Noname manuscript No.
(will be inserted by the editor)

Detecting the relative position of vehicles on two-lane highways in a smartphone-based video overtaking aid application

Subhadeep Patra · David Van Hamme · Peter Veelaert · Carlos T. Calafate · Juan-Carlos Cano · Pietro Manzoni

Received: date / Accepted: date

Abstract Here we present a smartphone-based real-time video overtaking aid for vehicular networks. The developed application aims to prevent head-on collisions that might occur due to attempts to overtake when the view of the driver is obstructed by the presence of a larger vehicle ahead. In such cases, the driver does not have a clear view of the road ahead and of any vehicles ahead that might be coming from the opposite direction, resulting in a high probability of accident occurrence. Our application relies on the use of dashboard mounted smartphones with the back camera facing the windshield and the screen towards the driver. A video is streamed from the vehicle ahead to the vehicle behind automatically, where it is displayed so that the driver can decide if it is safe to overtake. One of the major challenges is the way to elect the video source and destination among the vehicles in close proximity, depending on their relative position on road. For this purpose, we have focused on two different methods: the first relying solely on GPS data, which is faster but prone to errors due to GPS inaccuracies; the second method involving use of the camera and vehicle heading information, is accurate over short distances but is more computationally intensive.

S. Patra
Instituto de Biomecánica de Valencia,
Universitat Politècnica de València, Edificio 9C,
Camí de Vera, s/n, 46022 Valencia, Spain.
E-mail: subpat@doctor.upv.es

D. V. Hamme, P. Veelaert
TELIN, Image Processing and Interpretation,
Ghent University - iMinds, Gent, Belgium.

C. T. Calafate, J-C Cano, P. Manzoni
Department of Computer Engineering,
Universitat Politècnica de València,
Camí de Vera, s/n, 46022 Valencia, Spain.

Keywords VANET · ITS · Image processing · plate recognition · smartphone application · Android

1 Introduction

Overtaking accidents are considered by many sources [2] as one of the main reasons behind injuries and loss of lives. This is due to availability of scarce opportunities to practise overtaking during standard driving lesson. Groeger and Clegg [8] analysed 550 hours of lessons and arrived to the conclusion that practising overtaking only formed 5% of the total duration. It is thus no surprise that errors may be committed by both inexperienced and experienced drivers alike. University of Nottingham also identified overtaking manoeuvres as a critical source of accidents, and studied them in detail. The findings of the University of Nottingham can be accessed at [5]. Thus we decided to build an application that could assist drivers while overtaking without any user intervention. The system autonomously creates a network among the close-by vehicles, and provides drivers with a real-time video feed from the one located just ahead, making use of a dashboard mount smartphone for this purpose. Smartphones were chosen as a platform for development aiming to make our solution cost-effective and easy to adopt. Also, the evolution of smartphones towards high performance terminals with multi-core microprocessors packed with a large number of onboard sensors, justifies our interest in integrating smartphones to vehicular networks for the development of ITS applications.

The developed overtaking aid tries to prevent head-on collisions, and is specially useful in scenarios involving two-lane highways/ single carriageways with one lane per direction of traffic flow, and no central reservation to separate them. Also, dangerous situations which the application aims to prevent rarely occur on dual carriageways, or single carriageways with more than one lane per direction of traffic flow. The application provides drivers with a video stream from the vehicle just in front, travelling in the same direction. This way, the application provides an enhanced multimedia aid to the drivers based on which they might decide whether to overtake. This is especially useful in scenarios where the view of the driver is blocked by a larger vehicle, or when a long queue of cars is located ahead and the driver wishes to overtake. To eliminate any confusion regarding the source of the video stream, and have a clear idea of the distance that needs to be covered to overtake, streaming is only allowed between the vehicle just in-front and the vehicle following it. An added advantage of using this type of communication is the absence of multi-hop delay, resulting in improved performance.

Recent works on the design of ITS safety applications for smartphones are mostly aimed at monitoring of driver behaviour, road conditions, and pedestrian activity to avoid accidents. Examples of applications that studied driver behaviour include [11], [20], [10], [21], [7], and [22]. The application [11] uses unsupervised learning to create a driver profile which can be used later to provide recommendations and incentives to encourage the adoption of better driving style. BackPocketDriver [20] is a similar application that confirms the accuracy of onboard sensors for monitoring driver behaviour. While [10] compares the performance various algorithms to identify aggressive driving behaviour. In [21], the authors were able to recognise distracted driving from smartphone sensor data. [7], on the other hand, detects driver drowsiness based on image processing. Finally, [22] is an interesting work where the authors have developed an early recognition system that is able to determine inattentive driving event before their completion from audio signals. Another important factor that can affect passenger safety is road condition, which was studied extensively by El-Wakeel et al. [6], Kataoka et al. [9], and AbdulQawy et al. [1]. These works use crowdsensing to acquire data which is later classified, this information can be later used to generate alerts when a vehicle approaches an affected area. The third category of safety applications that rely on studying pedestrian activity includes [4] and [16]. Zadeh et al. [4] in their work, takes in to account location and age information of drivers and pedestrians, which is uploaded to a server making use of cellular network. The server processes the information, and taking into account other factors like weather conditions and time of the day, generates a warning message. [16], on the other hand, aims to minimise the use of smartphone by pedestrians while walking. It takes into account the angle of the head of the pedestrian and the way the smartphone is being held to detect dangerous situations.

