the same. It indicates that weather condition has a much less influence on average grayscale than presumed. Therefore, this proposed lane marker detection based on Otsu's method is robust to different weather conditions.

Table 3. Average grayscale values

Figure	Morning	Afternoon	Evening	Foggy	Sunny	Rainy
Average Grayscale	161	130	82	158	149	149

Vanishing Point Detection

After the performance of road recognition algorithm was validated, the next procedure was to locate the vanishing point based on detected parallel lines. As described in the previous section, k-means is used to classify the intersections of these lines, and the center of the largest cluster is regarded as the vanishing point. In order to validate this hypothesis, the same straight road was used for examination. This is because the vanishing point position should be rather consistent along a straight road. The total covered distance was 600m, and travel time was 57s. Meanwhile, the frame processing frequency was set to 2 Hz, leading to a sum of 114 frames. Therefore, a total of 114 vanishing points were located. In order to visualize the result, a sample frame was first used to demonstrate the k-means clustering technique.

As shown in Figure 8, all three lane markers are detected in this frame, and vanishing point is denoted by the green point. As six lines were detected, a sum of 15 intersections was obtained. Meanwhile, it can be noted from Figure 9 that these intersections were divided into three clusters. While cluster 2 and cluster 3 mainly contained outliners, most intersections were in cluster 1. Therefore, as described in the previous section, the center of cluster 1 was regarded as the vanishing point of this frame. As the geometry of the frame was nearly symmetry, the vanishing point should be roughly near the center of the image. Therefore, the detected vanishing point position was reasonable. Moreover, as the vanishing point position should be rather consistent along the straight route, the positions in 114 frames were compared to further validate the proposed algorithm.

