Tanzania Journal of Science 47(3): 1125-1137, 2021
ISSN 0856-1761, e-ISSN 2507-7961

Multi-Vehicle Speed Estimation Algorithm Based on Real-Time Inter-Frame Tracking Technique

Ernest Kisingo[1], Ndyetabura Hamisi[1], Hashim U. Iddi[2]* and Baraka J. Maiseli[2]
[1]*Department of Computer Science and Engineering, College of Information and Communication Technologies, University of Dar es Salaam, P. O. Box 33335, Dar es Salaam, Tanzania.*
[2]*Department of Electronics and Telecommunications Engineering, College of Information and Communication Technologies, University of Dar es Salaam, P. O. Box 33335, Dar es Salaam, Tanzania.*
**Corresponding author, e-mail: ernestkisingo@yahoo.com, hashimuledi@udsm.ac.tz, hashimuledi@gmail.com*
Co-authors, e-mails: yhamisi@gmail.com; barakaezra@udsm.ac.tz
Received 1 Apr 2021, Revised 24 Jul 2021, Accepted 28 Jul 2021, Published Aug 2021
DOI: https://dx.doi.org/10.4314/tjs.v47i3.22

Abstract

Inappropriate vehicle speeding remains a central factor that causes road accidents claiming millions of lives every year. This challenge has raised concerns for vehicle speed estimation as an attempt to promote speed enforcement methods. Traditionally, radar and lidar systems have widely been used for this purpose, despite their several shortfalls: cosine error effects, need for direct line-of-sight, and inability to simultaneously and accurately measure speed from multiple vehicles. The current work proposes an algorithm and a multi-vehicle speed estimation system in a multi-lane road environment to address multi-vehicle speed estimation shortfalls. The proposed solution exploits image processing and computer vision techniques to flag vehicles with inappropriate speeding patterns. A series of experiments showed that the developed system generates more accurate results than those given by the lidar system. In essence, the proposed system can estimate the speed of up to six vehicles concurrently. It can produce an average percentage error of 2.7% relative to the actual speed measured by a speedometer. This error is 5.4% lower than that demonstrated by the lidar system, emphasizing that the proposed system may be a more suitable approach to traffic laws enforcement.

Keywords: Computer vision, image processing, inter-frame differencing, vehicle tracking, road accident.

Introduction

Road accidents remain a major cause of death, injury, and disability for children and young adults aged 5-29 (WHO 2018). This energetic group plays fundamental roles in building any country's economy, hence necessitating the deployment of more effective strategies to minimize road accidents that claim their lives. High-income and low-income countries altogether have, for decades, been striving to formulate strict laws and operationalize effective enforcement methods to reduce road accident injuries that may lead to deaths or disability (Ralaidovy et al. 2018).

A report from the World Health Organization (WHO) on road safety showed speed as a key risk factor that predominantly causes road accidents (WHO 2018). To address this challenge, the WHO recommends the establishment of law enforcement campaigns to ensure compliance of drivers with speed limits set by authorities. This recommendation calls for huge investments of enforcement methods to detect and punish drivers that violate speed limit laws. Therefore, both manual and automatic methods have been adopted by different countries to minimize the risks of road accidents due to overspeeding (Tang et al. 2020, Yu et al. 2020).

Classical enforcement methods for vehicle speed detection (and measurements) that most countries have adopted include those based on lidar (light detection and ranging) and radar (radio detection and ranging) technologies. Despite their relatively lower initial costs, these methods suffer from some weaknesses that may affect law enforcement's primary goal in road traffic. Both lidar and radar speed detectors handheld devices manually controlled by trained personnel require perfect direct line-of-sight from the incoming vehicle to generate more accurate measurements. This requirement can hardly be achieved. Consequently, these detectors tend to introduce cosine error effects into the measurements (Mandava et al. 2018). Furthermore, readings from lidar and radar systems may be affected by environmental conditions, traffic density, the battery power of the device, and distance of a vehicle from the device, all of which cannot be controlled with certainty (Carmer and Peterson 1996, Mullen and Contarino 2000). Even more challenging, lidar and radar systems cannot efficiently capture the speed of multiple vehicles moving on the road (e.g., highway). Such systems cannot accurately measure speeds below specific limits (Wallace et al. 2020).

Recently, computer vision and image processing techniques have widely been proposed to detect vehicle speed (Shahverdy et al. 2020, Sonth et al. 2020, Zhang et al. 2020). These techniques offer additional advantages over the traditional ones: little or no human intervention, capability for multiple vehicle speed measurements, higher accuracy, and insensitivity against line-of-sight (Fernández et al. 2021). Therefore, motivated by these advantages, we have developed a vision-driven algorithm and system for detecting and measuring speeds from multiple vehicles. The proposed algorithm exploits the divide-and-conquer approach, where smaller sub-problem instances derived from the overall problem were formulated and logically combined for speed estimation. The system, designed using Unified Modelling Language, integrates software, hardware, and networking elements. Using speedometer recordings on moving vehicles as reference and considering the lidar speed detector as a control experiment, we observed that our system produces promising speed measurements with errors significantly minimized relative to the lidar device's measurements (Zhao et al. 2019). This observation provides a convincing argument that the proposed system may be more suitable for law enforcement in road traffic.