2 Application overview

Our application enables drivers to receive realtime video stream of the view similar to what is observed by the driver of the vehicle just ahead. It runs on the Android platform, and requires devices to be equipped with GPS, wifi, and a back camera. For our application to work satisfactorily, each vehicle need a dashboard mounted Android device running our application. The devices are positioned such that their back camera faces the windshield, so that it has a clear view of the road in front, and of the cars coming from the opposite direction. The back camera is used to record a video which is transmitted in real-time over the vehicular network to the vehicle located just behind where it is displayed.

Fig. 1 is a photograph taken from within the vehicle behind, during one of our real experiments with the application. It depicts the setup necessary for using our application. We can find a windshield mounted Android tablet running our application which is displaying the video stream received from the white vehicle ahead. Similarly, the vehicle ahead also has a windshield mounted device that is taking advantage of the back camera to record the video and stream it to the vehicle behind. If we look closely at the figure, we can see that video displayed on the receiver device shows the vehicle that is coming from the opposite direction. The driver can take this into account before deciding to overtake. Note that here in this case one device is the source of the video stream while the other is the receiver. But situations may occur where the same device receives a video stream from the vehicle in front, while at the same time recording a different video using its back camera and streaming it to the vehicle behind. Now, for streaming the video a communication between the devices running our application is needed. While cellular network may be used for this purpose but this may involve network usage charges, also cellular coverage may not be available everywhere. Also, android devices have an onboard wifi for communication but software limitations only allows devices to connect to wifi access points, not directly with other devices near it. Thus, we use a communication box called GRCBox [18] to create the required vehicular network for the exchange of data. Each vehicle is equipped with a GRCBox that allows adhoc communication between devices. Devices within a vehicle connects to the GRCBox within in, and the GRCBoxes are able to communicate

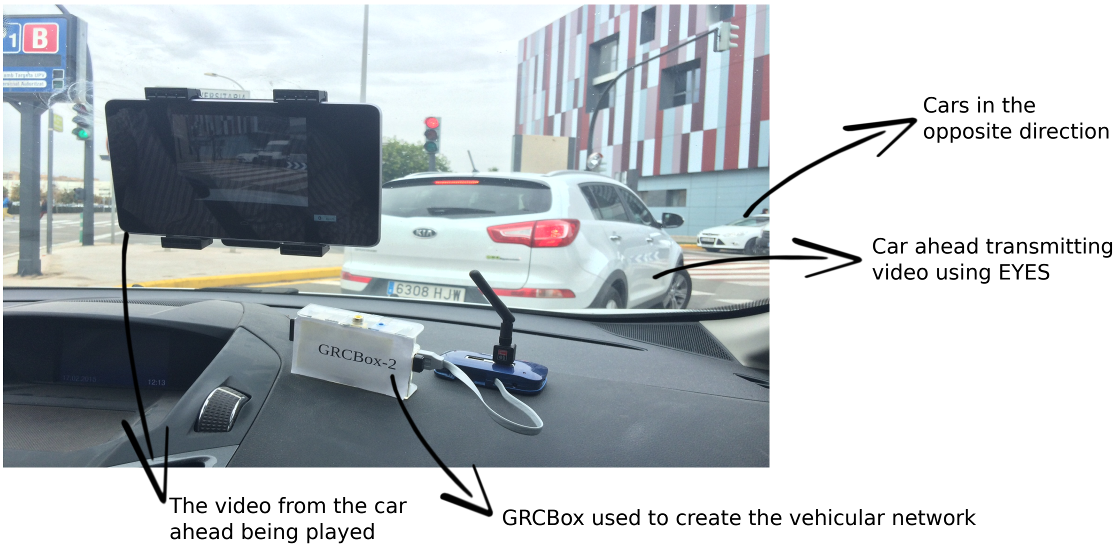

Fig. 1: Required setup while using our overtaking aid application.

with other GRCBoxes present in one-hop neighbouring vehicles, thus acting as a router and enabling our application to exchange data.

It is to be noted only video information is streamed from the vehicle ahead to the vehicle behind in our application, no audio is used. For encoding the video we use the MJPEG compression scheme, which involves compressing the video stream separately as JPEG [19] images. The default video settings used in the application offers a MJPEG video stream of HD resolution at 10 FPS with JPEG quality set to 50 percent for the video stream. The JPEG quality parameter supported by Android OS that may range from 1 to 100. The value of 1 results in highly compressed images with lowest perceivable quality, while the reverse occurs for the value of 100. The default settings was chosen owing to its satisfactory performance in terms of delay (below 500 ms), while meeting the throughput limitations of the GRCBoxes, on which our application depends. More details regarding the experiments performed to establish the chosen settings can be accessed at [12].

3 The critical question

The working of our application can be explained in three easy steps. In *step one* vehicles exchange advertisement messages to publish the availability of the video. Based on the information contained in the advertisement messages, the sender and the receiver vehicle of the video stream is selected. The *second step* involves the transmission of the video being captured by the sender, and its display at the receiver end. In this step, even though the video transmission in occurring, advertisement messages are continuously exchanged in the background to keep analysing whether video transmission is necessary. In the *third step*, which is also the last one, video transmission and playback are stopped when situations arise where video transmission between the send and the receiver is no longer necessary. An example of such situation is a successful overtake by the vehicle which was previously following the video sender.

