Mallikarjun Anandhalli* and Vishwanath P. Baligar

An Approach to Detect Vehicles in Multiple Climatic Conditions Using the Corner Point Approach

DOI 10.1515/jisys-2016-0073

Received June 4, 2016; previously published online January 13, 2017.

Abstract: This paper presents a new method of detecting vehicles by using a simple and effective algorithm. The features of a vehicle are the most important aspects in detection of vehicles. The corner points are considered for the proposed algorithm. A large number of points are densely packed within the area of a vehicle, and the points are calculated by using the Harris corner detector. Making use of the fact that they are densely packed, grouping of these points is carried out. This grouping indicates that the group of corners belongs to each vehicle, and such groupings play a vital role in the algorithm. Once grouping is done, the next step is to eliminate the background noise. The Lucas-Kande algorithm is used to track the extracted corner points. Each corner point of the vehicle is tracked to make the output stable and reliable. The proposed algorithm is new, detect vehicles in multiple conditions, and also works for complex environments.

Keywords: Vehicle detection, corners, grouping, tracking.

1 Introduction

A robust and reliable vehicle detection algorithm is proposed in this paper. The proposed method makes use of one feature in detecting and tracking of vehicles. Only the corner points of the vehicle are calculated to detect the vehicles. The proposed system is simple and time effective. Aerial surveillance has the advantage of a wide field of view. It provides a suitable view continuously to check the traffic density of the vehicles [6]. Vehicles are considered as special objects in intelligent transport systems [7, 12]. The study of vehicle detection, including analysis of vehicles and the density of vehicles running in a particular lane, is very important in achieving optimal control of congestion, which helps in easing traffic flow by not allowing congestion to occur [3].

The background subtraction method is best suited for images captured by static camera, and it cannot work under certain conditions such as camera shaking, complex background, etc. This difficulty needs to be addressed in order to improve the accuracy of the traffic surveillance system. In the proposed algorithm, the objective is to detect vehicles in static as well as in moving camera images captured under different environmental conditions and complex backgrounds. The main contributions of the paper lie in localizing the vehicles based on the corner points and clustering these points.

The rest of the paper is organized as follows. Section 2 gives an overview of the latest research related to vehicle detection. Section 3 describes the proposed methodology in detail. Section 4 presents the experimental results and discussions to analyze the system performance. Finally, the paper is concluded in Section 5. Section 6 presents the future scope of the paper.

*Corresponding author: Mallikarjun Anandhalli, Department of Computer Science Engineering, B.V.B. College of Engineering and Technology, Vidyanagar, Hubballi-580031, Karnataka, India, e-mail: malliarjun71@gmail.com

Vishwanath P. Baligar: Professor, Department of Computer Science Engineering, B.V.B. College of Engineering and Technology, Vidyanagar, Hubballi-580031, Karnataka, India

2 Literature Survey

Lin et al. [4] proposed a system to detect moving vehicles on an airborne platform. The non-road region is concentrated in the frame first, which is filtered out by allowing those pixels with the value of the road color. The no-vehicle region from the road region is segmented later. This leads to reduction in computational cost, and all moving vehicles are segmented after image subtraction. The proposed tool detects moving objects with 86% accuracy. However, the algorithm still has many missed detections and false alarms.

Tsai et al. [8] presented a novel method of vehicle detection using the color and edge features of vehicles from static images. The authors transformed colors from R, G, and B color components to a new color domain. Later, vehicles and non-vehicle colors were classified by using a Bayesian classifier. The corners, edge maps, and coefficient of wavelet transform were used in verifying the object as a vehicle. The algorithm still needs improvement in complex environments to reduce the false alarms.

3 Proposed Methodology

3.1 Steps

The proposed methodology comprises the following steps:

- (i) Detection of corner points in every 16th frame using the Harris corner detector.
- (ii) Removal of corner points that do not belong to the vehicle.
- (iii) Grouping of the extracted corner points.
- (iv) Tracking corner points with the Lucas-Kanade algorithm to stabilize the result.

The flowchart of the proposed algorithm is shown in Figure 1.

