

Retraction

Retracted: A Novel Traffic Surveillance System Using an Uncalibrated Camera

Journal of Advanced Transportation

Received 8 August 2023; Accepted 8 August 2023; Published 9 August 2023

Copyright © 2023 Journal of Advanced Transportation. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

- (1) Discrepancies in scope
- (2) Discrepancies in the description of the research reported
- (3) Discrepancies between the availability of data and the research described
- (4) Inappropriate citations
- (5) Incoherent, meaningless and/or irrelevant content included in the article
- (6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

- [1] S. Gupta, R. S. Kartha, E. Ramirez-Asis, J. Flores-Albornoz, R. Vilchez-Vásquez, and S. K. Singh, "A Novel Traffic Surveillance System Using an Uncalibrated Camera," *Journal of Advanced Transportation*, vol. 2022, Article ID 5867524, 10 pages, 2022.

Research Article

A Novel Traffic Surveillance System Using an Uncalibrated Camera

Sandeep Gupta ¹, Ranju S Kartha ², Edwin Ramirez-Asis ³, Judith Flores-Albornoz ³, Rosa Vélchez-Vásquez ⁴, and Sitesh Kumar Singh ⁵

¹Engineering Department, JECRC University, Jaipur, India

²Science and Engineering Department, Mangalam College of Engineering, Ettumanoor, Kottayam, Kerala, India

³Universidad Nacional Santiago Antunez de Mayolo, Huaraz, Peru

⁴Computer Science and Engineering, Universidad Nacional Santiago Antunez de Mayolo, Huaraz, Peru

⁵Department of Civil Engineering, Wollega University, Nekemte, Oromia, Ethiopia

Correspondence should be addressed to Sitesh Kumar Singh; sitesh@wollegauniversity.edu.et

Received 11 January 2022; Revised 25 January 2022; Accepted 29 January 2022; Published 23 February 2022

Academic Editor: Muhammad Arif

Copyright © 2022 Sandeep Gupta et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The objective of this paper is to present an effective and reliable method for the traffic surveillance using the concepts of digital image processing. The paper proposes a system that can detect, track, and estimate velocity of vehicles using an uncalibrated camera and also detect and recognize their registration number plate. This approach provides a cost-effective alternative for traffic flow monitoring and surveillance. This robust system finds its applications in urban traffic management systems, military installation, and research facility security systems. This approach is a computationally efficient approach for detecting and tracking moving cars on the road utilizing uncalibrated cameras mounted on the road. It is also helpful for military installation because all the security issues have been detected by using this approach.

1. Introduction

Effective monitoring of traffic has now become a very integral part of traffic management as the number of vehicles on the road is increasing every day. If traffic management authorities could analyse the flow of the traffic using average velocity of all the vehicles passing through each road with respect to time, then it would serve as an effective tool for them to manage traffic, and hence, large traffic jams could be avoided [1, 2]. This paper aims to propose a method that can detect and track the vehicles moving on the road using uncalibrated cameras installed on the roads in a computationally effective manner. Words ‘vehicles’ and ‘cars’ are used interchangeably throughout the paper, and they are intended to mean the same thing. This system will count the number of cars present on the road and their average speed at any particular instance. Furthermore, speed of each car along with its number plate will be determined, and its record will be maintained. This is done by developing a method that calibrates the road to map pixels in the frame to

actual distance on the ground. A system will be designed such that if speed of the car is above a certain limit, E-challan would be fined. The record of cars which passed through the road can serve as an effective database to locate a car in case of car theft, hit-and-run case, monitoring the movement of a particular suspect by police, etc. The scope of the traffic surveillance system which uses uncalibrated cameras is that it provides an effective and substantially less cost-effective way for traffic monitoring on the road, which is a critical demand in developing nations. Today, many research works are ongoing in the traffic surveillance system area. Sochor et al. [3] presented the analysis of an automatic calibration method based on statistics of dimensions. In this paper, a comprehensive dataset for pure visual speed measurement with a monocular camera is made and compared with various existing methods. Vanishing point detection is proposed for an automatic scene scale inference by alignment of the 3D model bounding box [4, 5]. A review of various methods of video-based intelligent transportation systems includes not only present applications but also

future use cases by analysing the techniques of image processing and using their results for future requirements of traffic engineering [6–8]. Paper [9] method is proposed for motion analysis of the surveillance video to determine three states of traffic congestion, slow, and smooth [10]. A method of vehicles' classification based on pixel length of the detected vehicle after removing shadows and negative effects is taken care of from horizontal occlusion [11, 12]. A system in research paper [13] is proposed for real-time video-based velocity estimation by extracting moving cars by frame subtraction and then tracking [14]. The straightening technique is proposed and is used to remove perspective effects, and a correlation technique is used to establish the necessary scale factor as these cameras are not calibrated, and thereafter temporal correlation between sequential straightened frames of the video is used to make speed estimates. The suggested traffic surveillance system uses uncalibrated cameras to provide an effective and relatively low-cost approach for vehicle surveillance on the road, which is a critical demand in underdeveloped countries [15]. We have developed a method that creates an adaptive background model for each and every pixel of the RGB frame for the detection of vehicles. RGB stands for red, green, and blue, which are the main colors used in additive colour synthesis. An RGB file is made up of red, green, and blue composite layers.

Once the vehicle is detected, it is tracked using a path predicting and matching algorithm [16]. Then, the velocity of the vehicles being tracked is estimated by our algorithm based on road calibration. When the vehicle reaches an optimum distance from the view of the camera, its number plate is extracted using the maximally stable extremal region (MSER) and recognized by optical character recognition (OCR). The current system's restriction is partial and total vehicle occlusion, which we are actively working on using feature-based segmentation for partial occlusion.