Since we have already selected the encoding scheme and related settings to be used for the video stream, and configured the network for the exchanging data, we are left to define ways to choose which vehicle is to stream the video and which one should receive it. Before diving into this matter, let us go through some of the assumptions that our application makes. Since our application is most effective in two-lane highways with one lane per direction of traffic flow, we will only consider such scenarios. Next we assume that vehicles will be travelling along the center of the lane, and overtaking manoeuvres would be taking place along straight road segments since overtaking at curves is dangerous. Finally, we consider that only a video stream from a vehicle travelling ahead of the receiver and moving in the same direction is relevant. Keeping these assumptions in mind we take a look at two different approaches to select the video source and destination, one based on just GPS information, and the other based on the fusion of license plate recognition data with vehicle heading information.

3.1 Location-based approach

In this approach each vehicle broadcasts its current and previous location as an advertisement message. These advertisement messages are processed by the receiver vehicle and a request for video stream is forwarded if the advertisement message is relevant.

3.1.1 Same lane test

We have already established that the source and receiver of the video has be travelling on the same lane thus first take a look at how to detect such situations.

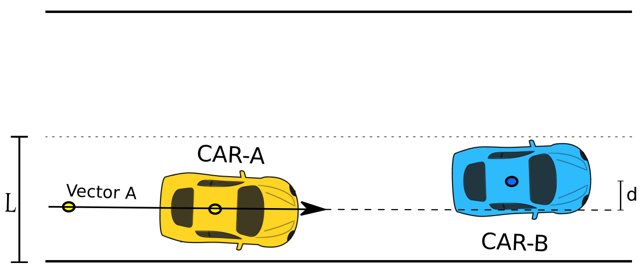

Fig. 2: Concept of the Same Lane Test.

Fig. 2 explains what we are trying to achieve. The figure shows CAR-A following CAR-B and both of them are on the same lane. To define the test to decide whether CAR-B is actually on the same as CAR-A, we first construct the driving vector of CAR-A from its current and previous known location. We assume that this Vector A coincides with the centerline of the lane of width L. Next we measure the perpendicular distance d between the current location of CAR-B and the Vector A. If d is less than equal to L/2, then CAR-B is located on the same lane.

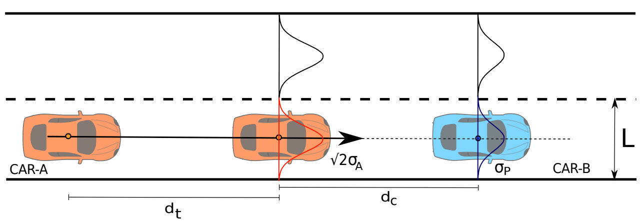

Fig. 3: Theoretical analysis of the concept.

Let us consult fig. 3 for the theoretical interpretation of the concept of the same lane test. We assume that the immediate error in the current and previous location of the vehicle behind (CAR-A) is characterised by a normal distribution with standard deviation σ_A since the two locations are close to one another. Thus, the uncertainty on the direction of its driving vector is given by $arctan(\sqrt{2}\sigma_A/d_t)$ where d_t is the distance between the two GPS observations. Now since both the vehicles are travelling along the center of the lane, the current location of CAR-B should coincide with the driving vector of CAR-A. The uncertainty distribution of the lateral position of the centerline of the lane at a distance d_c in front of CAR-A has standard deviation $\sigma_P = \sqrt{2}\sigma_A(d_c + d_t)/d_t$. And the True Positive Rate (TP) of the *Same Lane Test* is calculated as follows:

$$\begin{aligned} Accuracy = TP &= \frac{1}{2} + \frac{1}{2} erf(\frac{d_t L}{4(d_c + d_t)\sigma_A}) \\ &= \frac{1}{2} + \frac{1}{2} erf(\frac{L}{4(\frac{d_c}{d_t} + 1)\sigma_A}) \end{aligned} \quad (1)$$

Equation (1) shows that the accuracy is dependent on the term d_t/d_c, which is the ratio of the length of the driving vector of the vehicle behind and the distance between the two vehicles.

Tendency of σ_A	Value
Number of obs.	1284
Minimum	3
Maximum	12
Mean	7.24
Std. error	0.07
Std. deviation	2.42
95% C.I.	0.13

Table 1: Summary of error in the GPS observations.

To get general idea of accuracy expected from the same lane test, we collected some location data with a Motorola Moto G-3 device which is summarised in table 1. Values obtained were plugged in to equation (1) for different ratios of d_t/d_c, the results obtained are presented in fig. 4. The figure shows that accuracy of the same lane test for an average GPS error of 7 meters, can reach a maximum accuracy of 0.53%. Which means would fail to detect about half of the times cases where the two vehicles would be on the same lane. This claim was actually verified by real test performed, thus we had to look for ways to improve the accuracy of this test.

Remember that we assume that overtaking would be occurring only in straight segments of the road, thus an idea would be using multiple GPS points to find the driving vector of vehicles, this would reduce the later error in their position.

Fig. 5 depicts the actual same lane test. It can be seen from the figure that we now use multiple GPS

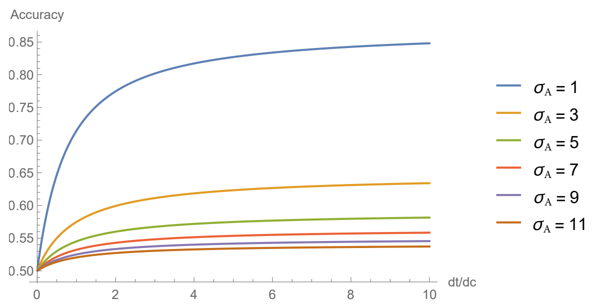

Fig. 4: Simulation of the Same Lane Test.

observations to construct the driving vector of each vehicle, and also correct the current location of the vehicles. Also, to make this test more robust and instead of calculating the perpendicular distance d of the vehicle ahead from the driving vector of the vehicle behind, we use the average of the all distance $d_1, d_2, d_3, ...$ between the corrected positions of the vehicle ahead and their projection on the vector of the vehicle behind. The average of these distances (d) is used for the evaluation of the test.