The frames are extracted from the video sequence captured from the moving camera. For every extracted frame, the proposed algorithm is applied to detect vehicles.

3.2 Corner Detection

The Harris corner detector, proposed by Harris and Stephen [2], is used to detect the corner points of the vehicle. The algorithm is invariant to rotation, translation, and scaling of an image.

Figure 1: Flow Diagram of the Proposed Algorithm.

A local window in an image is considered, and the change in intensity that results after shifting the window by a small amount in different directions is calculated. The window patch in the image is flat iff the change in intensity is small in all directions. The window patch in the image is an edge iff there is a small change in a direction and relatively large change in other directions. The window patch is a corner iff the change in intensity is high in all directions. A brief insight about the Harris corner detection algorithm is explained.

Harris and Stephen measured the variation of intensity at position $I(x,y)$ by considering a small square window, e.g. $(3 \times 3, 5 \times 5)$, in the image, and this window is shifted by one pixel in all the directions. The sum of the squared intensity difference of the corresponding position in two windows multiplied by a Gaussian window defines the intensity variations [refer to Eq. (1)]. Consider a 6×6 image as shown in Figure 2. Consider the red color 3×3 be the original window centered at location (x,y) with intensity $I(x,y)$ (A5). The purple color 3×3 is the shifted window patch with the shift (u, v) , i.e. one pixel along the diagonal with the intensity $I(x+u, y+v)$.

The mathematical equation for intensity variation is given as

$$E_{u,v} = \sum_{x,y} w_{(x,y)} |I(x+u, y+v) - I(x, y)|^2, \quad (1)$$

where (u,v) is the arbitrary shift in the x and y direction, respectively; $w(x, y)$ is the weight of the Gaussian window at position (x, y) ; and $I(x,y)$ and $I(x+u, y+v)$ are the intensities of the image at position (x, y) and the moved window, respectively.

$I(x+u, y+v)$ can be rewritten using Taylor series expansion as shown in Eq. (2):

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \text{neglecting higher order terms}, \quad (2)$$

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v.$$

Substituting $I(x+u, y+v)$ in Eq. (1):

$$E_{u,v} = \sum_{x,y} w_{(x,y)} \left| I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v - I(x, y) \right|^2,$$

$$E_{u,v} = \sum_{x,y} w_{(x,y)} \left| \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v \right|^2,$$

$$E_{u,v} = \sum_{x,y} w_{(x,y)} |I_x u + I_y v|^2,$$

$$E_{u,v} = \sum_{x,y} w_{(x,y)} (I_x^2 u^2 + I_y^2 v^2 + 2uv I_x I_y).$$

Figure 2: Calculations of Intensity Variation for a 3×3 Window.

The above equation can be rewritten in matrix form as below:

$$E(u, v) = [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix},$$

where M is

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}.$$

Here, I_x and I_y are image derivatives in the x and y directions, respectively. The response function is given by

$$R = \det(M) - k(\text{trace}(M)).$$

The above equation will determine if a window contains an edge, a corner, or a flat region.

- where k is constant;
- where $\det(M) = \lambda_1 \lambda_2$;
- $\text{trace}(M) = \lambda_1 + \lambda_2$;
- λ_1 and λ_2 are eigenvalues of M .

A few points need to be noted to define edge, corner, and flat region in the window:

- (i) If λ_1 and λ_2 are large, i.e. R is large, then the window has a corner.
- (ii) If either of λ_1 or λ_2 is large, with $R > 0$, then the window has an edge.
- (iii) If R is small, then the window has a flat region.

The corner points are not extracted for every frame of the video but for every 16th frame in the video to reduce the time complexity.

3.3 Background Subtraction

Not all the corner points extracted from the image belong to the vehicles because of the background environment. Hence, the points that do not belong to vehicles need to be removed from the image. In order to remove these points, a background subtraction technique is performed. Frame subtraction is carried out to remove non-vehicle corner points. The frame difference is a pixel level difference between two adjacent frames. Only those points that belong to the vehicle are retained, as shown in Figure 4A.

