

Vehicular networking and computer vision-based distance estimation for VANET application using Raspberry Pi 3

By Mulia Pratama



Vehicular networking and computer vision-based distance estimation for VANET application using Raspberry Pi 3

Mulia Pratama^{a,*}, Giambattista Gruosso^b, Widodo Budi Santoso^a, Achmad Praptijanto^a

^a Research Center for Electrical Power and Mechatronics, Indonesian Institute of Sciences
Jl. Raya Cisarua No. 154D, Bandung, 40135, Indonesia

^b Department of Electronics, Informatics and Bioengineering, Politecnico di Milano
Via Ponzio 34/5, 20133, Milano, Italia

Received 1 March 2019; accepted 2 December 2019; Published online 14 December 2019

Abstract

This research was implementing vehicle networking using WIFI connection and computer vision to measure the distance of vehicles in front of a driver. In particular, this works aimed to improve a safety driving environment thus supporting the current technology concept being developed for inter-vehicular networking, VANET, especially in its safety application such as Overtaking Assistance System. Moreover, it can wirelessly share useful visual information such as hazard area of a road accident. In accordance with Vehicle-to-Vehicle (V2V) concept, a vehicle required to be able to conduct networking via a wireless connection. Useful data and video were the objects to be sent over the network established. The distance of a vehicle to other vehicles towards it is measured and sent via WIFI together with a video stream of the scenery experienced by the front vehicle. Haar Cascade Classifier is chosen to perform the detection. For distance estimation, at least three methods have been compared in this research and found evidence that, for measuring 5 meters, the iterative methods shows 5.80. This method performs well up to 15 meters. For measuring 20 meters, P3P method shows a better result with only 0.71 meters to the ground truth. To provide a physical implementation for both the detection and distance estimation mechanism, those methods were applied in a compact small-sized vehicle friendly computer device the Raspberry Pi. The performance of the built system then analyzed in terms of streaming latency and accuracy of distance estimation and shows a good result in measuring distance up to 20 meters.

©2019 Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences. This is an open access article under the CC BY-NC-SA license (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).

Keywords: computer vision; Haar Cascade Classifier; distance estimation.

1. Introduction

Video streaming over the inter-vehicular network promises an improvement in road safety and driver-environment safety driving [1]. A vehicle with an intelligent transportation system and a camera adopted able to records surrounding geographic scenes of a highway and events during driving session and stream the video and data to other smart vehicles nearby through the advantage of an inter-vehicular network named VANET [2] for example. In VANET video and data streamed include the information of the road scenery and vehicle motion such as vehicle position, distance, and speed.

Retrieving the relative position i.e. the longitudinal distance of a vehicle in front of the driver by the camera would be the main objective of this work. With this technology enabled in a vehicle, it encourage the decision making by drivers to for example overtaking a road lead [3] thus aiming to support the service of the basic safety application built based on SAE J2735 BSM via WIFI or DSRC/WAVE message protocol [4] in V2V and V2I communication [5][6] concept such as Overtaking Assistance System [3], Forward Collision Warning and emergency electronic brake light (EEBL).

Similar works to this research were the research conducted by [7][8], employing the same vehicle detection methods, however, different methods to estimate the distance between vehicle and camera were proposed in this research. The works are done by [8] estimated distance by means of using geometric relation yielded a quite good result of accuracy 90.29 %

* Corresponding Author. Tel: +62-81394486064
E-mail address: mulia.pratama@lipi.go.id

doi: <https://dx.doi.org/10.14203/j.mev.2019.v10.7-16>

2088-6985 / 2087-3379 ©2019 Research Centre for Electrical Power and Mechatronics - Indonesian Institute of Sciences (RCEPM LIPI). This is an open access article under the CC BY-NC-SA license (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).
Accreditation Number: (LIPI) 633/AU/P2MI-LIPI/03/2015 and (RISTEKDIKT) 1/E/KPT/2015.

with claimed error average 9.71 %. The research done by [9] utilized a homography matrix and Rodrigues Identity to compute rotation matrix, the distance between vehicle calculated from this rotation matrix, however, it used a different method to detect the vehicles with the one proposed in this research.

This research mainly conducted on a card-sized computational resource namely Raspberry Pi 3 with a camera as image input; all vehicle detection, image processing and the computing of linear algebra done using the computer. Haar Cascade Classifier used for vehicle detection as this method comparably the fastest to date. If a detection occurred, the algorithm sends the vehicle position and width annotated by the detection mechanism to the distance measurement algorithm which consists of several methods which will be compared for the performance namely the Triangle Similarity, SolvePnP-Iterative, and SolvePnP-P3P with and without Kalman Filter. The useful data of longitudinal distance calculated, and visual images then are being sent via WIFI to other vehicles or Road-Side Unit.

Considering the ADAS roadmap and its development and V2X which include VANET, this research would contribute to improving people and road safety. The outcome of this research could increase driver awareness of the surrounding for example by adding alarm if cars or objects are within the proximity of a vehicle, also hopefully contribute to autonomous vehicle development.

II. Materials and Methods

This section describes in detail of methods used in this research. Mainly this section study about vehicle detection utilizing Haar Classifier and distance measurement methods. The workflow of the system under discussion described in the following Figure 1.