2. Detection of Vehicles

Detection of cars is the first step in the surveillance system. The detection method plays an important role for changing light conditions. The background modelling is flexible to these changes and updates itself based on the user-specified learning rate, whereas the foreground intensities have no effect on the background model. To detect a moving object requires the knowledge of a model that can differentiate between foreground and background. A competent detection method should be adaptable to the changing light conditions and periodic changes in the colour intensities of the background. Our approach for background modelling is adaptable to these changes and updates itself according to the user-specified learning rate, and the background model is not affected by the foreground intensities. During training for each pixel in the RGB plane, the values of intensity for N (number of training frames) frames are stored as a matrix of $L \times B \times N$, where L is the width of the frame and B is the height; then, for each pixel, the standard deviation and mean are calculated:

$$\begin{aligned} \text{Mean R}(l, b) &= \frac{1}{N} \sum_{n=1}^N \text{ImgR}(l, b, n), \\ \text{Mean G}(l, b) &= \frac{1}{N} \sum_{n=1}^N \text{ImgG}(l, b, n), \\ \text{Mean B}(l, b) &= \frac{1}{N} \sum_{n=1}^N \text{ImgB}(l, b, n), \end{aligned} \quad (1)$$

where ImgR , ImgG , and ImgB are the red, green, and blue components of the N training frame images and $\text{Mean R}(l, b)$, $\text{Mean G}(l, b)$, and $\text{Mean B}(l, b)$ are the mean matrices of R, G, and B planes, respectively.

$$\begin{aligned} \text{StdR}(l, b) &= \sqrt{\sum_{n=1}^N \frac{(\text{ImgR}(l, b, n) - \text{MeanR}(l, b))^2}{N}}, \\ \text{StdG}(l, b) &= \sqrt{\sum_{n=1}^N \frac{(\text{ImgG}(l, b, n) - \text{MeanG}(l, b))^2}{N}}, \\ \text{StdB}(l, b) &= \sqrt{\sum_{n=1}^N \frac{(\text{ImgB}(l, b, n) - \text{MeanB}(l, b))^2}{N}}. \end{aligned} \quad (2)$$

And $\text{StdR}(l, b)$, $\text{StdG}(l, b)$, and $\text{StdB}(l, b)$ are the standard deviation matrix of the R, G, and B planes.

Now, when the $(N+1)^{\text{th}}$ RGB frame (Img_{N+1}) arrives, it is subtracted from mean of the created background model, and the decision that whether the pixels belong to the foreground or the background is done using the following criteria:

$$\begin{aligned} \Delta R &= |\text{ImgR}_{n+1}(l, b) - \text{MeanR}(l, b)|, \\ f\text{ImgR}(l, b) &= \begin{cases} 1, & \Delta R > \alpha * \text{StdR}(l, b), \\ 0, & \text{otherwise,} \end{cases} \\ \Delta G &= |\text{ImgG}_{n+1}(l, b) - \text{MeanG}(l, b)|, \\ f\text{ImgG}(l, b) &= \begin{cases} 1, & \Delta G > \alpha * \text{StdG}(l, b), \\ 0, & \text{otherwise,} \end{cases} \\ \Delta B &= |\text{ImgB}_{n+1}(l, b) - \text{MeanB}(l, b)|, \\ f\text{ImgB}(l, b) &= \begin{cases} 1, & \Delta B > \alpha * \text{StdB}(l, b), \\ 0, & \text{otherwise,} \end{cases} \end{aligned} \quad (3)$$

where α is usually between 3 and 6, ΔR , ΔG , ΔB is the difference of red, green, and blue planes of the $(N+1)^{\text{th}}$ frame with their respective mean matrices, and $f\text{ImgR}(l, b)$, $f\text{ImgG}(l, b)$, and $f\text{ImgB}(l, b)$ are the foreground matrices of the RGB plane, respectively; then, the resultant foreground matrix ($Rf\text{Img}(l, b)$) is formed by taking the union of all the foreground matrices of the RGB plane because a significant change in colour intensity even in one colour plane indicates the presence of an object.

$$RfImg(l, b) = fImgR(l, b) \cup fImgB(l, b) \cup fImgG(l, b). \quad (4)$$

The union of these three matrices is taken by adding them and assigning each element of the resultant matrix with 1 to those having value either equal to or greater than 1. Blob analysis is then performed on this binary resultant foreground matrix, and blobs having size smaller than a minimum specified area are removed. After performing this, each blob of this matrix represents the foreground or the moving object.

$$\begin{aligned} \text{MeanR}(l, b) &= \frac{1}{N} [(N - \mu) * (\text{MeanR}(l, b) * \sim RfImg(l, b) + \mu * \text{ImgR}_{n+1}(l, b) * \sim RfImg(l, b) + N * \text{MeanR}(l, b) * RfImg(l, b)], \\ \text{StdR}(l, b) &= [((1 - \mu)\text{StdR}(l, b) + \mu\Delta R) * \sim RfImg(l, b) + \text{StdR}(l, b) * RfImg(l, b)], \\ \text{MeanB}(l, b) &= \frac{1}{N} [(N - \mu) * (\text{MeanB}(l, b) * \sim RfImg(l, b) + \mu * \text{ImgB}_{n+1}(l, b) * \sim RfImg(l, b) + N * \text{MeanB}(l, b) * RfImg(l, b)], \\ \text{StdB}(l, b) &= [((1 - \mu)\text{StdB}(l, b) + \mu\Delta R) * \sim RfImg(l, b) + \text{StdB}(l, b) * RfImg(l, b)], \\ \text{MeanG}(l, b) &= \frac{1}{N} [(N - \mu) * (\text{MeanG}(l, b) * \sim RfImg(l, b) + \mu * \text{ImgG}_{n+1}(l, b) * \sim RfImg(l, b) + N * \text{MeanG}(l, b) * RfImg(l, b)], \\ \text{StdG}(l, b) &= [((1 - \mu)\text{StdG}(l, b) + \mu\Delta R) * \sim RfImg(l, b) + \text{StdG}(l, b) * RfImg(l, b)]. \end{aligned} \quad (5)$$

'*' is the element-wise multiplication, '~' is the logical complement of the binary foreground matrix, and μ is the learning rate, i.e., the rate at which we want the background model to adapt to the changes. Keeping a larger value of μ will make the background model to adapt at a higher rate.