Considering the advances in vehicle speed estimation, the issue of simultaneous speed measurements from multiple vehicles has not been satisfactorily addressed. In a recent work by Fernández et al. (2021), the authors gave a limitation of the previous methods that they cannot effectively incorporate data of multiple vehicles moving in multiple lanes. Consequently, the speed-measuring devices provide limited information of speed measurements, thus affecting informed decisions derived from law enforcement policies. This work capitalizes on the limitation to build a system that can estimate speeds of multiple vehicles.

Materials and Methods

Proposed method for multi-vehicle speed estimation

Algorithm development

In this research, we have proposed a more effective method that can estimate speed of multiple vehicles. The method, which can work even in a multi-lane environment, can be summarized into five blocks (Figure 1): image frame extraction, moving vehicle detection, inter-frame vehicle tracking, vehicle speed estimation, and speed limit processing. These logically-organized blocks interact with one another for accurate measurement of vehicle speed. Our method receives three inputs, namely video sequences, calibration data, and speed data.

Image frame extraction

This initial step involves the extraction of image frames from a pre-recorded video file saved in the computer. The method uses the recording camera's intrinsic characteristics (such as video frame rate, colour format, and frame size) as calibration data to decide how image frames should be extracted. Furthermore, the smooth median filter (Kawala-Sterniuk et al. 2020) with a kernel size of three was invoked to suppress the frames' noise. We obtained this kernel size empirically through experimentation under conditions that the filtered frames should have minimum noise and well-preserved structures (critical image features, such as edges and contours). Finally, the frames were converted into a grayscale colour mode to simplify mathematics: as per our application, processing coloured or grayscale images makes no difference to the final result (Overspeed frame).

Equation (1) can summarize the image frame extraction step;

$$I_t = \Gamma\{\text{median}(\mathcal{H}(I), \Omega)\}, \qquad (1)$$

where I denotes the video sequence, $\mathcal{H}$ denotes a sampling operator for extracting the video sequences at pre-defined intervals governed by the calibration data, Ω is the kernel size of the median filter, Γ denotes the operator for converting a filtered frame into grayscale, and I_t is the output frame at a time, t.

Moving vehicle detection

Image frames contain vehicles and other objects (e.g., road signs, pedestrians, trees, and buildings) not required by our method for processing. Therefore, this step focuses on detecting moving vehicles from the frames and ensures that all unrequired objects are eliminated. The assumption is that only vehicles move relative to the objects, meaning that at this stage the method disregards a situation where both vehicles and objects move simultaneously. Despite several methods to achieve this goal, we opted for the inter-frame differencing method because of its lower computational load, ease of implementation, and higher performance (Sun et al. 2020).

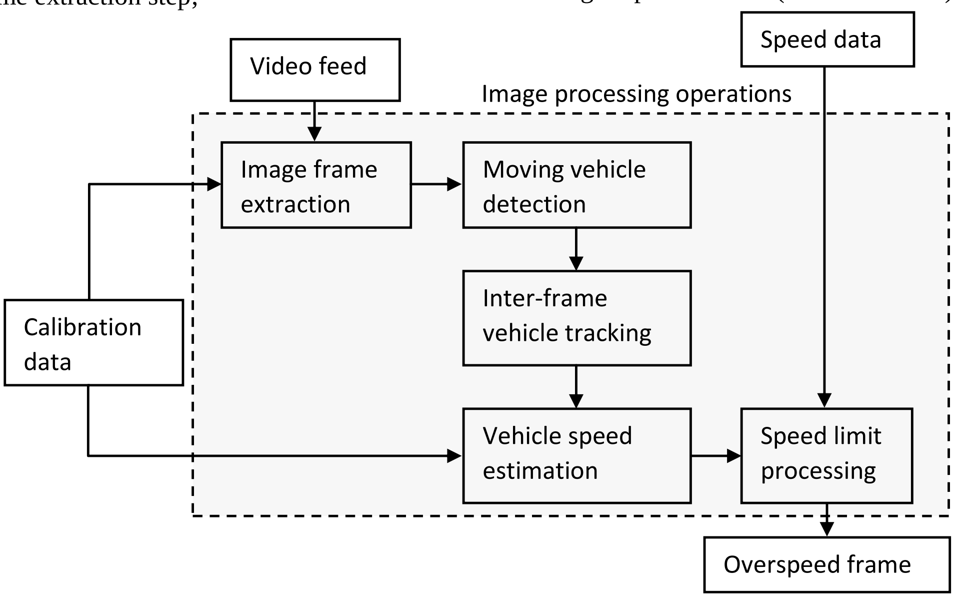

Figure 1: Block diagram of the proposed multi-vehicle speed estimation algorithm.