The camera captured an image frame to be analyzed for vehicular features indicating the frame consist of a vehicle image. If a vehicle confirmed then a bounding rectangle which approximates the width of the vehicle image is applied. Either the width or the coordinates of the rectangle drawn would then be used to estimate the distance between vehicle and camera. Distance information then sent via a wireless network to the other vehicles.

The system developed was implemented in a Raspberry Pi 3 platform including the usage of its 8 MP Raspberry Pi camera. Data and video were streamed via Raspberry Pi 3 embedded WIFI through Named Pipe FIFO and NetCat as a networking tool. The on-board WIFI chip is the advantage of Raspberry Pi 3 compared previous Raspberry Pi releases that require extra USB WIFI dongle, however, this research avoids the use of external USB WIFI dongle since there are many WIFI dongle manufacturers with their own characteristic in the market also some of them have an issue working with Raspberry Pi. Moreover, they might need more power than a Raspberry Pi port can support.

A. Haar-Cascade classifier

There exist a fast and trainable method for object recognition, The Haar Cascade Classifier [10] this method also used by [11] with the detector trained by BIT-Dataset and conclude that Haar Cascade Classifier is a good candidate for object detector. [12] combines the Haar detector with the K-nearest neighbour technique to identify the license plate of vehicles that claimed to offer great efficiency for practical use. According to [13], the classifier superior performance over image-intensity based algorithm encourages the use of Haar Cascade in this work.

Haar Cascade adopted by [10] was a machine learning mechanism. The classifier evaluates the

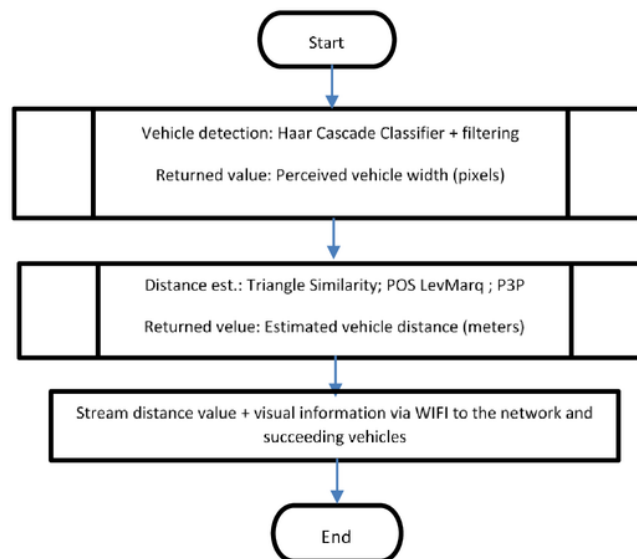


Figure 1. System flowchart



Figure 2. Haar feature shapes used to evaluate image features

simple feature of an incoming image rather than processing the image by pixel. As can be seen in Figure 2, image feature extraction involves Haar features, a geometrical shape of a rectangle consisting of a pair region of white and black. Value of each image feature determined by the sum of the pixel under the white Haar region deducted from the sum of the pixel under the black one. Thus, the white rectangle weight 1 (one) and black region weight -1 (minus one).

Two-rectangle feature (a) [10] used to extract edge feature on image and three-rectangle feature (b) used to extract line feature on image. For example, Figure 3 shows an applying Haar basic feature shapes over an image of a vehicle. The pixels sum of the black and white are summed up (by subtraction) and the difference is the value of the image feature underlying the mask Haar feature. To immediately evaluate the features and the author of [10] presented auxiliary Integral Image. Any large number of features extracted, their values can be quickly calculated with only four points of reference.

B. Triangle similarity

A simple spatial measurement based on geometrical properties of two similar triangles applied in this research referred to [14][15] and then be compared to other methods described later. Fed by a known distance of a camera to a vehicle and a known width of the vehicle, one can obtain a focal length of the camera. As can be seen in Figure 4, if DF is the real width of the vehicle and Ed is the real distance between camera to vehicle then one can find Bd' as the focal length of camera in pixel, provide that information of AC is available.

Information of AC length provided by the detection and tracking mechanism discussed previously and referred to as vehicle width in a pixel can be expressed using equation (1)

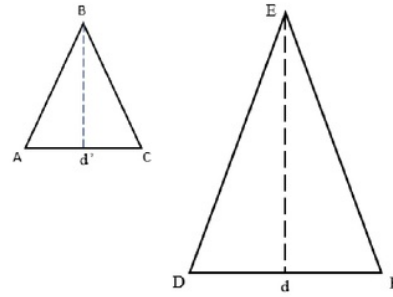


Figure 4. Two similar triangles

$$\frac{DF \text{ (meters)}}{Ed \text{ (meters)}} = \frac{AC \text{ (pixels)}}{Bd' \text{ (pixels)}} \quad (1)$$

by reversing (1), provided the focal length, real vehicle width in meters and perceived vehicle width in pixels, one can estimate the distance from the camera to the vehicle $E-d$. However, this is true if the vehicle positioned assumed exactly on the camera centerline.

