© Universiti Tun Hussein Onn Malaysia Publisher's Office

IJIE

Journal homepage: http://penerbit.uthm.edu.my/ojs/index.php/ijie
ISSN : 2229-838X e-ISSN : 2600-7916

The International Journal of Integrated Engineering

Contour Based Tracking for Driveway Entrance Counting System

Siti Norul Huda Sheikh Abdullah[1*], Abbas Salimi Zaini[2], Bedir Yilmaz[3], Azizi Abdullah[4], Nor Sakinah binti Md Othman[5], Ven Jyn Kok[6]

[1,2,3,5 6]Center for Cyber Security (CYBER),
Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia

[4]Center for Artificial Intelligence Technology (CAIT),
Faculty of Information Science and Technology,
Universiti Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia

DOI: https://doi.org/10.30880/ijie.2019.11.04.001
Received 18 April 2019; Accepted 7 July 2019; Available online 5 September 2019

Abstract: Managing vehicle in free-flow entrance is tiring to do manually by a guard control especially due to the increase in transportation demand. Providing an accurate vehicle counting approach is vital for traffic management and it will surely be an essential part in tomorrow's smart cities. Therefore, the main objective of this paper is to propose a more accurate vehicle counter by using the tracking and heuristic rules approaches. EzCam v1.0 is a vehicle surveillance system for a free-flow entrance where a module of vehicle counting based on proposed idea has been applied. The proposed method does not require high computational resources more than any relatively affordable non task specific hardware. It employs single threshold, contour extraction and sequential frame analysis and finally, vehicle counting process subsequently. The tracking-based method employs foreground object detection method and a mechanism for object filtering approach as compared to Chris Dahms approach which does not consider any object rejection and accept all contour information as relevant to be counted as vehicles. As a result, EzCam v1.0 which utilizes the exploited contour-based approach is able to achieve up to 94 percent of accuracy rate and outperforms the classic Chris Dahms method which obtained an accuracy of 88 percent. Therefore, the Exploited Contour based tracking method helps vehicle counting system to perform better accuracy in comparison to Chris Dahms approach.

Keywords: Counting, exploited contour based, foreground object detection, intelligent transportation system, surveillance.

1. Introduction

Urbanization has been one of the core socioeconomic trends that lead to the ever-expanding transportation supply and growing number of vehicles [1]. Generally, motorized vehicles, including cars, is the primary mode of transportation in urban areas. However, on the contrary to counting the number of people in a video, vehicle counting is largely under-studied but essential for intelligent transport management and forensic analysis [2]. For instance, real-time vehicle counting is useful for traffic control and traffic management [3]. In some cases, vehicle counting is important for parking management and it can be used to ensure safety of drivers and passengers on the road [4].

**Corresponding author: snhsabdullah@ukm.edu.my*

Several approaches have been proposed to achieve vehicle counting in unconstrained environments by relying on portable roadside sensors [5] or IoT using Raspberry Pi [6]. Promising results have also been demonstrated in recent studies that utilises conventional computer vision techniques [3,8,9,10] and convolutional neural network (CNN) based method [7]. Vehicle counting is generally accomplished by first inferring motion flow of objects in a video sequence, followed by using the detected salience or statistical features to count number of objects [8,9,10]. Optical flow is commonly employed to represent the motion flow of objects in the video sequences. Although optical flow is important to analyse motion flow, it is limited by the known drawbacks of optical flow which assumes brightness constancy. Specifically, the intensities of objects are assumed to remain fixed from one frame to the next. Such assumption rarely holds true for real-world surveillance videos due to camera stability and environment disturbance like camera shaking due to strong wind or large vehicle motion reflection [11].

To this end, unlike techniques that focuses on object recognition as a base for their vehicle counting applications, an end-to-end approach has been proposed in this paper which does not require any computational more than any relatively affordable non task specific hardware can provide. Specifically, the propose approach utilises contour-based tracking method to count the number of vehicles in a video.

This paper is organized into five sections namely introduction, related works, our proposed method, experimental results and conclusion.

2. Related Works

Vision based counting most likely dependent on top of other solution such as object detection. There are many ways to detect objects or vehicles on the road. Here, those methods have been categorized into two classes which are recent and traditional object detection methods. Recent methods consist of machine learning or deep learning methods [7-13] while, traditional object detection methods are optical flow, temporal differencing, background subtraction and statistical mean method [14-23].

2.1. Recent Object Detection Methods

2.1.1. Principal Component Pursuit (PCP)

Principal Component Pursuit (PCP) is the state-of-the-art method for video background modelling but it has not been utilized for object or vehicle detection. This is mainly due to the fact that existing PCP techniques are batch methods and this method has a high cost to execute which makes them inappropriate for real-time vehicle counting. However, a novel incremental PCP approach was developed which estimated the number of vehicles present in top-view traffic video sequences in real-time [13].