The method operates by subtracting adjacent frames to generate a frame with only moving pixels (blobs) representing moving objects. Given the complex nature and dynamics of the road traffic, there could be conditions when both vehicles and other unwanted objects moving in the frames' background thus resulting in noisy blobs. To address this challenge, thresholding techniques were applied to remove possible unwanted pixels resulting from the background dynamics (Ramakrishna and Kohir 2021, Touil et al. 2020). More specifically, the morphological operation called *erosion* was applied to eliminate holes that the previous process generated. Next, each blob received an identifier (integer) or BlobID with information on moving vehicles, such as centroid (two-dimensional coordinates representing the vehicle's center in the image) that was used in the subsequent steps for tracking and distance calculation.

Inter-frame vehicle tracking

Vehicle tracking involves finding movement patterns of a vehicle in the current frame relative to the previous one. The central question that this procedure address is whether consecutive image frames contain the same vehicle. Hence, we applied the Mahalanobis matching algorithm (Mei et al. 2016), defined as

$$d(\vec{x},\vec{y}) = \sqrt{(\vec{x}-\vec{y})^T\, S^{-1}(\vec{x}-\vec{y})}\,,$$

where $\vec{x}$ and $\vec{y}$ are two random vectors of the same distribution with covariance matrix S, and d denotes the Mahalanobis metric, which measures the distance between two points: centroid of a detected vehicle in the previous frame and distribution constituting all centroids of detected vehicles in the current frame. If S is the identity matrix, then the Mahalanobis distance reduces to the Euclidean distance; this condition implies that the vehicle between the frames is the same. In this case, matched vehicles in each frame are given the same BlobID. The process is repeated for consecutive frames such that the BlobID can be tracked over subsequent frames. For a different vehicle, S becomes a non-identity matrix, and hence the value of the BlobID is updated for this new vehicle.

Vehicle speed estimation

In this work, we define speed as the Euclidean distance covered by a vehicle per time. Hence, building on the inter-frame vehicle tracking step (Figure 1), vehicle distance between consecutive frames was obtained using the Euclidean distance formula

$$\text{Distance}(\boldsymbol{x},\boldsymbol{y}) = \sqrt{(x_2-x_1)^2+(y_2-y_1)^2}, \tag{2}$$

where $\boldsymbol{x}$ denotes the cartesian coordinate, (x_1, y_1), of the vehicle's centroid in the previous frame and $\boldsymbol{y}$ denotes the cartesian coordinate, (x_2, y_2), of the matched-vehicle's centroid in the current frame. The computed distance was then mapped into the real-world (actual) distance using the calibration data.

The time (in seconds) for the vehicle to cover the distance defined in Equation (2) was obtained using the video property, frame rate (or frame frequency, and expressed in frames per second). Hence,

$$\text{Time} = \frac{N}{\text{Frame Rate}}, \tag{3}$$

where N stands for the number of frames separating the centroids. Using Equations (2) and (3), the real-world speed of a vehicle (in kilometres per hour or km/hr) can be expressed as

$$\text{Speed} = K\,\frac{\sqrt{(x_2-x_1)^2+(y_2-y_1)^2}\times \text{Frame Rate}\times 3600}{N}, \tag{4}$$

where K represents the distance mapping factor in kilometres. Equation (4) shows that

speed and K are proportional: higher value of K translates to a wider Region of Interest, implying that more vehicles may be accommodated and processed simultaneously. For a camera fixed at a specific location on the road, K and frame Rate remains constant. Recalling Equation (4), therefore, the maximum vehicle speed can be observed when $N = 1$, noting that $N \in \mathbb{Z}^{+}$. Finally, speeds of vehicles in all frames are computed, and the values are passed to the last block, Speed limit processing (Figure 1).

Speed limit processing

Depending upon the road traffic enforcement laws for a specific road segment, recommended speed data are loaded into the system to determine vehicles that violate speed limits. For any vehicle in the current frame that violates the constraints, a red bounding rectangle with the corresponding speed value is drawn around a vehicle to provide a mark for identification. Then, this frame with the overspeed vehicle is saved and transferred through a File Transfer Protocol (FTP) to a central server for evidence keeping and further processing (e.g., manual or automated number plate recognition, offence and fine ticketing, and jurisdiction activities). However, implementation of these subsequent actions after retrieval of the overspeed vehicle lands outside the scope of this work.

System development

The two-dimensional video feeds to the proposed system represent three-dimensional coordinates of objects (vehicles) in the real-world environment. Therefore, implementation of the system considered the following assumptions: camera parameters, including frame rate and image size, are known; vehicle movements are linear on a flat ground plane; and, the relationship between image distance (in pixels) and coordinate distance on the ground is linear, meaning that the known length of pixels (pixel line) could safely be mapped to the ground without losing perspective. Development of the multi-vehicle speed estimation system followed the Object-Oriented System Design (OOSD) approach (Gessford 1992), which requires four steps: system analysis, system design, system implementation, and system testing.