There are cases when vehicle position deviates from camera centerline a few meters and as result for this to happen is that the focal length Bd' of the camera needs to be reinitialized since the distance of the vehicle to the camera now is different for that particular position. [14] did not examine this problem as a contrast to [15] that decide not to recommend using Triangle Similarity if such a case happened. The reinitialize of focal length can be done by a camera calibration process either automatically or manually which is impossible to be carried out while vehicle running on the road. To maintain the parameter the same for the analysis, this method tested with the assumption above retained.

C. Iterative using Levenberg-Marquardt optimization

The Triangle Similarity method previously described then will be compared with a perspective projection-based method. Figure 5 describes a Pinhole Camera mechanism to better gain insight to understand the Perspective-n-Point (PnP). One can infer from Figure 5, a model of a 3D box in universe reference



Figure 3. Haar feature used to detect rear windshield and rear tires of a vehicle that are distinctive feature of a vehicle

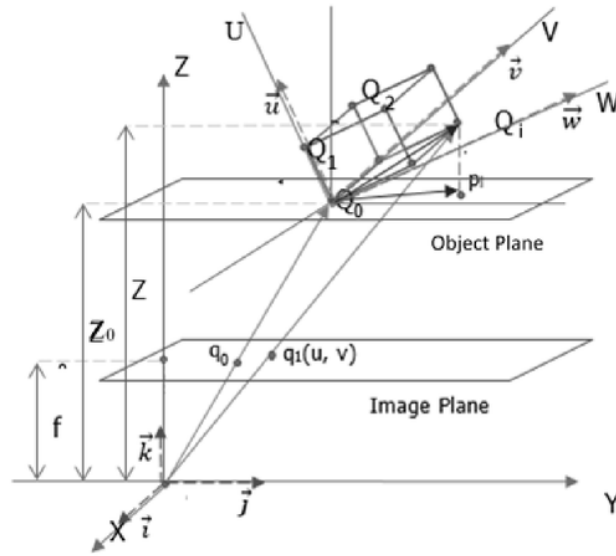


Figure 5. Pinhole camera illustration

characterized by its points Q_0 to Q_i is projected to a 2D image plane using perspective transformation. The model reconstructed in the image plane through a series of matrix operation involving rotation matrix refer to each axis and a translation matrix. Rotation matrixes can be simplified into a single 3x3 matrix R

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (2)$$

The vector of translation matrix T

$$T = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad (3)$$

Also, consider a camera matrix one interested to obtain

$$\text{camera matrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (4)$$

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (5)$$

A point of the model reconstructed in the image plane (u, v) by (2), (3) and (4) according to (5) that is the Pinhole Camera equation where: X, Y , and Z is the real coordinate of a point on a cube for example point Q_i . In universe of observer's reference system. u and v is the coordinate of point q_i in the observer's image plane reference system. Hence, q_i is the perspective projection of point Q_i of the model. c_x and c_y are principal points usually at the image center. f_x and f_y is the focal length of the camera used. S is scaling factor. The aim here is to find the Translation matrix T and

retrieve its value as the distance between a vehicle and camera one interested.

This research used the OpenCV library as computer vision tool to accomplish the computation. Using OpenCV solvePnP with an ITERATIVE method it follows that the computer vision using Levenberg-Marquardt Optimization to minimize the reprojection error that was the sum of the squared deviation between a point and its projection.

D. The P3P problem

OpenCV provides a way to solve the P3P problem according to [16]. In OpenCV, the function SolvePnP can be used to estimate the pose of an object model that is to obtain the rotation matrix R and translation matrix T. Flag SOLVEPNP-P3P uses 3 object points for calculating the pose plus one additional point to provide the best result.

The works by [16] present the complete solution classification for the Perspective-Three-Points (P3P) that shown P3P has multiple solutions from an algebraic perspective under certain real conditions. To understand the problem in research [10], Figure 6 provides a solution perspective where P, A, B, C are not-coplanar and P the center of perspective. Moreover, a, b' , and c' are $|CB|$, $|AC|$, and $|AB|$ respectively and $\alpha = \angle BPC$, $\beta = \angle APC$, $\gamma = \angle APB$, $p = 2 \cos \alpha$, $q = 2 \cos \beta$, $r = 2 \cos \gamma$. It also notified that X, Y, Z are representing the distance from point P to the corresponding point. From triangle PAC, PAB and PBC, P3P equation can be written as (6)

$$\begin{cases} Y^2 + Z^2 - YZp - a'^2 = 0 \\ Z^2 + X^2 - ZXq - b'^2 = 0 \\ Z^2 + Y^2 - XYr - c'^2 = 0 \end{cases} \quad (6)$$

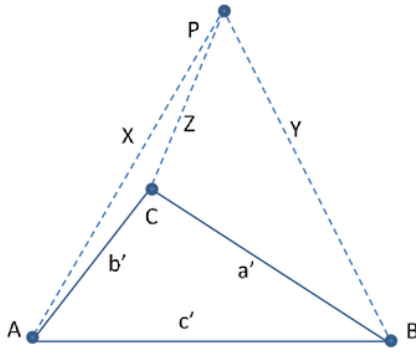


Figure 6. Illustration of P3P problem origin

according to [16] a set of solutions exist for X, Y, Z that are the space from the camera to the object, if comply with some condition specified on their paper. Substituting $X = xZ, Y = yZ, c' = \sqrt{v}Z, a' = \sqrt{av}Z, b' = \sqrt{bv}Z$ to (6), divided by Z it become

$$\begin{cases} y^2 + 1 - yp - av = 0 \\ x^2 + 1 - xq - bv = 0 \\ x^2 + y^2 - xyr - v = 0 \end{cases} \quad (7)$$

thus $v = x^2 + y^2 - xyr > 2$, Z can be obtained by $Z = c' / \sqrt{v}$, this value was the first order approximation of vehicle distance one interested. Moreover, one may find that by removing v from (7) one may turn to a quadratic equation which yield a finite number of solutions if not at least four.