Not including the foreground pixels while updating the mean and standard deviation greatly improves the detection of objects even if they stop for a while and then again move on. This improves computational efficiency as we do not need to model the foreground separately, i.e., it saves computation time which is really critical in real-time applications when we have to handle a large number of vehicles at the same time.

3. Tracking

Tracking is an integral part of the traffic surveillance system. Tracking the car once it has been detected can help in identifying and differentiating vehicles even if they are identical and far away from the view of the camera, i.e., when the registration plate number of the vehicle is not visible [16]. This paper proposes a tracking algorithm which uses centroid as the parameter to be tracked. It is chosen so because the changing size of the vehicle as it moves towards the camera has negligible effect on the position of the centroid with respect to the vehicle body [17]. The tracking algorithm uses a two-step process to track a detected car:

- (i) Closest Euclidean distance approach
- (ii) Prediction of position and matching approach

The background model thus created needs to be updated by the Img_{n+1} frame. Here, only those pixels' mean and standard deviation are updated which were not detected as foreground because updating a detected foreground pixel into the background model creates noisy results. Therefore, gradual changes and small sudden changes can be incorporated into the background model as shown in Figure 1. The mean and standard deviation matrix is updated by the following method:

3.1. Closest Euclidean Distance Approach. Initially, the first three frames after the vehicle detection or the three frames after the vehicle stops are being tracked and then the frames start moving again and, the tracking is based on the logic that the vehicle would be nearest to itself in the next frame than any other vehicle in the frame as there can be no abrupt changes in the path of the car (assuming frame rate to be high) [18]. The Euclidean length between the cluster centers of the first and second frames is computed, and the specific vehicle ID is moved to the center in the second frame that is closest to the centroid in the very first frame and within a minimal threshold distance. Whenever a vehicle is detected, a temporary ID number is given to that detection, and the physical features of the vehicle such as centroid coordinate, area, and frame counter are recorded and attached to the ID. When the next frame arrives, the Euclidean distance is calculated between the centroids of the first frame and the second frame, and whichever centroid in the second frame is nearest to the centroid in the first frame and within a minimum threshold distance, the particular vehicle ID is transferred to that centroid. Now, the velocity of the vehicle in pixel/sec can also be calculated as we know the time elapsed between the two frames [19].

$$\text{Vel}_{n,i}(x) = \frac{[C_{n,i}(x) - C_{n-1,i}(x)]}{T}, \quad (6)$$

$$\text{Vel}_{n,i}(y) = \frac{[C_{n,i}(y) - C_{n-1,i}(y)]}{T}, \quad (7)$$

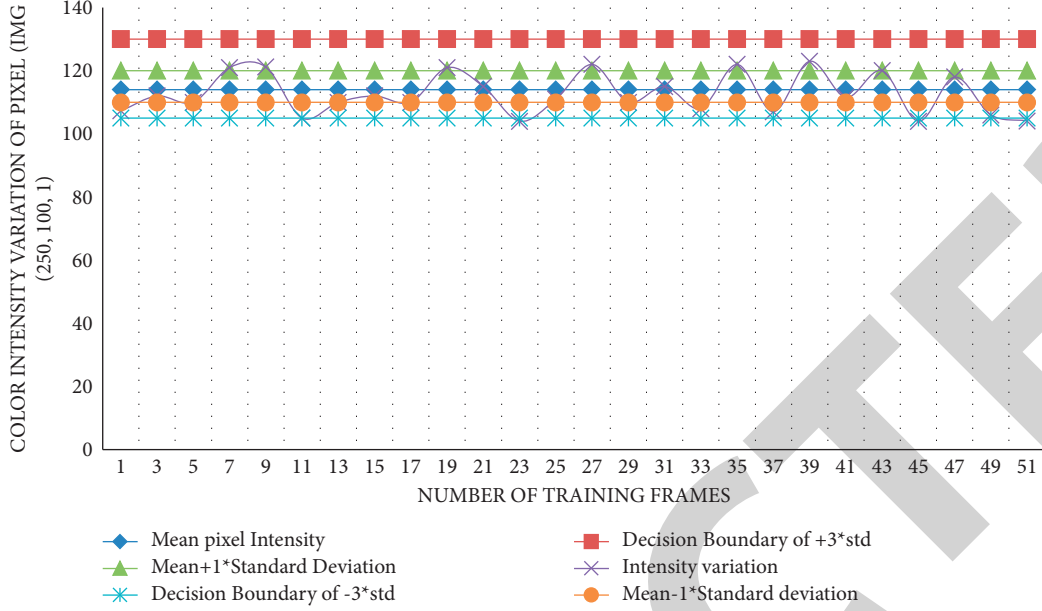


FIGURE 1: Graph representing the background model of pixels in different planes after training.

where $C_{n,i}(x, y)$ and $C_{n-1,i}(x, y)$ are the centroid coordinates of that vehicle (say ID no. i) in the n th frame and $(n-1)$ th frame, respectively, T is the time elapsed between the n th and $(n-1)$ th frame, and $Vel_{n,i}(x, y)$ is the velocity of that vehicle (say ID no. i) in pixels/sec in the n th frame.