System analysis

In this step, the Use Case diagram was used to identify actors and Use Cases, and to determine how these constructs interact (Figure 2). The Use Case diagram, in addition, shows different associations of the use cases. From the proposed algorithm (Figure 3), three actors were identified: operator, responsible for calibration of data entry; video source (pre-recorded video file), which generates video stream that the system can read and process; and central server, which stores image files generated from the system for further processing.

Furthermore, seven Use Case functions were identified: enter calibration data, validate user inputs, save user data into files, read video feed, determine vehicle speed, generate image file, and send image file. The first function (enter calibration data) provides an interface that allows the operator to interact with the system by entering station and calibration information. This function includes a feature to validate inputs of the operator based on the business rules. The second function (validate user inputs) deals with validation of station information and calibration data entered by the operator. The function uses pre-defined business rules, such as input format definition, to evaluate accuracy of user inputs. The next function (save user data into files) results when the previous function (validate user input) evaluates to success. This function facilitates saving of calibration data into a file, which may be retried by the system to compute vehicle speed. The fourth function (read video feed) connects with video sources, reads video stream, and converts it to the format suitable for vehicle speed determination. The fifth function (determine vehicle speed) gives our system's central goal to estimate vehicle speed using image processing and computer vision techniques. In the event of speed violation, this function is activated to generate an image file with a vehicle that has exceeded speed limits. The sixth function (generate image file) gets the current frame and converts it to the human readable format, then adds useful information to the converted file before saving it as an image. The last function (generate image file) interacts with the central server to deliver the image for further processing.

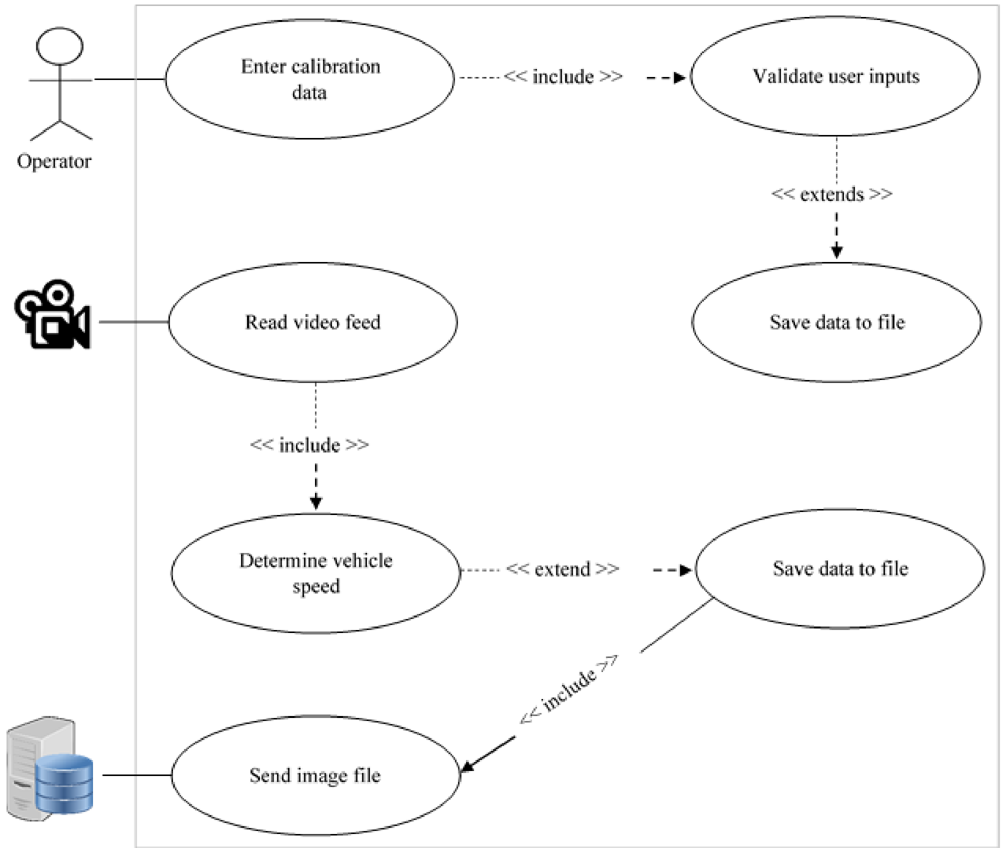

Figure 2: Use Case diagram of the system.