Since we are considering just undivided highways with two lanes and that the driving vectors coincide with the centerline of the lane, we can use these information to further improve the accuracy of this test. To do so, we find out if the current location of the vehicle ahead is to the right of the driving vector of the vehicle behind, if so, it is directly assumed that both vehicles are on the same lane. Otherwise, we compare the distance d to L/2, if d is less than equal to L/2, then both vehicles are on the same lane. Thus, CAR-B in fig. 6 is considered to be on the same lane if it lies between $+\infty$ to $-L/2$, otherwise on lies on the other lane. Note that the explanation here applies for right hand traffic, although a similar concept may be used for left hand traffic as well.

Next, we performed some tests in the parking area of the Ghent University Schoonmeersen Campus with two vehicles one always ahead of the other on the same lane, and evaluated whether we were able to detect these conditions using our same lane test.

Fig. 7 depicts the results obtained using the same lane test, taking advantage of different regression methods like Least Absolute Deviations (LAD) [14], Linear, Least Trimmed Squares (LTS) [15], Repeated Median [17], and Single Median or the Theil–Sen estimator [3] regression models to minimise the effect of later errors in the measurements. The observations are grouped according to the ratio of the length of the driving vector of the vehicle behind (d_t), and the distance from the vehicle behind to the projection of current corrected location of the vehicle ahead on the driving vector of the vehicle behind (d_c). Linear regression performed best in our case with detection rate varying from 0.63 to 0.75 for increasing values of d_t/d_c. This means that as the vehicle behind approaches the vehicle ahead, the detection rate improves. Note that we were able to improve the performance of the same lane test from 0.53% accuracy as predicted by equation (1) to up to 0.75%.

3.1.2 Same direction test

Once we know that a vehicle from which an advertisement message originates is travelling on the same lane, we need to verify if it is moving in the same direction as the vehicle behind. For this purpose we rely on the same direction test.

Fig. 8 explains how the same direction works. The have the driving vectors of the vehicles travelling ahead and behind. These vectors are constructed with the help of multiple GPS points in the way as the same lane test, and later regression methods are employed to improve the accuracy of the GPS observations. We then calculate the angle between the two vectors. If the calculated angle is under a threshold of 12.5 degrees, then we consider that the cars are travelling in the same direction. The threshold of 12.5 degrees was established from experiments described in [12].

Fig. 9 summarises the performance of the same direction test grouped according to the ratio of d_t/d_c. The experiment was performed using the same dataset as in case of the same lane test. It can be observed that the graph shows similar trends as the same lane test, the performance of the test improves for higher values of d_t/d_c. Also, linear regression outperforms the other regression methods, producing an efficiency between 82 to 98% as the value of d_t/d_c rises from less than 0.33 to more than equal to 0.99. Thus we were able detect up to 98% of the times when the a video advertisement originates from a vehicle travelling in the same direction as the receiving vehicle.

3.1.3 Ahead test

In the previous sections we have seen how to detect if a message source is travelling in the same direction and lane as the message destination. But, we still have to make sure that the message source is ahead of the receiver, before requesting a video stream from it, thus we introduce the ahead test to detect such situations.

Fig. 10 shows how the *Ahead Test* is performed. We construct the driving vectors of the two vehicles using its current and past locations. From these driving vectors, we can find out the corrected locations of each vehicle. We then construct a perpendicular line (P) at

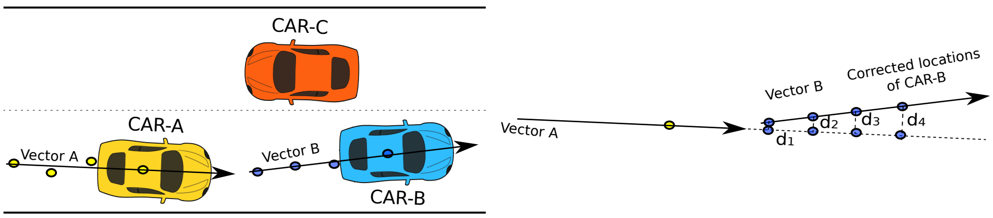

Fig. 5: Same Lane Test.

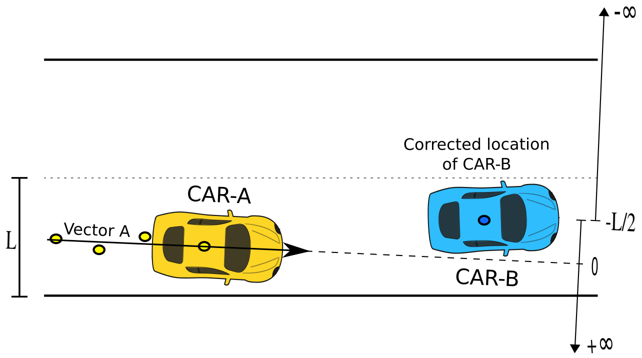

Fig. 6: Limits of the Same Lane Test.