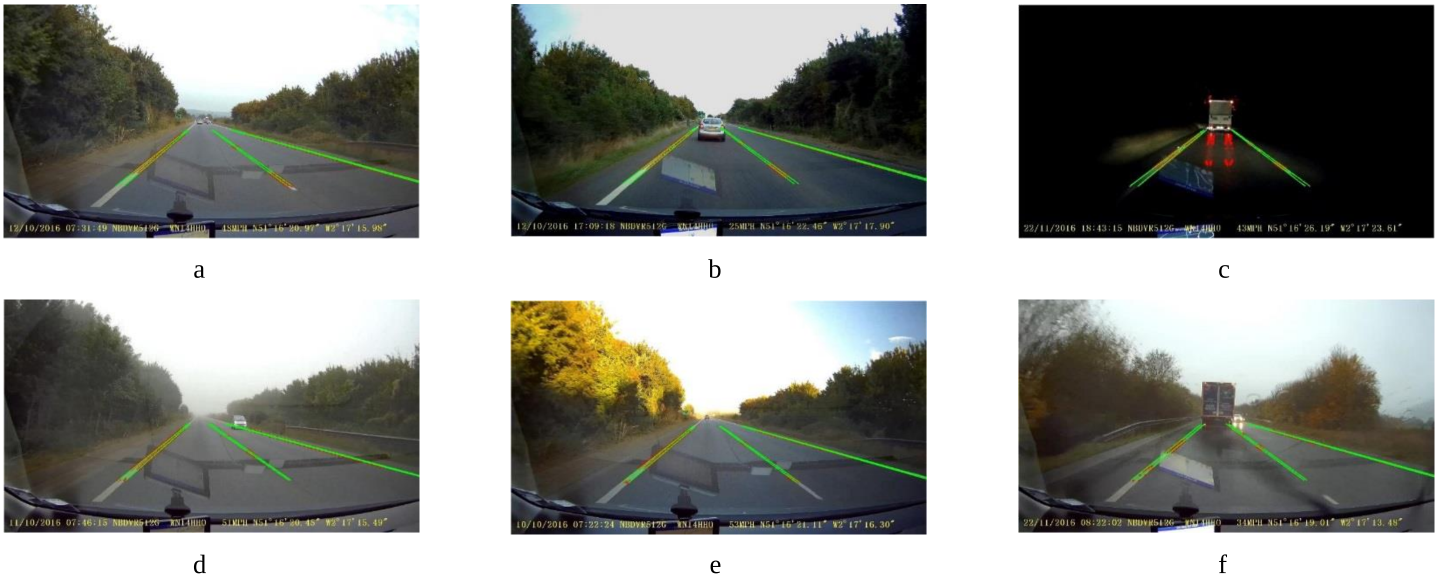

Figure 7. Lane marker recognition

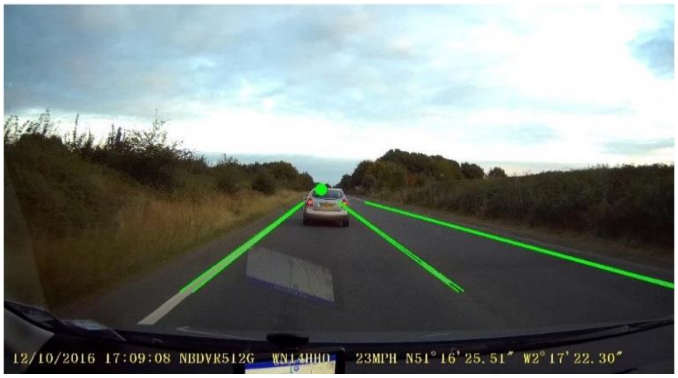
12/10/2016 17:09:08 NBDVR512G WN14HHO 23MPH N51°16'25.51" W2°17'22.30"

Figure 8. Lane markers for vanishing point detection

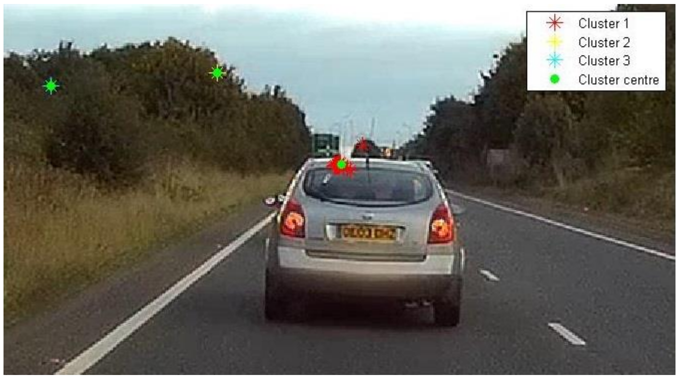

Figure 9. Vanishing point detection in one frame

As illustrated in Figure 10, red points are the recorded vanishing points of 114 continuous frames, and the green point denotes the cluster center of all vanishing points. Meanwhile, the intra cluster distance was computed. The accumulated distance to the cluster center was 831.6615, and the average distance was 7.3. This variation was acceptable considering the dimension of the image (1920 × 1080). Thus, it can be noted that the proposed vanishing point detection algorithm is qualified for further application.

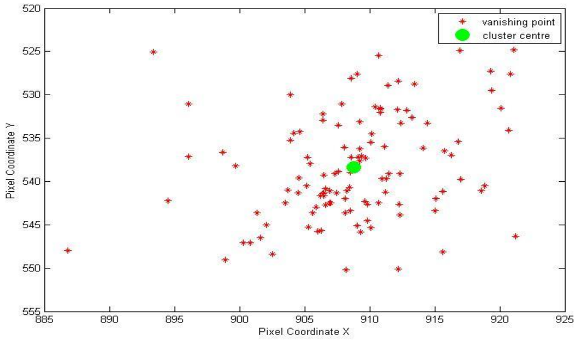

Figure 10. Accumulated vanishing point positions

Vehicle Detection

With the vanishing point located in each frame, the transformation matrix between the 3D world frame and 2D image frame can be easily computed at each time step. Thus, the next procedure is to detect the leading vehicle in each frame. As vehicle type and light condition may potentially influence the detection of taillights, these two factors were examined to justify the performance of the adopted algorithm.