E. Kalman filter

To smoothen thus hopefully improve the distance accuracy, Kalman Filter used with the help of noise characterizing to attain optimum Kalman Filter. The filter works based on a loop of measuring, predict and update. In order for the filter to work, a certain matrix should be provided: Transition matrix F , Measurement matrix H , Q process noise matrix, R the measurement noise that is the noise characteristic that will be discussed in the next part of the paper. Each measurement z as a result of the detection process will

be passed as matrix H . The filter then predict the next state of the system by

$$\bar{x} = Fx \quad (8)$$

$$\bar{P} = FPF^T + Q \quad (9)$$

with \bar{x} is the next state of the system predicted; x is the current (posteriori) state; \bar{P} predicted the state of covariance or error; and P is the current state of covariance. Then the filter updates the state of the system modified with the predicted state by the following equation

$$x = \bar{x} + Ky \quad (10)$$

$$P = (I - KH)\bar{P} \quad (11)$$

with z is the measurement and y is the residual defined as

$$y = z - H\bar{x} \quad (12)$$

and K as Kalman Gain

$$K = \bar{P}H^T(H\bar{P}H^T + R)^{-1} \quad (13)$$

Hereafter, the current state of the system x is updated and the value can be used for the distance measurement algorithm.

III. Results and Discussions

A. Streaming latency

The system was investigated for its performance in terms of latency, noise, and distance estimation accuracy. End-to-End latency mainly consists of 680x480 pixels video streaming latency and image processing which included the Haar Cascaded detection mechanism and distance estimation computation plus other additional negligible latency. The latency measurement for video streaming done on-line through with the help of an auxiliary camera and a digital stopwatch to count the whole measurement elapsed time. Auxiliary camera used taken from front laptop camera, the setup is illustrated in Figure 7. With this setup, both video previews from both cameras must be displayed adjacent to each other on the same display in a way so that when a print-screen button hit detected by

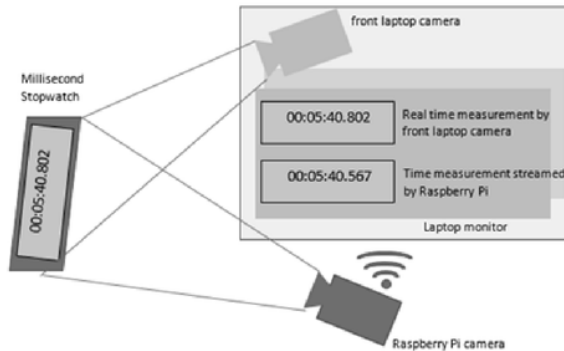


Figure 7. Latency measurement set up



Figure 8. Real-time vs. Streamed latency measurement

the laptop, its screen displaying both measurements is saved to be analyzed. The real setup for measuring video-only stream can be seen in Figure 8, it shows the area of the laptop monitor screen. Figure 8 is displayed as it is captured from the laptop screen. Both windows are inverted because each of these windows shows two cameras reading of the same stopwatch displayed back on the laptop screen.

The foremost display showed captured stopwatch image from local laptop camera and provided real measurement time counts about 21,780 milliseconds from start and served as offset basis to calculate End-to-End latency, while the image behind it was the streamed stopwatch images from Raspberry Pi which provided the streaming delayed behind by 297 counts at 21,483 milliseconds, real-time counts assumed ideal with no delay; note that milliseconds counting marked by the oval making. It was obvious that the time counted for real-time showed larger counts than the streaming, indicating that the stopwatch image frame from Raspberry Pi was experienced 297 milliseconds delay during travel to the laptop display.

Table 1 reports the result of video streaming latency measurement included the delays for detection mechanism and distance measurement computation. The rightmost column was the End-to-End latency calculated by (14)

$$\text{End-to-End} = \text{realtime elapsed time} - \text{streaming elapsed time} \quad (14)$$

Table 1.
End-to-end latency measurement

Methods	Time elapsed (in milliseconds)		
	Realtime	Streaming	Delta
ITERATIVE LevMarq	49:653	48:348	1,305
P3P	50:863	49:363	1,500
Triangle Similarity	55:524	54:034	1,490
video only	21:780	21:483	297

The total End-to-End latency measured 1,490 milliseconds for using the Triangle Similarity method and 1,436 milliseconds for ITERATIVE Levenberg-Marquardt while P3P showed no improvement, the delay was very huge. The allowed latency for the inter-vehicular network should be less than 200 milliseconds referred to [3] and [17]. Even for the aforementioned video-only streaming measured 297 milliseconds were out of the limit.