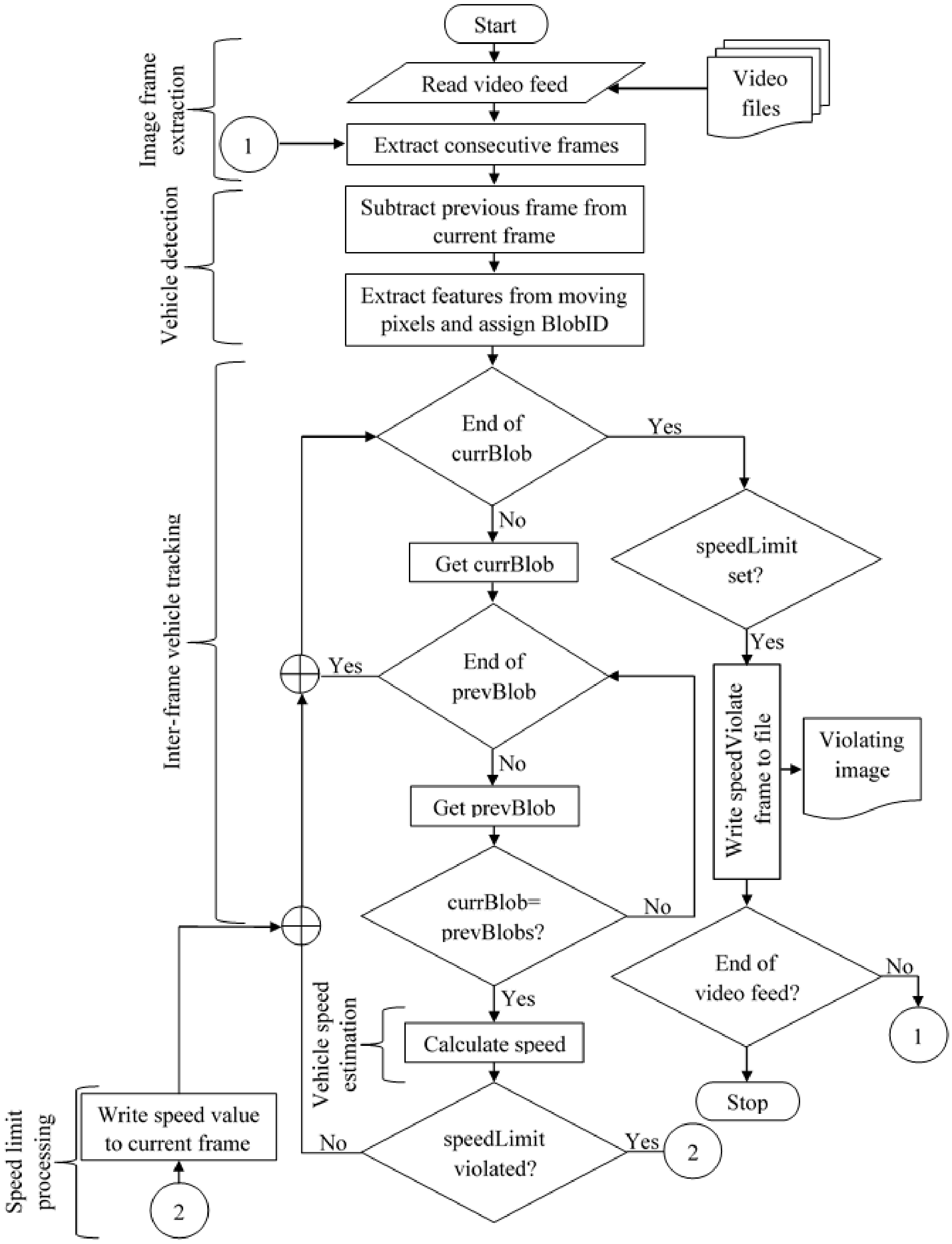

Figure 3: Proposed algorithm for multi-vehicle speed estimation.

System design

Building on the system analysis step's conceptual model, we designed the object-oriented model that maps technology-independent concepts into classes and pre-identified constraints. The system architecture, derived from system analysis, uses the client-server model, where clients present an onsite computational system that implements the algorithm for vehicle speed estimation. On the other hand, the server reflects a central system for the file repository required in subsequent processing stages. When the client detects speed violation, the system captures an image of a vehicle that has violated speed constraints, stores the image locally, and then sends that image to the central server using the file transfer protocol. Figure 4 shows the high-level architecture of the system.

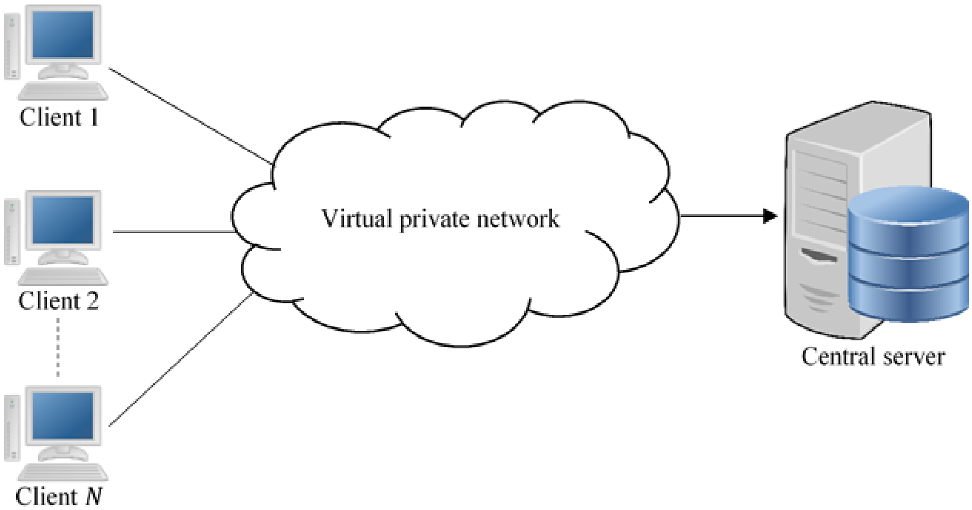

Figure 4: Physical architecture of the system.