As shown in Figure 11 and 12, two vehicle types under three weather conditions are used to validate the algorithm. The images were extracted from daily driving data. The two vehicle types were chosen as sedan and lorry, as they are the most common vehicles on road. Meanwhile, normal afternoon, foggy and rainy were adopted to investigate the influence of light condition. It was found that the proposed algorithm can sufficiently detect taillights of the leading vehicle in various weather conditions. Its robustness to vehicle type and light condition can hence be validated.

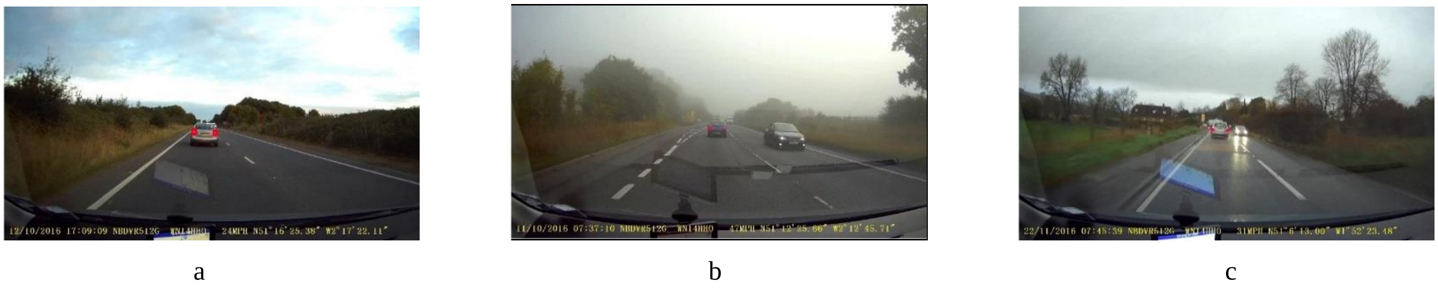

Figure 11. Sedan detection in normal afternoon, foggy and rainy

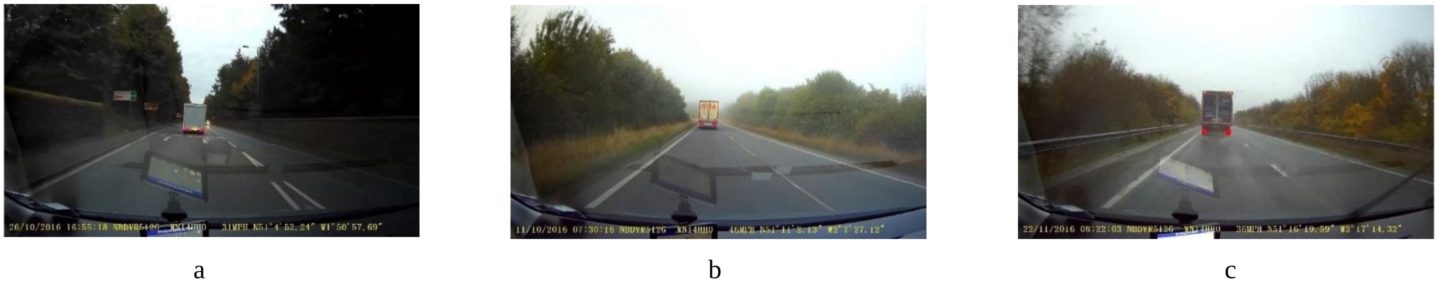

Figure 12. Lorry detection in normal afternoon, foggy and rain

Distance Computation

Similar to the validation of radar measurements, the accuracy of computed headway distance from dashcam needs to be examined. As estimating distance using dashcam requires a leading vehicle and lane markers, manually measure the gap distance was hence not safe and realistic. Therefore, the approaching to congestion scenario was used as an alternative option. With the leading vehicle stopped and host vehicle approaching, dashcam measurements were processed at its original frame rate, 30 Hz. Meanwhile, the host vehicle speed profile during this period was also extracted. As the sampling rate of vehicle speed was 2 Hz, the time gaps between these samples and the complete stop were calculated. Afterwards, the headway distances at each sampling time were computed using the time gaps and the mean value of adjacent vehicle speed samples.