An Investigation for the source of such massif delay had been done by means of employing the time stamp function inserted in the program script. The time stamp started to count just after the image captured and ended just after the distance computation finished and the image frame ready to be sent to the stream. Table 2 showed the time consumed. From the table above can be seen that even the time needed to do the computing was about one-fourth of total End-to-End latency. From this experiment, it can be assumed that the source of the long delay was not the detection and computing process. The most logical suspect caused the delay was the transmission path or the client unit that receive the video and data stream. For future research, the latency needs to be investigated whether it is device specific by means of comparing it to a more reliable computational resource such as PC with good GPU. The usage of external USB WIFI dongle also needs to be considered for future research if not using the DSRC/WAVE standards for V2X.

Table 2.
Detection and distance computation latency in milliseconds

Methods	Start	End	Latency
P3P	40:850	41:269	419
ITERATIVE LevMarq	35:849	36:285	436
Triangle Similarity	22:379	22:797	418

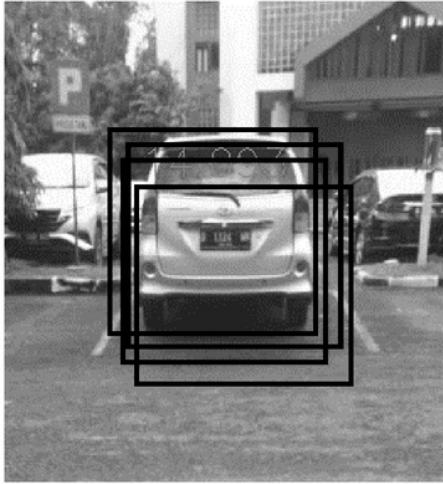


Figure 9. Noisy vehicle markings

B. Noise characteristic

Noise performance is critical for both visual and accuracy since the addition of noise worsens the visualization of the vehicle being monitored as the rectangle marking spread randomly around the vehicle image thus alters the result of distance measurement as an error. Figure 9 showed the fluctuation in vehicle identification, indicated by the dispersion of the bounding box around the vehicle image.

The detection comes from the cascaded classifier was fluctuating in terms of vehicle position on the image at each measurement frame every half second. Since the nature of the cascade classifier applied extensive numbers of Haar feature by sizes and shapes [10] in sliding window, the results produced resembles a random process which generates a stationary white noise added to the measurement, this part intended to investigate the characteristic of the noise whether it is

white or not and to see if optimum Kalman Filter can be exploited. White noise defined as a series of uncorrelated random variables following a continues distribution with zero mean and unity variance also a process is a white noise if its close-spaced time series sample is uncorrelated [18], moreover, if those samples were independent, it follows Gaussian distribution hence guaranteed for randomness.

The following Figure 10 shows the fluctuation of 1000 measurement of for example 15 meters distance using the P3P method. One may effectively eliminate the noise online by employing Optimal Kalman Filter. To this aim, one had to verify that the noise was white for the filter can be optimally utilized. As the figure below is shown, the zero mean and standard deviation of 0.904 are verified by the data.

Autocorrelation must be checked since it guarantees the randomness of the consecutive data generation. Figure 11 was the autocorrelation under confidence level of 95 %. The figure showed that the data was uncorrelated since most of the lag autocorrelations fell between the limit of the shaded area. Disregards lag 0, only one out of every 20 lags lie outside this limit, it was normal since it has a 95 % confidence level. The figure also showed that there were no patterned lags, for example a repeating lag position.

Another method to verify the noise is by mean of the Anderson-Darling distribution test [19], the data were tested against the normal distribution. The Anderson-Darling test return test statistic A^2 to be compared to a critical value of 0.576, 0.656, 0.787, 0.918 and 1.092 for significant level 10 %, 5 %, 2.5 %, 1 % and 0.5 % respectively. If the test returns A^2 value greater than the value mentioned above it then the test sample not likely produced from a normal distribution. Fed the measurement data to the test procedure, it returns A^2 value of 0.574, 0.653, 0.784, 0.914, and 1.088 respecting the same significant level indicating that the measurement data have normal (Gaussian) distribution. By those facts, it is concluded that the fluctuation generates a white Gaussian noise.

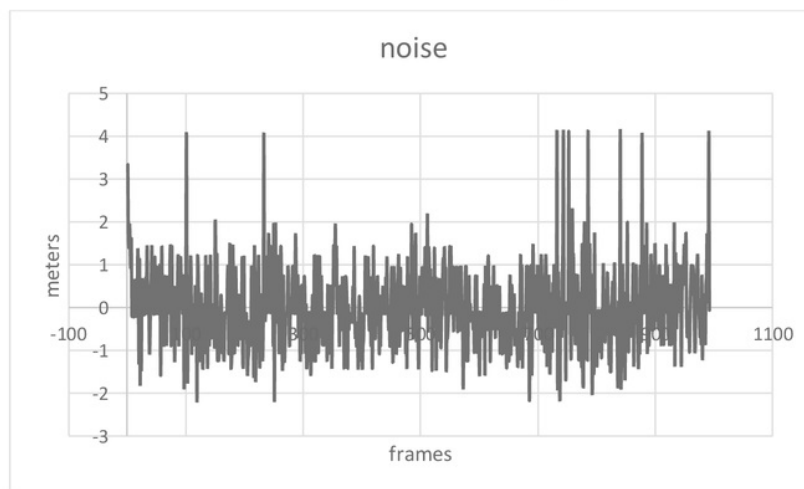


Figure 10. Noise from measuremet of 15 meters using P3P method with mean value subtracted