System implementation

This OOSD stage comprises two implementation parts, namely client and server applications. These sides of the system were implemented using rapid application development tools and techniques. The server, for instance, implements FTP application with an open-source application, FileZilla. To ensure the system's effective operation, we configured user privileges, security, and login conditions to allow file transfer between server and client.

System testing

Development of the system was concluded by evaluating its functionalities and performance with respect to our work's overall goal. Therefore, modular and integration techniques were used to accomplish system testing (Figure 5). The former technique, for instance, was used to test the inter-frame vehicle tracking component (Figure 1) by previewing and analyzing the resulting image after subtraction of consecutive frames followed by application of the morphological erosion operation.

(a) Current image frame (b) Previous image frame (c) Difference image frame

Figure 5: Modular system testing of the inter-frame vehicle tracking component.

Results and Discussions

Our system's field test results were collected, analyzed, and compared with those recorded by the actual readings of the vehicle's speedometer (considered as reference) and the traditional lidar system. The hypothesis is that, at higher vehicle speeds, the proposed system's results should have minimum errors relative to those depicted by the speedometer. Performance evaluation results were computed as follows (Fernández et al. 2021):

$$\%\ \text{Error} = \frac{|\text{Speedometer reading} - \text{Our reading}|}{\text{Speedometer reading}} \times 100.$$

Figure 6 shows an estimation of vehicle speeds without skipping frames in the video feed. Results from this Figure highlight that our curve closely tracks the ideal curve generated by measurements from the speedometer. Although the lidar system performs relatively well in this case, weighing the competitive advantages of the proposed system, such as multi-vehicle speed measurement capability, suggests that our approach provides significant merits under complex road traffic conditions. However, a note should be taken that frame skipping improves our system's accuracy, as demonstrated in the subsequent sections, and, in this experiment (Figure 6), vehicle speed estimation has been achieved without considering frame skipping.

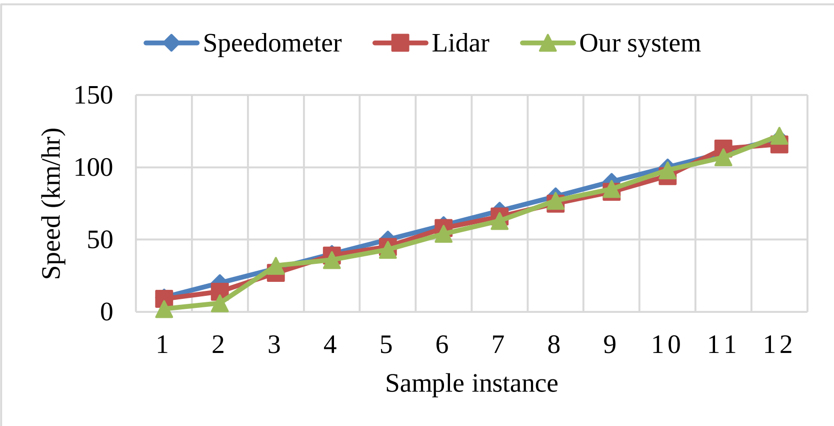

Figure 6: Vehicle speed estimation without frame skipping.

Using a frame skipping value of two, our system generates more accurate results at higher vehicle speeds (above 50 km/hr) than those demonstrated by the lidar system (Figure 7). Skipping some frames means minimization of errors caused by overlapping of events in the frames, which improves accuracy significantly. However, at lower speeds (below 50 km/hr), our system demonstrates relatively lower accuracy when compared with the lidar system. This observation was expected as it emanates from the intrinsic characteristics of the inter-frame differencing process for detecting a vehicle in the image frame. Future works may consider other more accurate vehicle detection approaches, such as those based on deep convolutional neural networks (Song et al. 2019).

Increasing the frame-skipping number to three, our system generates an average percentage error of 2.7%, which is 5.4% lower than that depicted by the lidar system (Figure 8). This interesting achievement implies that image processing techniques may be more appropriate in vehicle speed estimation. Such techniques to deal with varying road traffic conditions may be important to construct effective enforcement laws.

Because our system produces more accurate results for a frame-skipping value of three, we used this setting to conduct another series of experiments to estimate speed for different vehicle types. These experiments were carried out in a multi-lane road, where a camera was configured to face the road at a downward tilt angle of 25°. The camera's frame rate was 30 frames/second, and the dimension for each image frame was 640 x 360 pixels. Results showed that the two systems generate comparable results (Figures 9 and 10), thus signaling the possible successful deployment of the proposed system.