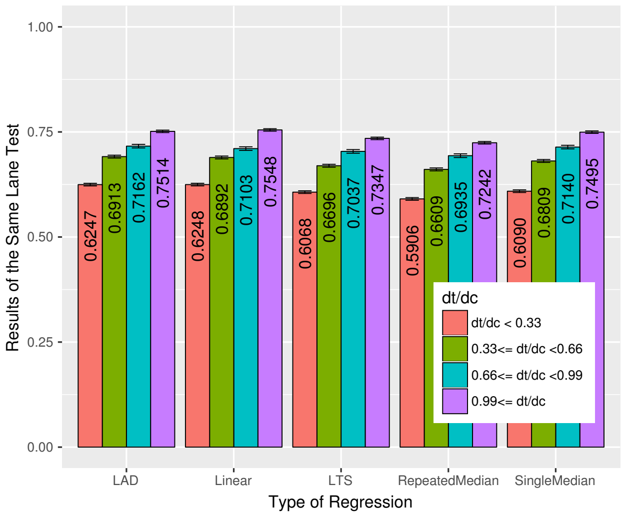

Fig. 7: Result from experiments with the Same Lane Test grouped according to the ratio of d_t/d_c when both vehicles were on the same lane.

the current corrected location of the vehicle travelling behind. If the current corrected location of CAR-B lies to the right of line P, then we can safely say that CAR-B is ahead of the other car.

Fig. 11 confirms that the accuracy in detection of a message source travelling ahead of the receiver drops from 100 to 97% as the value of ratio d_t/d_c increases from less than 0.33, to 0.99 and above. In this experiment, the message source was always ahead of the message receiver. Note that this is the only test where we observe a decline in detection rate as d_t/d_c increases, because we are only able to reduce the lateral error in GPS measurements not the vertical errors. Thus when the vehicle behind moves very close to the vehicle ahead with the intention of overtaking, the detection becomes more difficult.

Thus, we have proposed three test, namely the same lane test, same direction test and ahead test that only relies on location information of the vehicles to help us decide which vehicle should act as the video source and which one should be the video destination. From our experiments we see that with the same direction test and the ahead test, we are able to achieve up to 98% and 100% detection rates respectively. While the same lane test is most affected by GPS errors and a highest efficiency of up to 75% can be expected from it.

3.2 Data fusion approach

The location based approach to find out the relative position of vehicles, although is simple and fast, but it is affected by the accuracy of GPS observations. Thus we wanted to investigate into approach based on sensor data fusion, depending on camera and GPS data.

In this method, when the user first starts the application has to fill in the actual license plate number of the vehicle in which the device running our application would be placed. This information is broadcasted in the advertisement message along with the IP address, heading, and a timestamp. The vehicle behind which is also running our application receives advertisement messages from all one hop neighbours, but retains information of the vehicles that are travelling in the same direction as itself. This means, it will be able to generate a list of all one hop neighbours that are either travelling ahead or behind it in the same direction. Meanwhile, it also makes use of the camera to try and recognise the license plate of the vehicles appearing in the image. Thus, another list of vehicles that are just ahead of the current vehicle but may not be present on the same lane is generated by processing the image. Information from the two lists is merged to select the relevant video source and a request for the video can

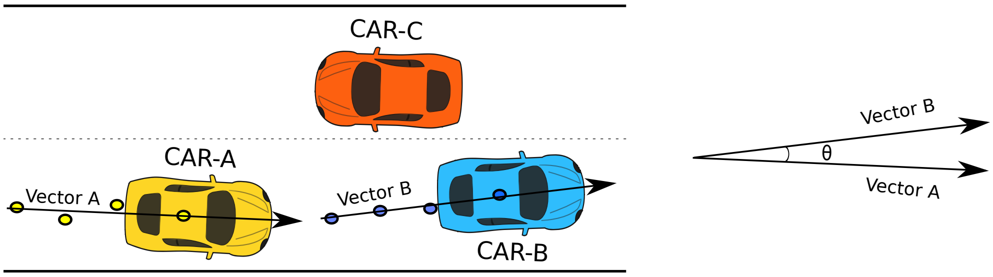

Fig. 8: Same Direction Test.

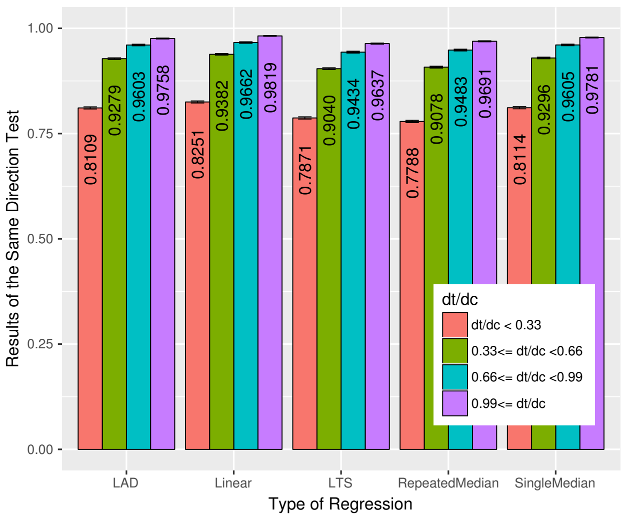

Fig. 9: Results of the Same Direction Test grouped according to the ratio of d_t/d_c when the vehicles are travelling in the same direction.

be sent. The steps involved in license plate recognition are described in [13].