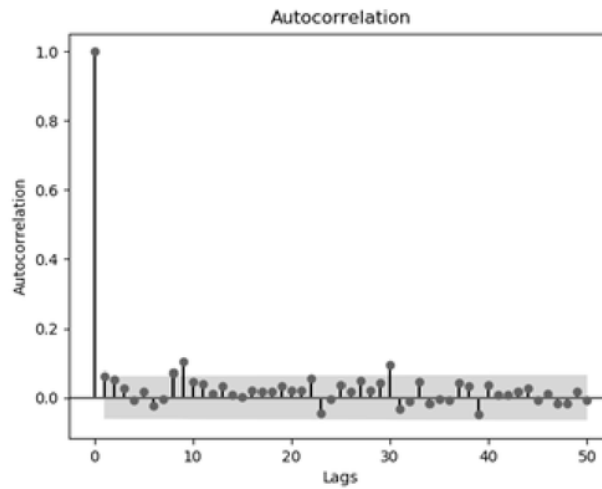


Figure 11. Autocorrelation from measurement data for 15 meters using P3P method

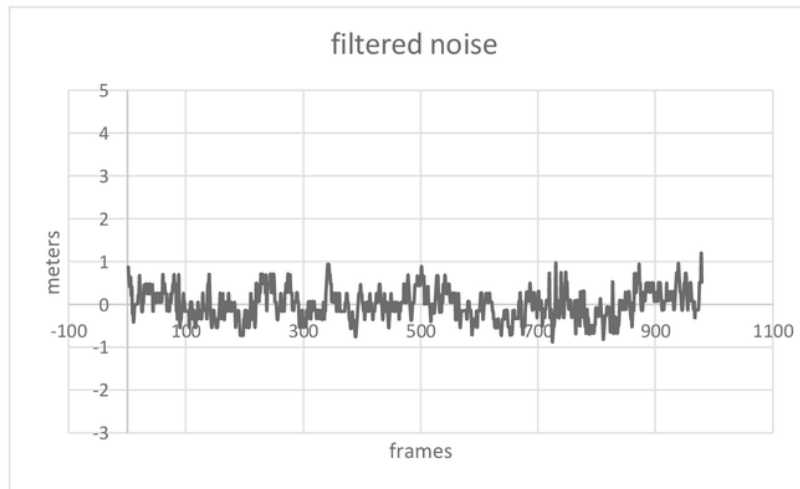


Figure 12. Noise from measurement of 15 meters using P3P method after filtering

With the noise characterized, the Kalman Filter can be optimally employed to the measurement with the assumption that the simulated process noise was also white and uncorrelated to the noise being discussed. With the unfiltered average signal of 15.48 meters and the noise rms of 0.900 meters, one may find the Signal to Noise Ratio (SNR) is 54.40 dB respect to the following formula (15)

$$SNR = 20 \times \log\left(\frac{\text{Signal}}{\sqrt{\text{Noise}^2}}\sqrt{N}\right) \quad (15)$$

Hopefully, Kalman Filter reduces the noise up to an order of magnitude. The result of filtering shows the noise reduction, Figure 12 showed the fluctuation after filtering. The noise rms improves to 0.34 meters but the signal average increase 15.51 meters thus improves SNR to 62.87 dB less than the expected, however, this adds offset error to the true 15 meters up to 0.51 meters

from 0.48 meters unfiltered originally. Kalman Filtering improves the visualization but worsens the accuracy to 3 cm for measurement of 15 meters distance. However, this assumption will be verified by investigating the observed behavior in other methods.

C. Performance of distance estimation

The system was tested in 5 to 20 meters involving three methods for distance estimation. These methods are discussed in this part were the Triangle Similarity, ITERATIVE with Lavenberg-Marquardt Optimization and P3P, last two adopted from camera imaging and homography technique. The experiment applied on one type of vehicle to maintain the uniformity of parameters so it was easier to be analyzed for example the camera matrix that contains focal length calibrated using the most common city car width about 1.64 meters.

Tabel 3.
Measurement results without filtering

Ground Truth	ITERATIVE LevMarq (Std. Dev)	P3P (Std.dev)	Triangle Similarity (Std. Dev)
5	5.80 (0.17)	6.78 (0.55)	6.24 (0.18)
10	10.77 (0.50)	12.63 (0.53)	11.80 (0.54)
15	13.28 (0.89)	15.48 (0.81)	14.61 (0.94)
20	16.84 (1.03)	19.29 (1.07)	18.55 (1.11)

Table 4.
Measurement Results Filtered

Ground Truth	ITERATIVE LevMarq (Std. Dev)	P3P (Std. Dev)	Triangle Similarity (Std. Dev)
5	5.79 (0.13)	7.14 (0.14)	6.23 (0.14)
10	10.88 (0.32)	12.70 (0.39)	11.79 (0.39)
15	13.31 (0.37)	15.51 (0.35)	14.53 (0.36)
20	16.92 (0.61)	19.38 (0.63)	18.48 (0.64)

Moreover, measuring with various kinds of vehicles avoided because it will guide to the further comparative analysis of the performance of the Haar Cascade Classifier such as True Positive rate (TPR) which is not the scope of this paper. The dataset for training the classifier used here is not standard datasets such as BIT-dataset, KITTY or EPFL. All methods will be compared for its performance in terms of accuracy for a filtered and unfiltered result for about 1000 samples averaged. The setup was a parking area with markings every two meters. A car was set in a position while Raspberry Pi and the camera placed in a distance of 5, 10, 15, and 20 meters from the rear end of the car. Light intensity not recorded thus to maintain the same ambient and environment condition the camera also records a video while measuring, the video will be used to test the other methods.