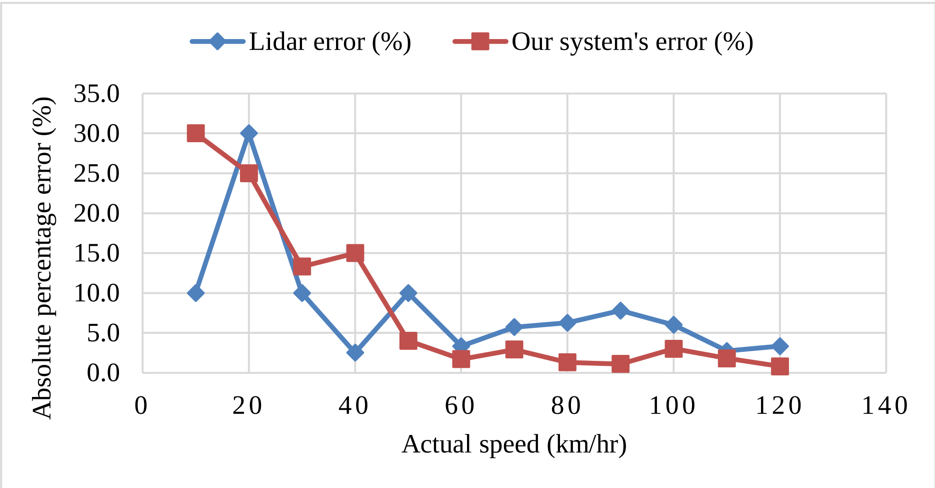

Figure 7: Percentage errors of lidar and proposed systems with frame skipping equals two.

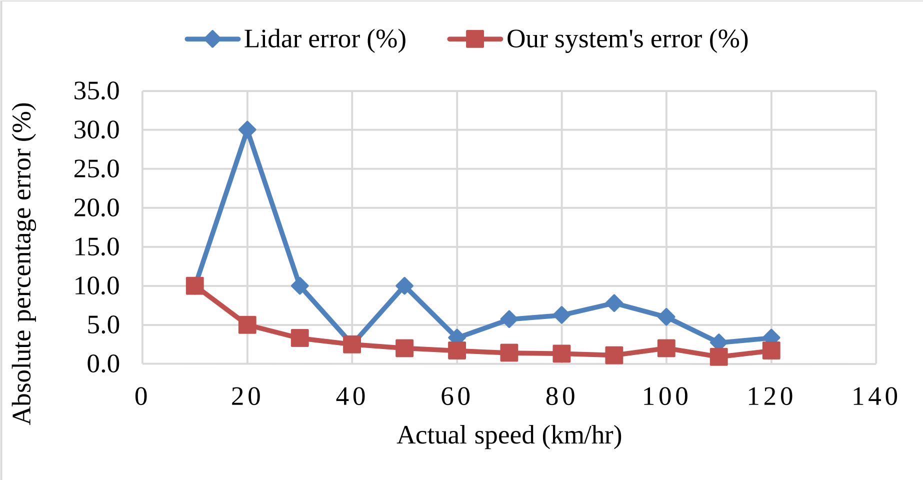

Figure 8: Percentage errors of lidar and proposed systems with frame skipping equals three.

Figure 9: Estimated vehicle speeds using the proposed system.

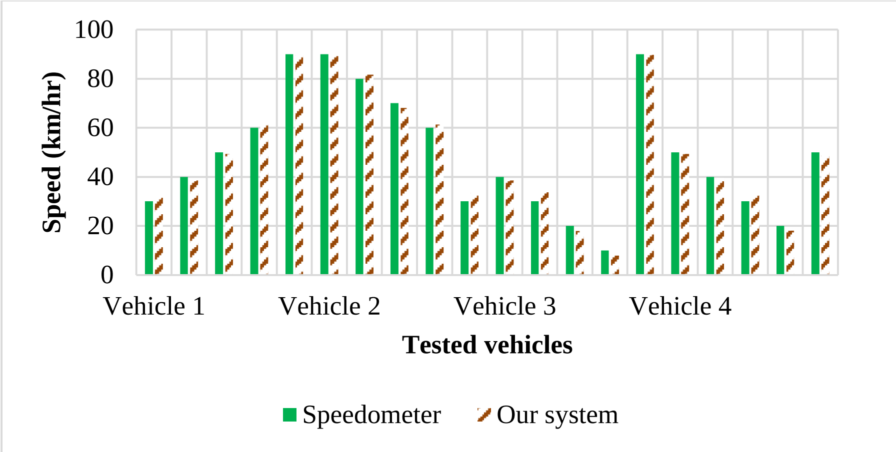

Figure 10: Speed measurements between ideal and proposed systems for different vehicles.

In all the experiments, the maximum test speed was limited to 120 km/hr for safety reasons, considering the road conditions and the type of test vehicles used. Therefore, the performance of our system may be affected at speeds higher than 120 km/hr. Given the results' trends, however, it would be reasonable to expect that the proposed system would produce an outstanding performance at much higher speeds.