In fig. 12 we see four vehicles: A, B, C and D. Vehicles A, B and C are travelling in the same direction, while vehicle D is travelling in the opposite direction. Consider just the case where video is to be streamed from vehicle C to B. Vehicle B which is the receiver of the video, would first receive advertisement messages from the neighbouring vehicles, containing the license plate and heading information of the message source. Since only vehicles A and C are travelling in the same direction as vehicle B, their information is stored in a list of possible video source. Meanwhile from the image captured by the dashboard mounted smartphone inside vehicle B, we would be able to identify license plates of vehicle C and D, since these vehicles are ahead of vehicle B. Comparing this information with the previous list created from advertisement messages, we find that only vehicle C is common to both and thus is selected as the video source. A video request is sent from vehicle B to C, and video streaming starts.

3.2.1 Static scenario

The first set of experiments with the data fusion approach was conducted in static scenarios. In these test, we acquired images of static vehicles from a parking area, which were later processed by our algorithm. Since this method is computation intensive, we will take a look at the processing time of images by some Android devices.

Fig. 13 shows the processing time of the images of different resolutions by five different android devices, namely Nexus 7 Tablet, Motorola Moto G3, Nexus 5X, Nexus 6 and Samsung Note 10.1. It was observed that images of lower resolutions were processed faster, and devices with faster processors took less time to identify the license plate in the images. The best performance was achieved by Samsung Note 10.1 which processed 1.8 seconds for HD, 1.4 seconds for VGA, and 1.1 seconds for QVGA resolutions.

Next, we define an accuracy term for our algorithm. Since, we are trying to identify the license plate from images to later compare the identified license plate information with the ones acquired from the advertisement messages, so that the video destination can request for the video aid from the video source. The accuracy of our algorithm depends on the percentage of characters of the plate that the algorithm is able to identify correctly. In the next experiment, we calculate the accuracy of plate recognition in the Moto G3 device for varying resolutions.

In fig. 14 apart from experimenting with three different resolutions, we have also varied the JPEG quality of the images from 20 to 80. The quality is a parameter used by the Android OS while encoding JPEG images, the value of quality may range from 1 to 100. The value of 1 means lowest quality while 100 produces images with maximum quality. It was observed that a quality value of below 20, produced very bad quality images, while values above 80 did not produce any significant improvement in the visually perceived quality. Also, the accuracy of algorithm is measured between 0 and 1. The value of 1 reflects an exact match where the identified

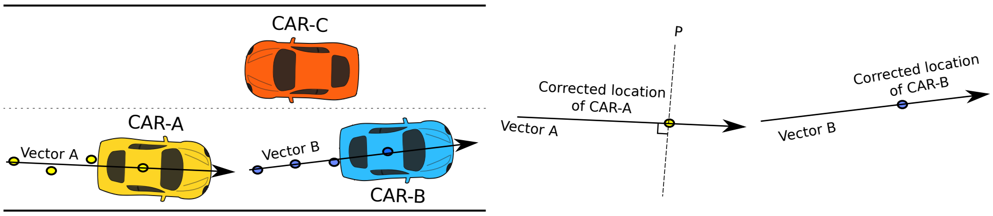

Fig. 10: Ahead Test.

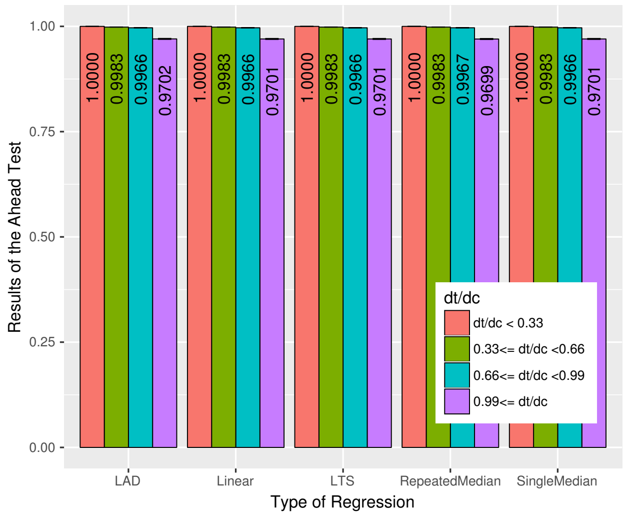

Fig. 11: Result from experiments with the Ahead Test grouped according to the ratio of d_t/d_c when one vehicle is ahead of the other.

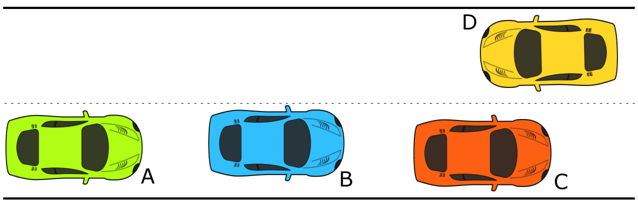

Fig. 12: Required setup while using our overtaking aid application.

plate perfectly coincides with the actual plate, while 0 indicates no match or no plate was detected in the image. It was observed that for lower resolutions, the quality factor has more effect on accuracy than higher resolutions. The accuracy of recognition varied from 0.12 to 0.5 for QVGA, while for VGA it stayed between 0.83 to 0.89, finally for HD resolution it ranged from 0.89 to 0.91.