Table 3 present the averaged result for unfiltered 5 and 10 meters; the table also provides the standard deviation from the measurement placed inside the parentheses next to the averaged values. From the table, we can infer that the smallest offset error to 5 and 10 meters attained by the ITERATIVE Levenberg-Marquardt method with an error less than 1 meter to the true value while P3P contributes to the biggest offset, 1.78 meters and 2.63 meters for 5 and 10 m respectively. The performance gets more complicated when referring to Table 3 for 15 and 20 meters since the smallest offset error achieved by Triangle Similarity and P3P methods, they are 0.39 meters and 0.71 meters respectively retain the offset down below 1 meter respecting the ground truth value. The largest offset error now held by ITERATIVE with Levenberg-Marquardt iteration yields 1.72 meters and 3.16 meters respectively for 15 and 20 meters. As for the standard deviation, they will be compared with the filtered measurement and be examined for any improvements. Passing the measurements to a Kalman Filter, Table 4 resumes the observation. Comparing to the previous table, the difference between unfiltered and filtered result is not significant, the biggest come from 5 meters of P3P, and it differs up to 30 centimeters.

From this experiment employing ITERATIVE with Lavenberg-Marquardt and Kalman Filtering was the best practice with accuracy 84.2 %, the error rate was 15.8 % for 5 meters distance and for 10 meters the accuracy was 91.2 %. Meanwhile, for 15 and 20 meters, the P3P method was reliable enough with accuracy 96.9 % and an error rate of 3.1 % at best. Respectively compared to the unfiltered measurement, for 5 and 10 meters the maximum accuracy was 92.3 % produce error rate 7.7 % while for 15- and 20-meters maximum accuracy was 96.8 % with error rate 3.2 %. The filtering effect also can be seen especially for measuring 15 and 20 meters; the standard deviation reduced to about 40 % to 60 % of the initial unfiltered measurement.

IV. Conclusion

To embody the research to a working experiment, a system was built incorporated Raspberry Pi 3 as its main computational Platform. Using the system, there are two main mechanisms discussed in this paper: video streaming latency and distance estimation accuracy. Video streaming was able to be executed by the system but the latency was very huge up to 1500 milliseconds in which more than one-fourth of it was the process latency but VANET required less than 200 milliseconds for vehicular video streaming thus in term of latency the system built was not satisfied, the biggest delay was caused by the transmission or the client for the first approximation, required further research on this field. Suggestion for future research is to use WIFI dongle with external power supply or with more advanced topics, the adoption of DSRC/WAVE chip V2X standard instead of ordinary WIFI. Distance estimation highly dependent on the result of the vehicle detection mechanism mainly its fluctuation, to address that matter Kalman filter employed to filter the noise; unfortunately, it added offset error to the real measurement up to 30 centimeters at 5 meters. Moreover, in terms of accuracy the system built has good accuracy rated at 96.9 % for measuring 20 meters incorporating solve PnP-P3P method.

Acknowledgment

The authors would like to express the deepest appreciation to all those who provided a suggestion, comment, advice, and assistance on the completion of this research. Special gratitude would be given to Mr. Ahmad Dimyani, who's contributed to substantial suggestions and also coordinate the completion of the project. Furthermore, the authors would also like to acknowledge the role of the staff of TELIMEK ICE Laboratory, who assists in the use of all required equipment, the provision on the necessary materials, and also on the complete sampling images collection. Special gratitude was presented to Mr. Amin and Mr. Kristian Ismail, who helped the authors on assembling the whole prototype and apprehending the task of filtering measurement.