3.4 Grouping of Objects

Once the extraction of the interest points from each object is done, grouping of these interest points is carried out. Assuming that the interest points extracted from each vehicle or object are highly dense in that area, i.e. closely packed together, we take advantage of these closely packed points and carry out grouping, as shown in Figure 4B. The grouping of points should be such that this group of points should correlate with that of the vehicle or object. In order to perform grouping, a minimum value of distance is considered between points, and this value is called the threshold value. If the distance between the points satisfies the threshold value, then the points are grouped as one. The distance between the points is calculated by the Euclidean formula [10]. Let $P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ be the two points on the plane; then, the Euclidean distance measurement between these points is as shown below:

$$f = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}. \quad (3)$$

For every point extracted, its distance from its neighbor is calculated. If the distance between the point and its neighbor is less than the threshold, then these two points are grouped as one. In this way, grouping is done for all the extracted points. The threshold value is nothing but the minimum distance value set between the points.

3.5 Tracking

The interest points that are extracted and grouped may not be reliable from frame to frame. We may lose some interest points from frame to frame because of environmental conditions. Thus, the output will vary drastically. In order to solve this problem, the interest points that are extracted are tracked from frame to frame. Tracking avoids loss of interest points in the video sequence. Tracking of these points is done by using the Lucas-Kanade algorithm [5]. The Lucas-Kanade algorithm is applied to estimate the position of the motion of a pixel from one image to another.

4 Results and Discussion

The experimental results of the proposed method are shown in Figures 3–8 and explained in detail in the respective section. The experiment was conducted on standard video datasets (25 video sequences). The results of the proposed work are shown in Table 1. Different videos of various climatic conditions, like rainy,

Figure 3: Results of Corner Detection.
(A) Input image. (B) Result of corner detection.

Figure 4: Grouping of Objects.
(A) Results of corner detection. (B) Grouping of corner points.

Figure 5: Results of Rainy Video Sequence.
(A) 80th frame of vehicle detection. (B) 95th frame of vehicle detection.

Figure 6: Results of Low-Resolution Video Sequence.
(A) Vehicle detection of video sequence 1. (B) Vehicle detection of video sequence 2 with non-road region.

Figure 7: Results of High-Resolution Video Sequence.
(A) Vehicle detection of video sequence within the city limits with more number of vehicles. (B) Vehicle detection of video sequence within the city limits with less number of vehicles.

Figure 8: Results of Fog Video Sequence.

(A) Vehicle detection of video sequence 1 with less number vehicles. (B) Vehicle detection of video sequence 1 with less number of corner point features. (C) Truck detection at fog climate in video sequence 2. (D) Vehicle detection in fog with number of vehicles in video sequence 2.

foggy, and sunny day with high and low resolution, were considered for the top view as an input to the system, not only for the different types of climates but also for the moving and static cameras. The proposed method also works for the rear view of vehicles and is discussed with single and multiple rear views of the vehicles. The algorithm detected all the vehicles in the video inputs, and the respective results are shown in the corresponding subsection. The proposed method works on the corner point feature of the vehicles and is implemented with Opencv and C++. This is configured on a 2.5-GHz, dual-core processor and a 4-GB RAM. The system processes more than 15 frames per second (fps). Detection precision is a parameter considered to check the quality of the system.

Figure 4A gives a pictorial representation of how the algorithm works. The dotted colored points are the feature points that are extracted from each vehicle. The Euclidean distance is considered to group these points. Upon seeing the results, we conclude that the method works well for all the video sequences used.

4.1 Detection in Rainy Climate

The algorithm was tested with videos of different climate conditions. One of the weather conditions tested in this section is rainy. The proposed algorithm was tested on this video, and it worked exceptionally well as shown in Figure 5A. The frame rate of the video was 10 fps, and the detection rate was 15 fps for this video. All of the vehicles were detected in the video. A few of the resultant frames of this video are shown in Figure 5A and B. Only the main lane was considered for the vehicle detection.

Table 1: Detailed Study of Vehicle Detection under Different Climate Conditions.