This pixel velocity is also attached along with physical features of the vehicle to that ID. This process is repeated again when the third frame arrives. After this, we have sufficient information of the velocity of the centroids for the past two frames, so now, we will switch the tracking to the prediction and matching mode which will give better results. The results from equations (6) and (7) are shown in Figures 2 and 3, respectively.

3.2. Prediction of Position and Matching Approach. As the velocity of the past two frames is known, we will predict the position of the vehicle in the next frame. When the next frame arrives, again the detection of the vehicles is performed. The vehicle whose centroid lies nearest to the predicted position within minimum threshold distance (to accommodate the error in prediction) is given the same ID as it was in the previous frame. The position is predicted along the two axes x and y using the nonlinear equation with respect to time.

$$Pp_{n+1,i}(x) = Vel_{n,i}(x) * T + 0.5 * \frac{(Vel_{n,i}(x) - Vel_{n-1,i}(x))}{T'} * T^2,$$

$$Pp_{n+1,i}(y) = Vel_{n,i}(y) * T + 0.5 * \frac{(Vel_{n,i}(y) - Vel_{n-1,i}(y))}{T'} * T^2. \quad (8)$$

$Pp_{n+1,i}(x, y)$ is the predicted position of the vehicle ID no. i in the $(n+1)$ th frame, $Vel_{n,i}(x)$ and $Vel_{n-1,i}(x)$ are the velocity of that vehicle (ID no. i) in the n th frame and $(n-1)$

th frame, T' is the time elapsed between the n th and $(n-1)$ th frame, and T is the time elapsed between the n th and $(n+1)$ th frame.

3.3. Track Deletion Algorithm. It is an integral part of tracking dynamic objects because maintaining the track for those cars which have left the frame is unnecessary as it just wastes computation time, and in case of momentary occlusion of the car, the track should not be lost. So, on loss of track, the algorithm continues to predict the position of cars based on the previous frame data of velocity for 5 consecutive frames. Tracking algorithms may forecast the future position of several moving objects based on the history of actual roles supplied by sensor systems. If the track still remains lost, then it deletes the record of it.

4. Velocity Estimation

As the objective of this paper is to use an uncalibrated camera in order to provide a less expensive method of reliable speed estimation, it proposes a method to automatically calibrate the road and then use it in velocity estimation of the vehicle moving over it. As we know that the two parallel sides of the road in 3D appear to converge at a point in a 2D camera frame as shown in Figure 4, we can calculate the converging point (say x_0, y_0) if we can find the equation of the two parallel sides of the road in the frame (say $y = m_1 * x + c_1$ and $y = m_r * x + c_r$).

Suppose $C_1(x, y)$ is the centroid of a car in frame 1, and in the next frame, $C_2(x, y)$ is the centroid as shown in Figure 5. Now, if we join C_1 and C_2 with x_0, y_0 , then these two lines (say $y = m_1 * x + c_1$ and $y = m_2 * x + c_2$) would also be parallel in the real world.

Hence, to find distance travelled along the road in pixels, we can calculate distance along any of the four parallel lines

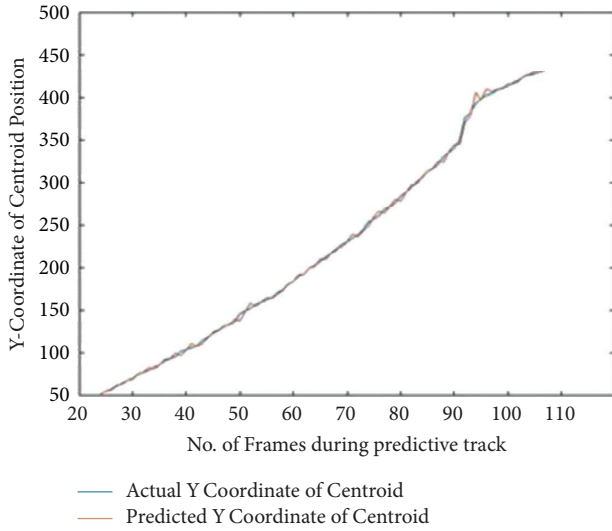


FIGURE 2: The difference between the predicted Y-coordinate and the actual Y-coordinate of the vehicles' centroid during predictive track.

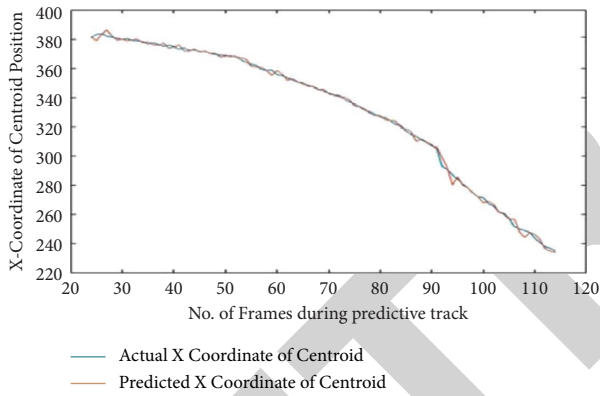


FIGURE 3: The difference between the predicted X-coordinate and the actual X-coordinate of the vehicles' centroid during predictive track.