References

- [1] M. Mittal, "A Study of Live Video Streaming over Highway Vehicular Ad hoc Networks," *Int. J. Comput. Appl.*, vol. 1, no. 21, pp. 86–90, Feb. 2010.
- [2] C. Iza-Paredes, A. M. Mezher, and M. Aguilar Igartua, "Adaptive Video-streaming Dissemination in Realistic Highway Vehicular Ad-Hoc Networks," in *Proceedings of the 13th ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks - PE-WASUN '16*, 2016, pp. 1–10.
- [3] A. Vinel, E. Belyaev, K. Egiazarian, and Y. Koucheryavy, "An Overtaking Assistance System Based on Joint Beaconing and Real-Time Video Transmission," *IEEE Trans. Veh. Technol.*, vol. 61, no. 5, pp. 2319–2329, 2012.
- [4] B. Cronin, "Vehicle Based Data and Availability," U.S. Department of Transportation, 2016.
- [5] U. D. Gandhi, A. Singh, A. Mukherjee, and A. Chandak, "Smart vehicle connectivity for safety applications," in *2014 International Conference on Reliability Optimization and Information Technology (ICROIT)*, 2014, pp. 262–266.
- [6] C. B. Math, H. Li, and S. Heemstra de Groot, "Risk Assessment for Traffic Safety Applications with V2V Communications," in *2016 IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–6.
- [7] N. F. Mustamin, "Relative distance measurement between moving vehicles for manless driving," in *2017 International Seminar on Application for Technology of Information and Communication (ISemantic)*, 2017, pp. 1–4.
- [8] A. A. Ali and H. A. Hussein, "Distance estimation and vehicle position detection based on monocular camera," in *2016 Al-Sadeq International Conference on Multidisciplinary in IT and Communication Science and Applications (AIC-MITCSA)*, 2016, pp. 1–4.
- [9] H.-K. Cheung, W.-C. Siu, S. Lee, L. Poon, and Chiu-Shing Ng, "Accurate distance estimation using camera orientation compensation technique for vehicle driver assistance system," in *2012 IEEE International Conference on Consumer Electronics (ICCE)*, 2012, pp. 227–228.
- [10] S. Jabri, M. Saidallah, A. El Belrhiti El Alaoui, and A. El Fergougui, "Moving Vehicle Detection Using Haar-like, LBP and a Machine Learning Adaboost Algorithm," in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, 2018, pp. 121–124.
- [11] E. Baser and Y. Altun, "Detection and classification of vehicles in traffic by using haar cascade classifier," *Proc. 58th ISERD Int. Conf. Sci. Innov. Eng.*, no. December, pp. 19–22, 2016.
- [12] M. Rokonzaman, M. A. Al Amin, M. H. K. M. U. Ahmed, and M. T. Rahman, "Automatic vehicle identification system using machine learning and robot operating system (ROS)," in *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, 2017, pp. 253–258.
- [13] Y. Hasan, M. U. Arif, A. Asif, and R. H. Raza, "Comparative analysis of vehicle detection in urban traffic environment using Haar cascaded classifiers and blob statistics," in *2016 Future Technologies Conference (FTC)*, 2016, pp. 547–552.
- [14] G. Kim and J.-S. Cho, "Vision-based vehicle detection and inter-vehicle distance estimation for driver alarm system," *Opt. Rev.*, vol. 19, no. 6, pp. 388–393, Nov. 2012.
- [15] R. K. Megalingam, V. Shriram, B. Likhith, G. Rajesh, and S. Ghanta, "Monocular distance estimation using pinhole camera approximation to avoid vehicle crash and back-over accidents," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–5.
- [16] X. Ying, G. Wang, X. Mei, S. Yang, and H. Zha, "The perspective-3-point problem when using a planar mirror," *Proc. - Int. Conf. Pattern Recognit.*, pp. 4033–4037, 2014.
- [17] B. Bellalta, E. Belyaev, M. Jonsson, and A. Vinel, "Performance Evaluation of IEEE 802.11p-Enabled Vehicular Video Surveillance System," *IEEE Commun. Lett.*, vol. 18, no. 4, pp. 708–711, Apr. 2014.
- [18] R. M. Howard, "White noise: A time domain basis," *2015 Int. Conf. Noise Fluctuations, ICNF 2015*, pp. 1–4, 2015.
- [19] P. Closas, J. Arribas, and C. Fern, "Testing for normality of UWB-based distance measurements by the Anderson-Darling statistic," *Futur. Netw. Mob. Summit 2010 Conf. Proc.*, pp. 1–8, 2010.

Vehicular networking and computer vision-based distance estimation for VANET application using Raspberry Pi 3

ORIGINALITY REPORT

6%

SIMILARITY INDEX

PRIMARY SOURCES

1	Kadek Heri Sanjaya, Yukhi Mustaqim Kusuma Sya'Bana, Shaun Hutchinson, Cyriel Diels. "Preliminary investigation of sleep-related driving fatigue experiment in Indonesia", Journal of Mechatronics, Electrical Power, and Vehicular Technology, 2018 <small>Crossref</small>	194 words — 4%
2	media.neliti.com <small>Internet</small>	29 words — 1%
3	epdf.tips <small>Internet</small>	26 words — < 1%
4	"Advances in Image and Graphics Technologies", Springer Science and Business Media LLC, 2015 <small>Crossref</small>	16 words — < 1%
5	home.deib.polimi.it <small>Internet</small>	13 words — < 1%
6	"Computational Fluid Dynamics for Engineers", Springer Nature, 2005 <small>Crossref</small>	8 words — < 1%
7	Christiana Ioannou, Vasos Vassiliou. "Security Agent Location in the Internet of Things", IEEE Access, 2019 <small>Crossref</small>	8 words — < 1%
8	Zaiwen Wen. "A Line Search Multigrid Method for Large-Scale Nonlinear Optimization", SIAM Journal on Optimization, 2009	8 words — < 1%

9

Ki-Yeong Park, Sun-Young Hwang. "Robust Range Estimation with a Monocular Camera for Vision-Based Forward Collision Warning System", The Scientific World Journal, 2014

Crossref

7 words — < 1%

10

Jianliang Tang. "A Study on the P3P Problem", Lecture Notes in Computer Science, 2008

Crossref

6 words — < 1%

EXCLUDE QUOTES

OFF

EXCLUDE BIBLIOGRAPHY

ON

EXCLUDE MATCHES

OFF