Sl. no.	Frame size/climate (time duration)	Fps	Average time taken to process each frame (ms)	Vehicles detected	No. of vehicles present (TP)	Missed vehicles (FP)	Falsely detected vehicles (FN)	Accuracy
1	320*240/Rainy (5 s)	10	61.2/30.2	7	7	0	0	100
2	320*240/Stock footage (rainy) (20 s)	29.75	62.189/31.9	13	13	0	0	100
3	320*240/Snow (1 min 10 s)	30	62.189/31.9	13	11	0	2	84.61
4	320*240/Sunny (33 s)	15	62.11/31.25	38	38	0	0	100
5	320*240/Sunny (30 s)	25	62.11/31.25	45	45	0	0	100
6	640*480/Sunny (high video) (58 s)	25	210.25/150.1	37	36	0	1	97.29
7	320*240/Fog sequence 1 (32 s)	30	62.186/32.85	46	45	0	1	97.82
8	320*240/Fog sequence 2 (2 min 36 s)	30	62.186/32.85	117	116	2	1	97.47
9	320*240/Fog sequence 3 (2 min 36 s)	30	62.186/32.85	136	134	2	2	97.10
10	704*480/Rear view (moving camera) (10 s)	30	257.73/180.636	1	1	0	0	100
11	704*480/Rear view (moving camera) (10 s)	30	348/189	4	3	0	1	75
12	720*640/Complex background (33 s)	25	319.159/160	15	14	0	1	93.33
13	720*640/Complex background (hard) with camera shaking (1 min 10 s)	25	319.159/160	45	41	0	4	91.11
14	720*480/Complex background 1 at the intersection (4 min 36 s)	30	300.25/135.23	78	70	0	8	89.74
15	720*480/Complex background 2 at the intersection (hard)/4 min 32 s	30	304.25/142.23	117	103	0	14	88.03
16	640*480/Intersection of roads/1 min	30	92.81/39.4	17	16	0	1	94.11
17	320*240/Fog sequence 2 complex background/2 min 36 s	30	62.189/31.9	11	11	1	0	91.67

4.2 Detection in Sunny Day with Low Resolution

The video was considered and tested on a sunny day with different scenes with a low-resolution camera. Undoubtedly, the algorithm worked well on the video with a much faster detection rate. The frame rate of the video was 30 fps, and the detection rate was 15 fps for this video. The corners of the vehicle were extracted accurately. Moreover, the corner points extracted from the vehicle were not densely packed when compared with a high-resolution video. All of the vehicles were detected in the video, and there were no missed detections. A few of the resultant frames of this video are shown in Figure 6A and B.

4.3 Detection in Sunny Day with High Resolution

The next video considered was a sunny day video taken with a high-resolution camera. Undoubtedly, the algorithm worked exceptionally well on the video. The corners of vehicles were extracted more accurately and the corner points extracted from the vehicle were more densely packed, making the grouping of the points and identification of the vehicles easy. The frame rate of the video was 30 fps, and the detection rate was 10 fps for this video. All of the vehicles were detected in the video, and there were no missed detections. A few of the resultant frames of this video are shown in Figure 7A and B.

4.4 Detection in Fog Video

To detect vehicles in the fog condition, an experiment was conducted by using a fog video. The proposed algorithm worked well on the fog video and detected all the vehicles present in the video, and the corner points extracted from the vehicle were not densely packed when compared with a low-resolution video. The frame rate of the video was 30 fps, and the detection rate was 15 fps for this video. All of the vehicles were detected in the video, and there were no missed detections. A few of the resultant frames of this video are shown in Figure 8A–D.

4.5 Rear-View Vehicle Detection with Moving Camera

The algorithm was tested with the rear view of the vehicle video. The proposed algorithm worked exceptionally well not only on the top-view videos but also on the rear view of the vehicle. It was tested on the rear view of the single vehicle. The corners of the vehicle were extracted accurately, and the corner points extracted from the vehicle were densely packed; however, the corner points of the surrounding environmental conditions were also extracted. This gave rise to noise; hence, thresholding should be effectively taken care of. The frame rate of the video was 30 fps, and the detection rate was 8 fps for this video. The vehicle was detected in the video, and there were no missed detections. A few of the resultant frames of this video are shown in Figure 9A–D.