While this measurement was less accurate, it was sufficient for this application as dashcam data were used as relative measurements, mainly to record the changing tendency of headway distance. Therefore, this scenario was located from the video footage and corresponding data were extracted. In this scenario, the approaching to static period of the host vehicle lasted for six seconds after the leading vehicle was completely stopped. The obtained result was shown in Figure 13.

It can be noted that although the headway distance measurements from dashcam and accumulated vehicle travel showed some deviance, the decreasing tendency matched quite well. Thus, the proposed headway estimation using dashcam is feasible, as it can be used in Kalman filter to improve continuity. Moreover, the deviance issue can also be mediated using radar measurements.

Meanwhile, the processing speed of the proposed algorithm was also evaluated. It was found that the average processing time for each frame was 0.667s during the post-processing phase. As the entire video was loaded at once for processing, which could occupy some computing memories, this processing speed was rather satisfying. With the ultimate aim of this study to be post-processing camera and radar data to achieve sensor fusion, the real-time in-vehicle performance of this algorithm has hence not been evaluated. However, it should be noted that the proposed algorithm can potentially be faster in real-time scenarios.

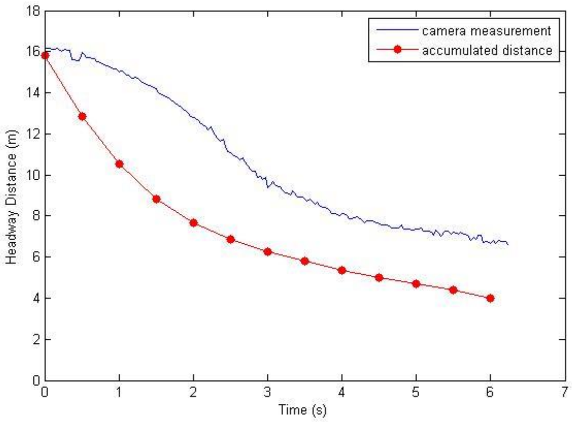

Figure 13. Dashcam validation

Correlation between Radar and Camera

With the radar and camera separately examined, the correlation between radar and camera needs to be assessed. This is mainly because radar measurements contain multiple objects, and the desire target is the leading vehicle. While it is difficult to directly identify the leading vehicle from radar plot, camera data can be used to isolate the same object in radar measurements. The leading vehicle position can be first computed from camera footage, and a search zone can hence be created around this detected position to find the corresponding object in radar measurements. Therefore, the correlation between radar and camera data needs to be examined. This can be achieved by plotting radar measurements on corresponding camera footage. A corresponding MATLAB function was created, and a captured frame was shown in Figure 14.

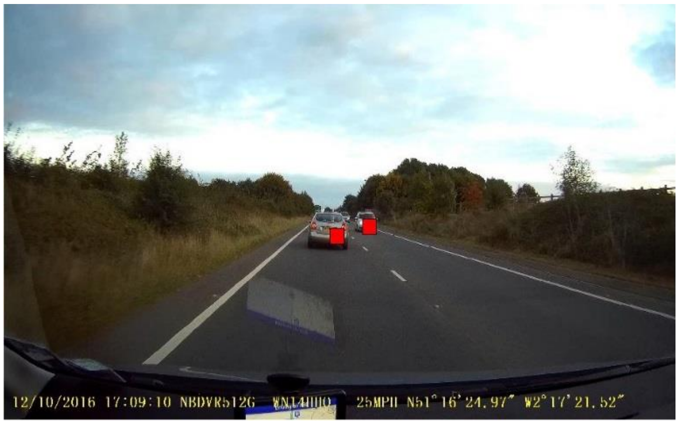
12/10/2016 17:09:10 NBDVR512G WN14HHO 25MPH N51°16'24.97" W2°17'21.52"