($y = m_1 * x + c_1$, $y = m_r * x + c_r$, $y = m_1 * x + c_1$, and $y = m_2 * x + c_2$) between the y -coordinate of C_1 and C_2 . If we use $y = m_1 * x + c_1$, then at y_1 of C_1 , we get $x_1 = (y_1 - c_1) / m_1$, and similarly, at y_2 of C_1 , we get $x_2 = (y_2 - c_1) / m_1$; then, the Euclidean distance between two frames along the road is given by

$$\text{Distalongroad} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}. \quad (9)$$

To find distance travelled perpendicular to the road in pixels, we can find the distance between the line $y = m_1 * x + c_1$ and $y = m_2 * x + c_2$ at x_1 of C_1 or x_2 of C_2 by using

$$\text{Disthorizontal} = \left| \frac{x_1 - (y_1 - c_2)}{m_2} \right|. \quad (10)$$

Now, the challenge is to develop a method that can automatically calibrate the road. Circular-shaped bright white discs are kept along two edges of the road at equal

distance (say k units apart) as shown in Figure 6. These circular discs are kept temporarily, and after calibration, they will be removed.

So, when we run this algorithm, it first converts the RGB frame into a binary image giving the level of intensity above 90% of 255 (as the image is in the uint8 format) as 1 and below that 0. Now, we can segment our disc out of this binary image by using the eccentricity and area property of the circular disc, i.e., eccentricity of the circular disc will be less than 0.9 (as the circular disc may appear elliptical from the camera placed with view along the road), and area should be between a min and max value.

Once the circular discs are segmented as shown in Figures 7 and 8, we can use blob analysis to find out the centroids of these discs. Then, we use the left and right edge centroids to best fit two lines $y = m_l * x + c_l$ (for the left side of the road) and $y = m_r * x + c_r$ (for the right side of the road), respectively. After this, we find the x_0 and y_0 points by equating the two equations.

$$x_0 = \frac{cl - cr}{mr - ml} \quad (11)$$

$$y_0 = ml * x_0 + cl.$$

As the actual distance on the ground is known between each centroid (k units), this method fits a polynomial of order 2 using the centroids of circular discs which are on the left side of the road as shown in Figure 9.

Now, to find distance along the road, we use this graph to map the pixel distance (Distalongroad) in the frame to the actual ground distance. For the actual horizontal distance, the total width of the road is calculated at y_1 of C_1 in pixels by using

$$\text{Totalhorizdist} = \left| \frac{y_1 - cl}{ml} - \frac{y_1 - cr}{mr} \right|. \quad (12)$$

Actual total width of the road can be measured, say W_{Actual} ; then, the horizontal distance in pixels moved by the car between the two frames can be mapped to real-world distance by using the following equation:

$$\text{Realthordist} = \frac{\text{distorizontal}}{\text{Totalhorizdist}} * W_{\text{Actual}}. \quad (13)$$

The velocity along the road V_{parallel} can be calculated using the following formula: $\text{Realthordist} / \text{time elapsed}$ between frames. The velocity perpendicular to road V_{horiz} can be calculated using formula $\text{Realthordist} / \text{time elapsed}$ between two frames. The resultant velocity V_r is calculated:

$$V_r = \sqrt{(V_{\text{parallel}}^2 + V_{\text{horiz}}^2)}. \quad (14)$$

As the camera view is along the road and each car is being tracked, the velocity of the incoming car can be monitored till a larger distance, and issues of rash driving and overspeeding can be brought under control via e-ticketing. The average speed of all the cars passing through can serve as an effective tool to manage traffic conditions.

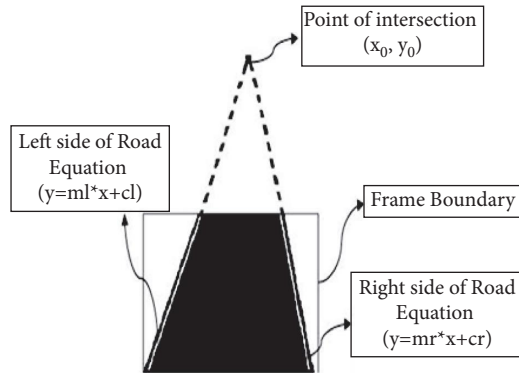


FIGURE 4: Illustration of the projection of parallel edges of the road backward to obtain the point of intersection.

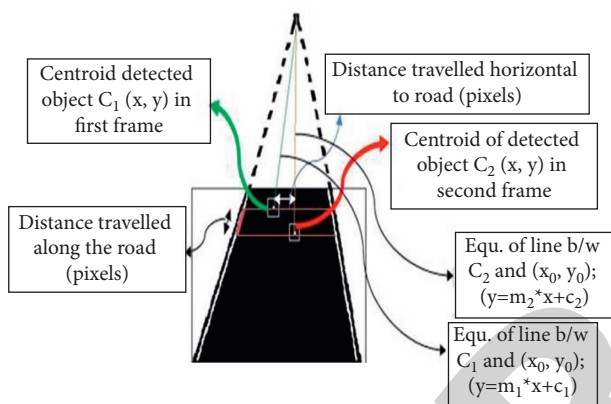


FIGURE 5: Explanation of distance travelled along and horizontal to the road.

5. Number Plate Recognition

The car, once detected, would be tracked, and its velocity would be monitored. As the vehicle approaches towards the camera, at an optimum distance, this method would start detection and recognition of the registration number plate using the following algorithm as shown in Figure 10.

The steps for number plate extraction which we followed in this work are given as follows:

- (1) Detected car/vehicle image is first cropped to its lower half as the number plate is usually in the lower half part of the image as shown in Figure 11.
- (2) The edges in the image where the change in the colour intensity (frequency) is high are enhanced by high-frequency emphasis by adding a high-pass filtered version of the image with the original image.