4.6 Rear-View Multiple Vehicle Detection with a Moving Camera

The algorithm was tested with respect to detection and tracking of multiple vehicles in the rear-view video of the vehicle. The proposed algorithm not only worked exceptionally well on the single-vehicle video but also on the detection of the multiple vehicles in the rear view of the vehicle. The algorithm was first tested on the rear view of a single vehicle, and then on multiple vehicles. Working on the detection of vehicles would give rise to noise in the rear view of the video; hence, thresholding should be effectively taken care of. The frame rate of the video was 30 fps, and the detection rate was 8 fps for this video. The vehicle was detected in the video, and there were no missed detections. A few of the resultant frames of this video are shown in Figure 10A–D.

Figure 9: Results of Rear-View Video Sequence of Single-Vehicle Detection.

(A) Vehicle detection at curving point with moving camera capturing rear view of the vehicle. (B) Vehicle detection with no false detection at curving in presence of pedestrian. (C) Vehicle detection with moving camera. (D) Vehicle detection with moving camera in presence of noise (pedestrian and building).

Figure 10: Results of the Rear View of Multiple-Vehicle Detection.

(A) Multiple vehicle detection with high speed moving camera. (B) Multiple vehicle detection with moving camera in presence of noise (building).

Figure 11: Experimental Results.

(A) Complex background. (B) Complex background with camera shaking. (C) Complex background with more number of pedestrians at the intersection point. (D) Complex background at the intersection point. (E) Detection in a snowfield. (F) Detection in fog.

4.7 Performance Evaluation

The experimental results of the proposed algorithm are shown in Figure 11. The tabular representation of all the results and their accuracy, time taken, view, and fps are discussed in Table 1. The parameters are defined to evaluate the detection results of the proposed algorithm [9, 11], as follows: correctness, completeness, and quality, as shown in Eqs. (4) to (6).

Table 2: Comparative Analysis.

Sl. no.	Title	Accuracy	Time	View
1	Morphological operation [12]	91.57	Day	Top
2	Rear-lamp detection and tracking in color video [7]	92	Day	Rear
3	Detection and motion analysis in low-altitude airborne video [4]	89	Day	Top
4	Dynamic Bayesian network [1]	92.31	Day	Top

$$\text{Correctness} = \frac{TP}{TP + FP} * 100\%, \tag{4}$$

$$\text{Completeness} = \frac{TP}{TP + FN} * 100\%, \tag{5}$$

$$\text{Quality} = \frac{TP}{TP + FP + FN} * 100\%, \tag{6}$$

where TP is true positive: the number of correctly detected true vehicles; FP is false positive: the number of correctly detected false vehicles; and FN is false negative: the number of vehicles that are not detected.

The total overall correctness, completeness, and quality of vehicle detection were 99.32%, 94.63%, and 93.95% for the set of videos.

The computational complexity of the proposed algorithm is $O(N^2)$, where N is the dimension of the image. The time complexity varies with respect to the resolution of the videos. To reduce time, the corner detection algorithm is applied on every 16th frame and the Lucas-Kande algorithm is used to make the proposed algorithm stable by tracking the corner points to the first 15 frames. Tracking the corner points takes

Figure 12: Comparative Analysis Graph.

fewer milliseconds. For example, the average time complexity of the proposed algorithm for an image size of 320×240 is 92.36/61.2 ms, which means that the average time taken to detect vehicles by detecting the corner points of the vehicles is 92.36 ms. The average time taken to detect vehicles while tracking corners is 61.2 ms. Tracking is performed for the next 15 frames out of 16 frames. The average time taken for every video is shown in Table 1.