Figure 14. Plotting radar objects on video footage

The two red squares denoted radar measurements at the same time step, (0 , 15) and (3 , 17) respectively. While the correlation on the oncoming vehicle was good, there was a deviance on the leading vehicle. The pixel coordinate of the converted radar plot was (958 , 601), and the detected leading vehicle in this frame was (927 , 580), leading to a pixel distance of 37.44. As this deviance was relatively small, it can be easily compensated by setting a larger search zone. Therefore, it can be noted that the correlation between radar and camera is satisfying.

Kalman Filter Optimization

After examining the correlation between radar and camera, the final procedure is to apply Kalman filter to optimize the distance measurement. The purpose of this optimization is to compensate the drawbacks of each individual sensor. For example, while the radar data is more accurate, it can lose targets in some measurement cycles. Meanwhile, camera measurements are more continuous with reduced accuracy. Therefore, Kalman filter is adopted to fuse both sensory measurements and generate optimized distance estimations.

In order to examine the performance of the established Kalman filter, the radar data and dashcam footage of a 350s journey were isolated from recorded files. This extracted trip covers a total distance of approximately 4000m, and contains three closing-in events. Thus, it could be a suitable evaluation of the filter. The separate sensory measurements and fused estimations were illustrated in Figure 15.

It can be found from Figure 15 that the correlations between each sensor and the Kalman filter are satisfying, as the shape of each plot is similar. Moreover, the radar measurements are smoother and more stable, which proves its accuracy. Thus, the selection of radar as an absolute measurement was justified. Meanwhile, while the continuity of camera data is commendable, it shows intense oscillations and multiple outliners. Moreover, its accuracy also deteriorates in short range. From extracted data, it was found that the average deviation from camera to radar data was 1.35m for distance larger than 10.5m, and 3.92m for short range. While both sensors have corresponding drawbacks, the performance of the established Kalman filter is promising. It can be noted that the filter's estimations show an improved continuity and reduced oscillations compared with each individual sensor. When radar went offline, the filter used its prediction model and camera measurements to improve continuity and reduce oscillations. Meanwhile, radar data were used to improve estimation accuracy when received.

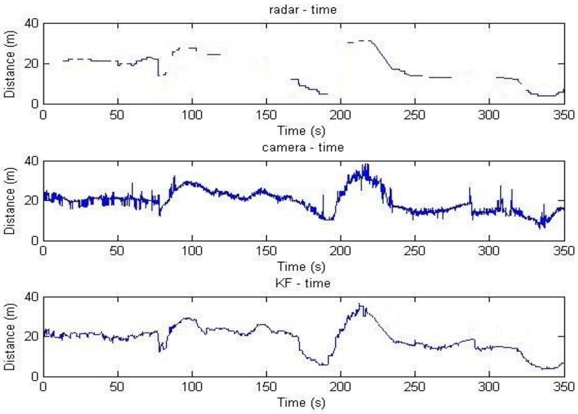

Figure 15. Kalman filter optimization

Conclusion

The primary aim of estimating distance by fusing radar and monocular camera was achieved. The transformation matrix between camera and radar was first computed to facilitate the conversion of sensory measurements. Afterwards, the approach for computing headway distance from monocular camera footage was developed.

The established approach consists of two steps. First is to formulate the transformation matrix between image frame and world frame, and second is to detect the leading vehicle in each image. For the matrix formulation, while the fixed intrinsic parameters of the adopted dashcam were easily obtained through camera calibration, the extrinsic parameters were automatically computed at each time step to exclude the interference caused by vehicle vibration and road gradient. In order to facilitate this computation, a novel and effective lane marker detection algorithm was first created. The vanishing point of each image can hence be located by clustering the intersections of all lane markers. Afterwards, the pitch and yaw angles of the dashcam were computed using the vanishing point position. The desired transformation matrix can hence be formulated with both intrinsic and extrinsic parameters. Meanwhile, the taillight feature was adopted to detect the leading vehicle. With the transformation matrix and pixel coordinates of the leading vehicle, the headway distance can hence be computed from the recorded video footages.