The number plate detection is done by extracting the MSER which is a method of blob detection as shown in Figure 12. It is a very effective and well-known method for text detection in an image. It is based on the idea of taking regions which stay nearly the same through a wide range of thresholds within the maximum and minimum threshold area. Maximally stable extremal regions (MSERs) are a



FIGURE 6: Original view of the road (experimental setup).



FIGURE 7: View of the road after threshold linearization.

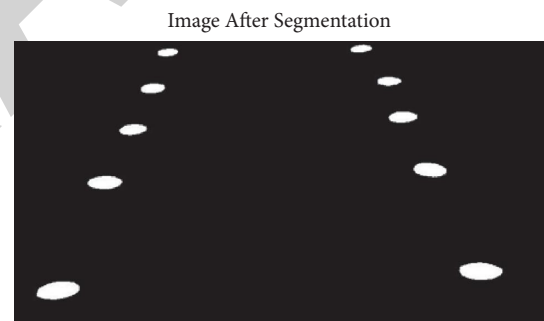


FIGURE 8: View after segmentation based on area and eccentricity properties.

method of object identification in pictures used in computer vision. Optical character recognition (OCR) technique is a commercial solution for extracting data from a printed or written text in a scanned document or image file and translating the text into a computer format for data processing such as editing or searching.

- (3) These detected MSERs are converted into the binary image with the MSER as logical 1 and 0. After this, blob analysis is performed to determine various properties of the detected MSER.
- (4) Then, the binary image of the MSER obtained is first filtered with the general geometric properties of alphanumeric characters, i.e.,
 - (a) $0.7 < \text{aspect ratio} < 3.6$
 - (b) $\text{Eccentricity} < 0.995$
 - (c) $\text{Euler number} > -3.0$

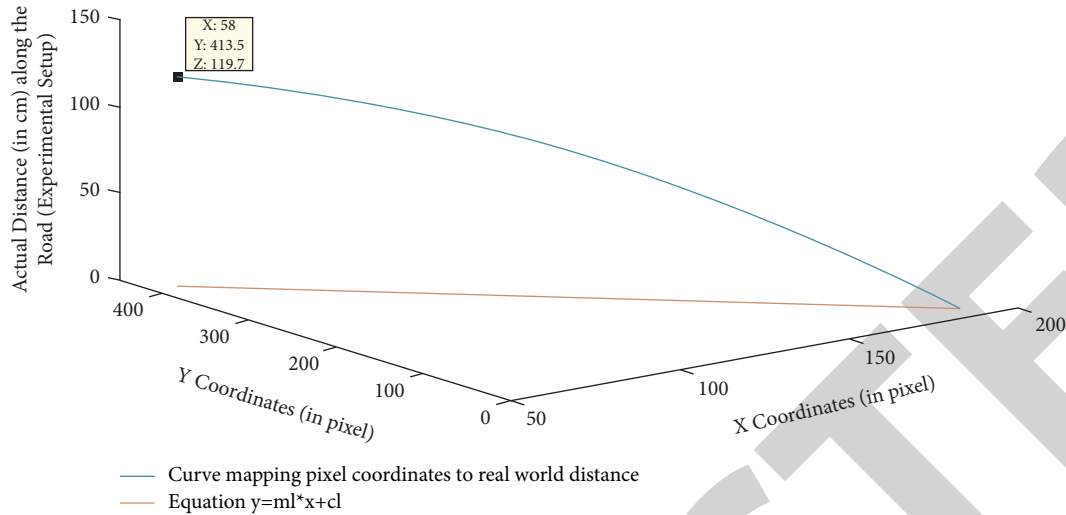


FIGURE 9: Graph mapping the pixel distance to actual distance on the ground assuming the farthest circular disk to be the reference point.

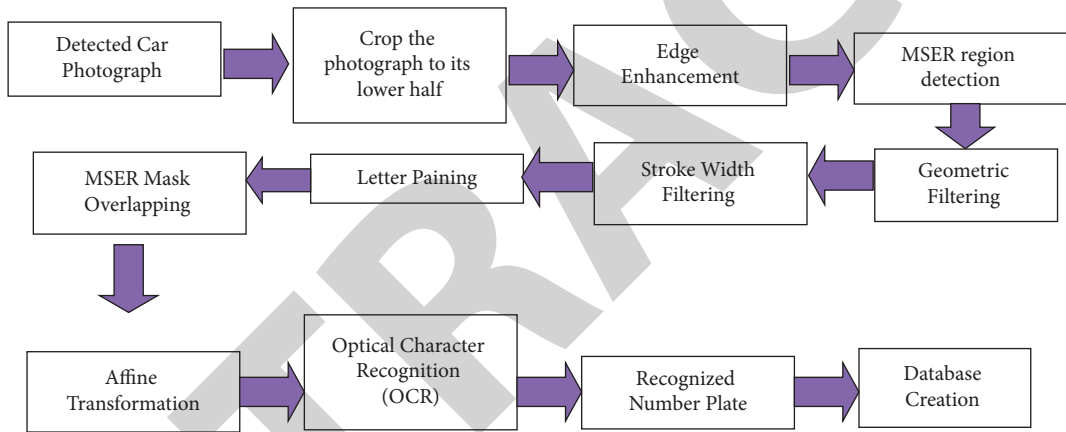


FIGURE 10: Flowchart of the methods used to detect and recognize the number plate of a vehicle.



FIGURE 11: Original image for number plate detection.

region tends to have larger stroke width variations. The stroke width is calculated by finding distance transform of the binary image of the MSER blobs and using the index locations of skeletonized blobs to find the maximum variation along the stroke of the blob as shown in Figure 13. Then, find the standard deviation and mean of the separations, and take the ratio of the two to obtain the normalised standard deviation. If it is greater than 0.5, then the detected blob is not an alphanumeric character. Then, such blobs are removed from the binary image.