Conclusion

This work designed and developed an algorithm and a multi-speed vehicle estimation system using image processing and computer vision techniques. The developed system can simultaneously estimate four vehicles' speed, giving an average percentage error of 5.4% lower than that of the lidar system. Our system's accuracy increases with vehicle speed, hence making the system more suitable in enforcing laws that provide legal punishments for vehicles that exceed specified (high) speed limits. The proposed system, which can give real-time speed for multiple vehicles, offers high immunity against the environment's interferences. This potential advantage addresses the limitation of existing approaches. As a future research opportunity, we recommend scholars to develop alternative calibration methods that can dynamically map ground and image distances. Our system exploited a pixel-line approach that demands fixation of the calibrated ground distances. Hence, recalibration was necessary each time the camera was moved to other road locations.

Acknowledgement and declaration

The authors of this work declare that they have no conflict of interest to disclose.

References

Carmer DC and Peterson LM 1996 Laser radar in robotics. *Proc. IEEE*. 84(2): 299–320.

Fernández D, Hernández A, and García I 2021 Vision-based vehicle speed estimation: A survey. *IET Intell. Transp. Syst*. DOI: 10.1049/itr2.12079

Gessford JE 1992 Object-Oriented System Design. *J. Database Manage. (JDM)* 3(4): 27–37.

Kawala-Sterniuk A, Podpora M, Pelc M, Blaszczyszyn M, Gorzelanczyk EJ, Martinek R and Ozana S 2020 Comparison of smoothing filters in analysis of EEG data for the medical diagnostics purposes. *Sensors* 20(3): 807.

Mandava M, Gammenthaler RS, and Hocker SF 2018 Vehicle speed enforcement using absolute speed handheld lidar. In *IEEE Vehicular Technology Conference* Vol. 2018-August. Institute of Electrical and

Electronics Engineers Inc.

Mei J, Liu M, Wang YF and Gao H 2016 Learning a mahalanobis distance-based dynamic time warping measure for multivariate time series classification. *IEEE Trans. Cybern*. 46(6): 1363–1374.

Mullen LJ and Contarino VM 2000 Hybrid lidar-radar: seeing through the scatter. *IEEE Microwave Mag*. 1(3): 42–48.

Ralaidovy AH, Bachani AM, Lauer JA, Lai T and Chisholm D 2018 Cost-effectiveness of strategies to prevent road traffic injuries in eastern sub-Saharan Africa and Southeast Asia: New results from WHO-CHOICE. *Cost Eff. Resour. Alloc.* 16(1): 1–10.

Ramakrishna N and Kohir VV 2021 Diabetic retinopathy detection at early stage using a set of morphological operations. In: *Intelligent Computing in Control and Communication* (pp. 467-478). Springer, Singapore.

Shahverdy M, Fathy M, Berangi R and Sabokrou M 2020 Driver behavior detection and classification using deep convolutional neural networks. *Expert Systems with Applications* 149: 113240.

Song K, Yang H and Yin Z 2019 Multi-Scale Attention Deep Neural Network for Fast Accurate Object Detection. *IEEE Trans. Circuits Syst. Video Technol*. 29(10): 2972–2985.

Sonth A, Settibhaktini H and Jahagirdar A 2020 Vehicle speed determination and license plate localization from monocular video streams. In: *Adv. Intell. Systems and Computing* Vol. 1022 AISC.

Sun W, Du H, Ma G, Shi S, Zhang X and Wu Y 2020 Moving vehicle video detection combining ViBe and inter-frame difference. *Int. J. Embedded Syst*. 12(3): 371–379.

Tang F, Gao F and Wang Z 2020 Driving capability-based transition strategy for cooperative driving: from manual to automatic. *IEEE Access* 8: 139013–139022.

Touil A, Kalti K, Conze PH, Solaiman B and Mahjoub MA 2020 Automatic detection of microcalcification based on morphological operations and structural similarity indices. *Biocybern. Biomed. Eng.* 40(3): 1155–1173.

Wallace AM, Halimi A and Buller GS 2020 Full waveform LiDAR for adverse weather conditions. *IEEE Trans. Veh. Technol.* 69(7): 7064–7077.

WHO (World Health Organization) 2018 *Global status report on road safety 2018: Summary*.

Yu K, Lin L, Alazab M, Tan L and Gu B 2020 Deep learning-based traffic safety solution for a mixture of autonomous and manual vehicles in a 5G-enabled intelligent transportation system. *IEEE Trans. Intell. Transp. Syst.* 22(7): 4337-4347.

Zhang J, Xiao W, Coifman B and Mills JP 2020 Vehicle tracking and speed estimation from roadside lidar. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens* . 13: 5597-5608.

Zhao J, Xu H, Liu H, Wu J, Zheng Y and Wu D 2019 Detection and tracking of pedestrians and vehicles using roadside LiDAR sensors. *Transp. Res. Part C: Emerg. Technol.* 100: 68-87.