Aside from the camera data analysis, a Python based analyzing tool was developed to decode the radar measurements. Afterwards, in order to locate object depicting the leading vehicle in radar measurements, a search zone was created using the position of the detected leading vehicle in camera data and the transformation matrix between both sensors. The consistency of radar measurements is also checked to exclude abrupt changes caused by interference.

Afterwards, accuracies of both sensors were first separately examined, and the influences of light condition and vehicle type on camera measurements were investigated. The correlations between both sensors were also examined, and Kalman filter was adopted to optimize distance estimations. It was found that the established system showed a strong robustness to various scenarios and the optimizing performance of Kalman filter was satisfying. Thus, the established system is sufficient for further applications.

References

[1] Tzirakis, E., and Zannikos, F., “Impact of driving styles on fuel consumption and exhaust emissions: Defensive and aggressive driving style,” *10th International Conference on Environmental Science and Techonology*, 2007.

[2] Alessandrini, A., Cattivera, A., Filipp, F., and Ortenzi, F., “Driving style influence on car CO_2 emissions,” *Proc.7 – “Emission Inventories – Meeting the Challenges Posed by Emerging Global, National, and Regional and Local Air Quality Issues”*, 2012.

[3] Ho, S. H., Wong, Y. D., and Chang, V. W. C., “What can eco-driving do for sustainable road transport? Perspectives from a city (Singapore) eco-driving programme,” *Sustainable Cities and Society*, 2015, vol. 14, pp. 82-88.

[4] Wei, J. Q., Snider, J. M., Kim, J. S., Dolan, J. M., Rajkumar, R., and Litkouhi, B., “Towards a viable autonomous driving research platform,” *IEEE Intelligent Vehicles Symposium*, 2013.

[5] Göhring, D., Wang, M., Schnürmacher, M., and Ganjineh, T., “Radar/Lidar sensor fusion for car-following on highways,” *5th International Conference on Automation, Robotics and Applications*, 2011.

[6] Hou, A. L., Chen, J., Jin, Z. J., Liao, Q., and Geng, Y., “Binocular vision measurement of distance based on vehicle logo location,” *Applied Mechanics and Materials*, 2012, vol. 229-231, pp. 1154-1157.

[7] Long, H. M., Guo, H. Y., Liang, F., and Liu, G. H., “Distance measurement algorithm based on binocular stereo vision,” *Applied Mechanics and Materials*, 2014, vol. 635-637, pp. 948-952.

[8] Patel, D. K., Bachani, P. A., and Shah, N. R. , “Distance measurement system using binocular stereo vision approach,” *International Journal of Engineering Research & Technology*, 2013.

[9] Stein, G., Mano, O., and Shashua, A., “Vision-based ACC with a single camera: Bounds on range and range rate accuracy,” *IEEE Intelligent Vehicles Symposium*, 2003.

[10] Zhang, Z., “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000, vol. 22, no. 11, pp. 1330-1334.

[11] Rodriguez, A., and Laio, A., “Clustering by fast search and find of density peads,” *Science*, 2014, vol. 344, pp. 1492-1496.

[12] Nieto, M., Salgado, L., Jaureguizar, F., and Cabrera, J., “Stabilization of Inverse Perspective Mapping Images based on Robust Vanishing Point Estimation,” *Proceedings of the 2007 IEEE Intelligent Vehicles Symposium*, 2007.

Contact Information

Yuxiang (Felix) Feng,

Work phone: (+44) 7463 359554

yuxiang.feng@bath.edu

Definitions/Abbreviations

VW	Volkswagen
ECU	Engine Control Unit
OBD	On-board diagonstics
IPM	Inverse Perspective Mapping