- (d) $0.25 < \text{occupancy ratio (area of the blob/area of the bounding box)} < 0.92$
- (e) $\text{Solidity} > 0.4$
- (5) The left MSER is then filtered using the stroke width property which is a measure of width of the curves and lines that make up alphanumeric characters as shown in Figure 14. Text characters tend to have little stroke width variation, whereas the nontext
- (6) After these letters are paired whose height and mean stroke width are almost the same ($\pm 15\%$ tolerance) and the line joining their centroid has a slope of ± 15 degrees with the horizontal (x -axis of the frame), assume that the number plate is rectangular. This pairing is done in steps of two letters at a time as shown in Figure 15. Then, a bounding box is made around the whole line of alphanumeric characters, and if the aspect ratio is

greater than 2 and the bounding box is having a significant number of blobs (depending on the number of characters in the registration number), then it may be the detected number plate, but cropping this detected number plate from the original image of the vehicle and sending it to ANPR may result in false recognition as the border of the number plate or nontextual marking may also get included. To improve this paper, the binary image of the filtered MSER is utilized as a mask as shown in Figure 16.

- (7) This mask is cropped at the location of the detected number plate. This cropped mask image is logically inverted and added with the cropped original image at the place of the detected text. So, it is like seeing the number plate through a paper cutting. Adding is done in such a way that any intensity of pixel greater than 255 is made 255. Sometimes, the space between two alphanumeric characters is detected as a MSER blob text region as shown in Figure 17.
- (8) Therefore, we use thresholding using the following criteria to convert the colour image into the binary image:
 - (a) The pixel intensity <30% of 255 is given white or 1
 - (b) The pixel intensity >30% of 255 is given black or 0
 - (c) Now, this binarized image is sent for further processing

The resultant output is shown in Figure 18.

- (9) As the aim of this paper is to find a method to detect and recognize the registration number of vehicles on the road, it is highly likely that the car might have a slant approach towards the camera, and at that time, recognizing the detected number plate might give false/incorrect results. Figure 19 shows the slant approach of the car towards the camera and the detected number plate.
- (10) So, to avoid this, we extract the detected number plate and find the centroid of all the text blobs and fit a straight line through these coordinates and then use the slope (say θ) of this line to rotate the number plate image using affine transform with the help of this rotation matrix, and the results are shown in Figure 20.

$$\text{Rotation} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (15)$$

- (11) Then, the resultant image is sheered to obtain the upright number. The relationship between sheering and the slope of the tilted number plate is given by the sheering matrix.



FIGURE 12: Detected MSER within a maximum and minimum threshold area.



FIGURE 13: MSER blobs filtered using the geometric property.



FIGURE 14: MSER blobs filtered using stroke width variation as a parameter.



FIGURE 15: Detected text after letter pairing.

$$\text{Sheering} = \begin{bmatrix} 1 & 0 & 0 \\ -\sin \theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (16)$$

- (12) Finally, the resultant number plate is sent to the optical character recognition (OCR) for being detected as the alphanumeric registration number as shown in Figure 21.



FIGURE 16: Detected number plate after aspect ratio filtering.



FIGURE 17: Cropped and inverted binary mask overlapped with the original number plate.



FIGURE 18: Detected number plate obtained after clearing the noise of the background.



FIGURE 19: Slant approach of the car towards the camera and the detected number plate.

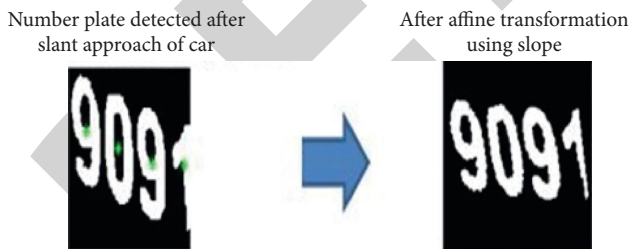


FIGURE 20: Result after affine rotation.

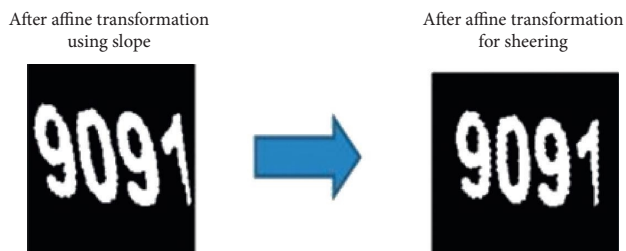


FIGURE 21: Result after affine sheering.

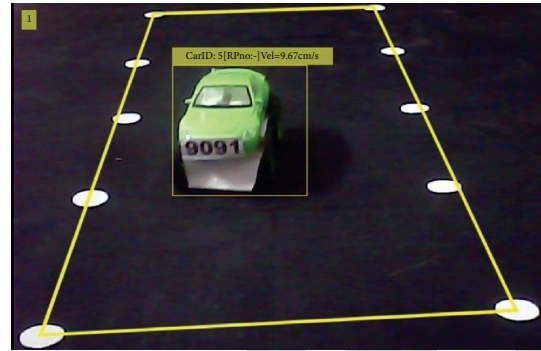


FIGURE 22: Frame showing the car being tracked on the experimental setup with ID 5 and monitoring its speed.