4.8 Comparative Studies

Different techniques have been used to detect and track the vehicles in aerial (top) and rear views. Table 2 explores the comparative analysis of four different techniques based on accuracy, time, and view. None of the techniques that have been implemented works on both top and rear views of the vehicles, as discussed. The proposed algorithm works on both top and rear views of the vehicles. The overall quality of the proposed algorithm for both the top and rear views of the vehicles in the video was 93.95% for the set of data. For the top-view videos with different climate conditions, the quality of the algorithm was 98% after testing it with 10 different videos. The graphical representation in Figure 12 shows the comparative study of different algorithms and their accuracy. We can see that the proposed system performs outstandingly well when compared to the accuracy of the top and rear views.

5 Conclusion

The proposed method works for any kind of environmental video. The work needs to be considered when the vehicles are too close to each other. The algorithm completely depends on the interest points of the vehicle extracted, and it is simple and time effective. All the vehicles are detected by the system. There may be a scope for improvement in the method by considering a few other features of vehicles that can help in detection to solve the merging problem, i.e. when two vehicles run close to each other.

6 Future Scope

There is no doubt that the system detects all vehicles efficiently; however, the problem that needs to be addressed is merging of two objects. The points are grouped based on their distance from each other; hence, two objects are detected as one when they run too close to each other. The work should still be carried out for grouping of objects by considering the maximum number of points for each object, so that points of one object are not considered in the grouping of another object.

Acknowledgments: The tested videos were downloaded from http://i21www.ira.uka.de/image_sequences/ and <http://www.svcl.ucsd.edu/projects/traffic/>, and two video datasets were borrowed from the Internet.

Bibliography

- [1] H. Y. Cheng, C. C. Weng and Y. Y. Chen, Vehicle detection in aerial surveillance using dynamic Bayesian networks, *IEEE Trans. Image Process.* **21** (2012), 2152–2159.
- [2] C. Harris and M. Stephens, A combined corner and edge detector, in: *Proceedings of The Fourth Alvey Vision Conference*, vol. 15, pp. 147–151, 1988.
- [3] Y. Li, B. Li, B. Tian and Q. Yao, Vehicle detection based on the and-or graph for congested traffic conditions, *IEEE Trans. Intell. Transport. Syst.* **14** (2013), 984–993.

- [4] R. Lin, X. Cao, Y. Xu, C. Wu and H. Qiao, Airborne moving vehicle detection for video surveillance of urban traffic, in: *2009 IEEE Intelligent Vehicles Symposium*, Xi'an, 2009, pp. 203–208.
- [5] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, in: *Proceedings of Imaging Understanding Workshop*, pp. 121–130, 1981.
- [6] R. O'Malley, E. Jones and M. Glavin, Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions, *IEEE Trans. Intell. Transport. Syst.* **11** (2010), 453–462.
- [7] B. Tian, Y. Li, B. Li and D. Wen, Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance, *IEEE Trans. Intell. Transport. Syst.* **15** (2014), 597–606.
- [8] L. W. Tsai, J. W. Hsieh and K. C. Fan, Vehicle detection using normalized color and edge map, *IEEE Trans. Image Process.* **16** (2007), 850–864.
- [9] C. Wiedemann, C. Heipke, H. Mayer and O. Jamet, Empirical evaluation of automatically extracted road axes, in: *Empirical Evaluation Methods in Computer Vision*, K. Bowyer and P. Phillips, Eds., pp. 172–187, IEEE Comput. Soc. Press, New York, 1998.
- [10] Wikipedia contributors, Euclidean distance, in: *Wikipedia, The Free Encyclopedia*, 8 September 2015, web 5 December 2015.
- [11] F. Yamazaki, W. Liu and T. T. Vu, Vehicle extraction and speed detection from digital aerial images, in: *IGARSS 2008–2008 IEEE International Geoscience and Remote Sensing Symposium*, Boston, MA, 2008, pp. III-1334–III-1337.
- [12] Z. Zheng, G. Zhou, Y. Wang, Y. Liu, X. Li, X. Wang and L. Jiang, A novel vehicle detection method with high resolution highway aerial image, *IEEE J. Select. Top. Appl. Earth Observ. Remote Sens.* **6** (2013), 2338–2343.