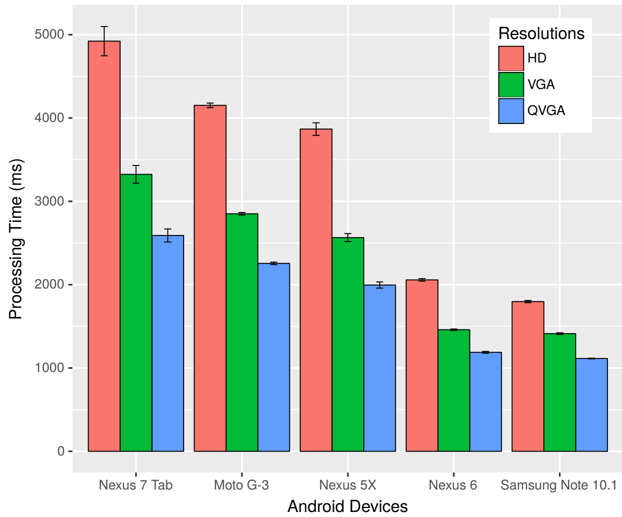

Fig. 13: Time taken to process images of different resolutions by Android devices.

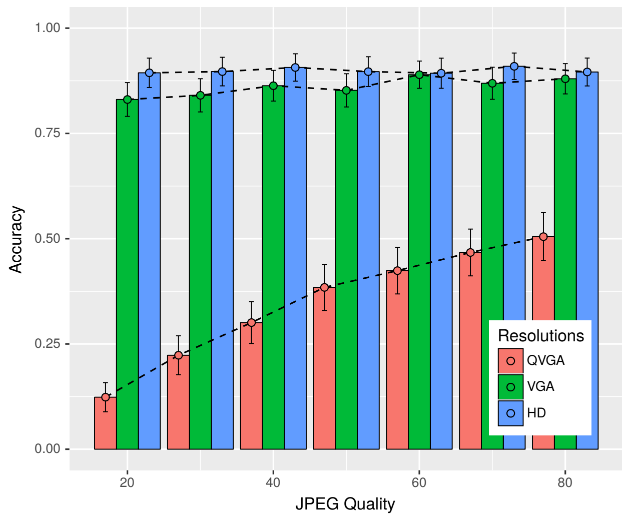

Fig. 14: Accuracy of plate recognition for different JPEG qualities.

3.2.2 Mobile scenario

Once we made sure that the algorithm was functional using data collected from static environments, we put our method to test in mobile scenarios. In this experiment we used two real vehicles with android devices mounted within, and they followed a route 3.76 km long around Universitat Politècnica de València during a sunny day. While the vehicles were in motion, the device within the vehicle behind, tried to identify the license plate of the vehicle ahead. The results obtained are summarised below.

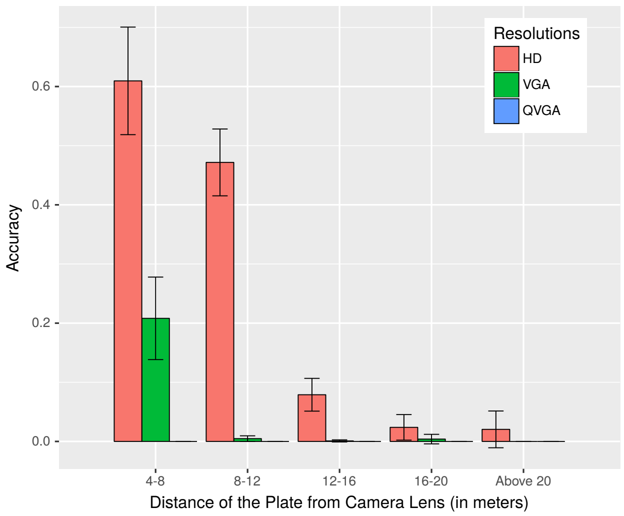

Fig. 15: Accuracy of the recognised plates for varying distances in a mobile scenario.

Fig. 15 show that lower resolutions of VGA and QVGA does not work well in scenarios involving motion. Also accuracy of plate recognition increased as distance between vehicles decreased. The best performance was seen for HD resolution for distances between 4 to 8 meters, with the average accuracy of 0.61. Note that this is lower than what we had observed in our experiments in static scenarios. The average accuracy was lower as the number of failures in plate recognition was higher in scenarios with mobility as the quality of images acquired by the camera was affect by vibrations due to motion.

Thus we see that even though using plate recognition is more robust way to identify the vehicle travelling ahead, it is more computation intensive and the results obtained depend on the quality of the images.

4 Conclusions

In vehicular network scenarios, situations arise where a communication has to be setup between two vehicles are located relative to each other in particular manner: ahead, behind, travelling in the same or different directions. An example of an application that depends on the knowledge is our video overtaking aid which facilitates the streaming of realtime video from the vehicle ahead to the vehicle behind. This application is useful when the view of the driver in the vehicle behind is blocked by the presence of a larger vehicle ahead. In such situations, a video transmission from the dashboard mounted device within the vehicle ahead to the vehicle behind, can aid the driver in seeing what lies ahead and decide if it safe to overtake. For the streaming of the video, the video source has be ahead of the receiver, travelling on the same lane and direction. For identifying if the relative positions of vehicles we had evaluated two different methods, one relying on just GPS data, and the other depends on data fusion from GPS and the camera. The first method depending on only location information, is simpler and faster but if affected by the accuracy of the GPS observations. While the second method, although more robust, is more computationally intensive and its efficiency depends on the quality of the images acquired. Thus the method to be employed depends on the application requirement, for higher speed, the approach based on only GPS data can be employed. While, for applications where accuracy is more important, the second approach that relies on the combination of data from the GPS and camera may be put to use.

Acknowledgements This work was partially funding by the "Ministerio de Ciencia, Innovación y Universidades, Programa Estatal de Investigación, Desarrollo e Innovación Orientada a los Retos de la Sociedad, Proyectos I+D+I 2018", Spain, under Grant RTI2018-096384-B-I00.