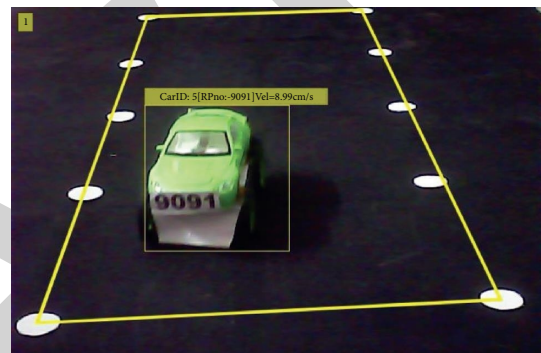


FIGURE 23: Frame showing the successful detection and recognition of the number plate at an optimum distance from the cameras' view.

6. Experimental Result

The methodology was tested, and it gave reliable results for the detection of tracking, speed, and number plate on the move. The entire algorithm could be executed on the laptop with i3 Core in real time because of its computational simplicity. The experimental results are shown in Figures 22 and 23.

7. Conclusion

The proposed traffic surveillance system provides an effective and relatively less expensive method for vehicle monitoring on the road using uncalibrated cameras which is a major requirement in developing countries. The proposed traffic surveillance system employs an uncalibrated camera to give a less expensive way of trustworthy motion control, and it presents a technique for automatically calibrating the road and then using it in vehicle velocity estimation. The speed estimation based on road calibration can be implemented easily in real scenarios, and the calibration is required only once as the camera is fixed with its view along the center of the road. The limitation of the present system is partial and complete occlusion of vehicles which we are currently working on to use a feature-based segmentation for partial occlusion.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research work was self-funded.

References

- [1] T. Kumar and D. S. Kushwaha, "An efficient approach for detection and speed estimation of moving vehicles," *Procedia Computer Science*, vol. 89, pp. 726–731, 2016.
- [2] Q.-C. Mao, H.-M. Sun, L.-Q. Zuo, and R.-S. Jia, "Finding every car: a traffic surveillance multi-scale vehicle object detection method," *Applied Intelligence*, vol. 50, no. 10, pp. 3125–3136, 2020.
- [3] J. Sochor, J. Roma, Š. Jakub et al., "Comprehensive data set for automatic single camera visual speed measurement," *IEEE Intelligent Transportation Systems Society*, vol. 5, pp. 1–11, 2018.
- [4] J. Sochor, R. Juránek, and A. Herout, "Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement," *Computer Vision and Image Understanding*, vol. 161, pp. 87–98, 2017.
- [5] K.-J. Kim, P.-K. Kim, Y.-S. Chung, and D.-H. Choi, "Multi-scale detector for accurate vehicle detection in traffic surveillance data," *IEEE Access*, vol. 7, pp. 78311–78319, 2019.
- [6] M. Bommès, A. Fazekas, T. Volkenhoff, and M. Oeser, "Video based intelligent transportation systems - state of the art and future development," *Transportation Research Procedia*, vol. 14, pp. 4495–4504, 2016.
- [7] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod, "Robust text detection in natural images with edge-enhanced maximally stable external regions," in *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP)*, pp. 2609–2612, Brussels, Belgium, September 2011.
- [8] R. Shankar, T. V. Ramana, P. Singh, S. Gupta, and H. Mehraj, "Examination of the non-orthogonal multiple access system using long short memory based deep neural network," *Journal of Mobile Multimedia*, vol. 18, no. 2, pp. 451–474, 2021.
- [9] K. Mahajan, U. Garg, and M. Shabaz, "CPIDM: a clustering-based profound iterating deep learning model for HSI segmentation," *Wireless Communications and Mobile Computing*, vol. 2021, Article ID 7279260, 12 pages, 2021.
- [10] X. Li, Y. She, D. Luo, and Z. Yu, "A traffic state detection tool for freeway video surveillance system," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 2453–2461, 2013.
- [11] G. Zhang, R. P. Avery, and Y. Wang, "Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1993, no. 1, pp. 138–147, 2007.
- [12] S. Chopra, G. Dhiman, A. Sharma, M. Shabaz, P. Shukla, and M. Arora, "Taxonomy of adaptive neuro-fuzzy inference system in modern engineering sciences," *Computational Intelligence and Neuroscience*, vol. 2021, Article ID 6455592, 14 pages, 2021.
- [13] K. Sharma and S. Gupta, "Positive and negative vibe classifier by converting two-dimensional image space into one-dimensional audio space using statistical techniques," in *Advances in Renewable Energy and Sustainable Environment*, vol. 667, pp. 59–67, LNEE Springer, Singapore, 2021.
- [14] H. A. Rahim, U. U. Sheikh, R. B. Ahmad, and A. S. M. Zain, "Vehicle velocity estimation for traffic surveillance system," *World Academy of Science, Engineering and Technology*, vol. 70, pp. 772–775, 2010.
- [15] F. W. Cathey and D. J. Dailey, "A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, pp. 777–782, Las Vegas, NV, USA, June 2005.
- [16] T. Gera, J. Singh, A. Mehbodniya, J. L. Webber, M. Shabaz, and D. Thakur, "Dominant feature selection and machine learning-based hybrid approach to analyze android ransomware," *Security and Communication Networks*, vol. 2021, Article ID 7035233, 22 pages, 2021.
- [17] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 246–252, Fort Collins, CO, USA, June 1999.
- [18] F. Wu, C. Zhu, J. Xu, M. W. Bhatt, and A. Sharma, "Research on image text recognition based on canny edge detection algorithm and k-means algorithm," *International Journal of System Assurance Engineering and Management*, 2021.
- [19] H. Li, P. Bai, and H. Song, "Car tracking algorithm based on Kalman filter and compressive tracking," in *Proceedings of the 7th IEEE International Congress on Image and Signal Processing (CISP)*, pp. 27–31, Dalian, China, October 2014.