References

1. AbdulQawy, A., Elkhouly, R., Sallam, E.: Approaching rutted road-segment alert using smartphone. In: 2018 13th International Conference on Computer Engineering and Systems (ICCES), pp. 341–346 (2018). DOI 10.1109/ICCES.2018.8639363
2. Highway Traffic Safety Administration, N., et al.: National motor vehicle crash causation survey: Report to congress. National Highway Traffic Safety Administration Technical Report DOT HS **811**, 059 (2008)
3. Akritas, M.G., Murphy, S.A., Lavalley, M.P.: The Theil-Sen estimator with doubly censored data and applications to astronomy. Journal of the American Statistical Association **90**(429), 170–177 (1995)
4. Bastani Zadeh, R., Ghatee, M., Eftekhari, H.R.: Three-phases smartphone-based warning system to protect vulnerable road users under fuzzy conditions. IEEE Transactions on Intelligent Transportation Systems **19**(7), 2086–2098 (2018). DOI 10.1109/TITS.2017.2743709
5. Clarke, D.D., Ward, P., Jones, J.: Overtaking accidents. Transport Research Laboratory (1998)
6. El-Wakeel, A.S., Li, J., Noureldin, A., Hassanein, H.S., Zorba, N.: Towards a practical crowdsensing system for road surface conditions monitoring. IEEE Internet of Things Journal **5**(6), 4672–4685 (2018). DOI 10.1109/JIOT.2018.2807408
7. Galarza, E.E., Egas, F.D., Silva, F.M., Velasco, P.M., Galarza, E.D.: Real time driver drowsiness detection based on driver's face image behavior using a system of human computer interaction implemented in a smartphone. In: Á. Rocha, T. Guarda (eds.) Proceedings of

the International Conference on Information Technology & Systems (ICITS 2018), pp. 563–572. Springer International Publishing, Cham (2018)

8. Groeger, J., Clegg, B.: Why isn't driver training contributing more to road safety? In: Behavioural Research in Road Safety IV. Proceedings of a seminar held 6-7 September 1993, Brunel University.(TRL published article PA 3035/94) (1994)
9. Kataoka, K., Gangwar, S., Mudda, K.Y., Mandal, S.: A smartphone-based probe data platform for road management and safety in developing countries. In: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), pp. 612–615 (2018). DOI 10.1109/ICDMW.2018.00095
10. Ma, Y., Zhang, Z., Chen, S., Yu, Y., Tang, K.: A comparative study of aggressive driving behavior recognition algorithms based on vehicle motion data. IEEE Access **7**, 8028–8038 (2019). DOI 10.1109/ACCESS.2018.2889751
11. Mantouka, E.G., Barmpounakis, E.N., Vlahogianni, E.I.: Identifying driving safety profiles from smartphone data using unsupervised learning. Safety Science **119**, 84 – 90 (2019). DOI https://doi.org/10.1016/j.ssci.2019.01.025. URL `http://www.sciencedirect.com/science/article/pii/S0925753517319689`
12. Patra, S., Calafate, C.T., Cano, J.C., Veelaert, P., Philips, W.: Integration of vehicular network and smartphones to provide real-time visual assistance during overtaking. International Journal of Distributed Sensor Networks **13**(12), 1550147717748114 (2017)
13. Patra, S., Zamora, W., Calafate, C.T., Cano, J.C., Manzoni, P., Veelaert, P.: Using the smartphone camera as a sensor for safety applications. In: Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good, GoodTechs '19, pp. 84–89. ACM, New York, NY, USA (2019). DOI 10.1145/3342428.3342659. URL `http://doi.acm.org/10.1145/3342428.3342659`
14. Phillips, R.F.: Least absolute deviations estimation via the EM algorithm. Statistics and Computing **12**(3), 281–285 (2002)
15. Rousseeuw, P.J., Van Driessen, K.: Computing LTS regression for large data sets. Data mining and knowledge discovery **12**(1), 29–45 (2006)
16. Shikishima, A., Nakamura, K., Wada, T.: Detection of texting while walking by using smartphone's posture and acceleration information for safety of pedestrians. In: 2018 16th International Conference on Intelligent Transportation Systems Telecommunications (ITST), pp. 1–6 (2018). DOI 10.1109/ITST.2018.8566727
17. Siegel, A.F.: Robust regression using repeated medians. Biometrika **69**(1), 242–244 (1982)
18. Tornell, S.M., Patra, S., Calafate, C.T., Cano, J.C., Manzoni, P.: GRCBox: extending smartphone connectivity in vehicular networks. International Journal of Distributed Sensor Networks **11**(3), 478064 (2015)
19. Wallace, G.K.: The JPEG still picture compression standard. Communications of the ACM **34**(4), 30–44 (1991)
20. Warren, I., Meads, A., Wang, C., Whittaker, R.: Monitoring driver behaviour with backpocketdriver. In: I. Awan, M. Younas, P. Ünal, M. Aleksy (eds.) Mobile Web and Intelligent Information Systems, pp. 57–70. Springer International Publishing, Cham (2019)
21. Xie, J., Hilal, A.R., Kulic, D.: Driver distraction recognition based on smartphone sensor data. In: 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 801–806 (2018). DOI 10.1109/SMC.2018.00144
22. Xu, X., Yu, J., Chen, Y., Zhu, Y., Qian, S., Li, M.: Leveraging audio signals for early recognition of inattentive driving with smartphones. IEEE Transactions on Mobile Computing **17**(7), 1553–1567 (2018). DOI 10.1109/TMC.2017.2772253