2.1.2. OverFeat Framework

OverFeat Framework is a machine learning framework which has a mixture of Convolution Neural Network (CNN) and one machine learning classifier such as Support Vector Machine (SVM) or Logistic Regression. This framework is being used in the object detection and also in object counting. CNN is used only for feature extraction due to the high amount of training data which results in slow and cumbersome training part. The extracted feature part is then passed on to a machine learning classifier, which does not require a huge amount of training data. The framework consists of three steps which are feature extraction, training and testing. This framework gives out accurate results but with a drawback of high computational cost [7].

2.2. Traditional Object Detection Methods

2.2.1. Optical Flow

Optical flow is frequently used as a critical component of video editing applications such as for object tracking, segmentation and selection [14]. This method used the vector characteristics of the object's motion which changes over time to identify the motion area in an image sequence. Optical flow is a dense field of displacement vectors which defined the translation of each pixel in a region. Optical flow is a complex algorithm with complicated computation and provides better results under the conditions that the camera is moving. In addition, additional disadvantages of this method are that it is highly sensitive to noise, and it cannot be applied to video streams in real time without specialized hardware [15]. Examples of a few well-known optical flow algorithms are SimpleFlow [14], Lucas & Kanade [16], Horn and Schunck [16], and TV-L1 [17].

2.2.2. Temporal Difference

Temporal difference method is a technique that used the pixel-wise difference between two or three consecutive frames in video imagery to extract a moving object. This method is computationally simple, fast and is a highly suitable to dynamic environment [18]. However, there is a drawback to this method which is, it is less effective in drawing out all information about the target object, especially when the target object has a uniform texture or when the object moves slowly. When a foreground object moves slowly or stops moving in between frames, temporal difference technique fails at detecting a change between consecutive frames and loses the target object information. Thus, this technique requires a special supportive algorithm to detect stopped or slow-moving objects [19]. To sum up, this method is applicable to moving objects and cannot be used for real time applications [15].

2.2.3. Statistical Mean Technique

Statistical mean technique is introduced to overcome shortcomings in current object detection algorithms. This method is based on the computed statistics of each and every consecutive frame of video or computed statistics of every k frames by using down sampling to lessen the processing time [19]. Even though statistical mean technique provides excellent results, but sometimes it suffers from the unnecessary effects contributed by foreground objects. This means that, whenever an object or more than one object remains in a video for a long period of time this would lead to an erroneous mean model. For example, if an object exists in half of the total number of frames, the result of the moving object detection will be hard to achieve. This incorrect result is produced due to the superfluous effect of that object [19].

2.2.4. Background Subtraction

Background Subtraction (BS) is a known technique or approach that allows an image's foreground to be derived for further processing. In other words, BS is a method of extracting the moving foreground or objects from the static background. Most cases, this technique has been proven successful whenever the camera is totally static with a stable noise-free background [20]. Factors such as weather, light, shadow and also clutter interference will affect the results and the accuracy of obtaining an optimal foreground [21]. Although there are variety of BS methods, a common denominator among the techniques is that they make an assumption that the video sequence or images that are being observed contains moving objects in front of a static background. A few well-known BS techniques are Basic Motion Detection (Basic), One Gaussian (1-G), Gaussian Mixture Model (GMM), Kernel Density Estimation (KDE), Codebook (CB^{RGB}), and Eigen Backgrounds (Eigen) [20].

The tracking of the object will be conducted after it is detected. From this paper [25], object detection methods are divided into three classes which are correspondence-based, transformation-based and contour-based. Another related work that also used contour-based tracking is the technique proposed by Chris Dahms [23]. It is a simple and a more robust technique that can be used for vehicle counting. It applies image subtraction followed by contour extraction. Contour information is then used to track the object detected. However, there is no object rejection in the method used by Chris Dahms [23] because all contour information will be accepted as an object. In the proposed method, heuristic rule which includes object rejection is introduced to increase the accuracy of vehicle counting.

Our proposed method will be using exploited contour-based tracking which extends the contour-based tracking by extracting object contours given an existing contour from the previous frame and heuristic contour decision information.

2.3. Object Counting Methods

Visual based counting task has always been a problem in the image processing field. Task such as counting the number of cells in a microscopic image, the number of humans present in an image of a crowd, and in this case is to count the number of vehicles in a free-flow entrance. There are two clear classes of counting methods which are supervised and unsupervised way. Unsupervised method is a grouping-based method that groups object based on similarities and conduct the counting afterwards. Such method is known to have low in accuracy as it is not robust and flexible at handling different situations [29]. Therefore, supervised methods are widely popular at handling object counting task.

Supervised counting are methods that can be based on detection and regression. Counting based detection method is performing the counting task after successfully detecting the objects. One of the ways to perform this is by doing background subtraction, where only the moving foreground object is present. In order to count the foreground moving objects, rectangle regions are selected and set on the image. Counting will be conducted if a vehicle passing through the rectangle regions. To identify an object or vehicle moving through the regions is by computing the mean of these rectangle regions. If the mean of the region is zero, then there are no vehicles passing through which no counting task will be performed. This method resolves relatively well towards various problematic situations such as presence of shadows and ghosts [26]. Furthermore, is by detecting objects using blob detection. Tracking of the blobs required an algorithm that can handle situations where the blobs would merge, split, appear and disappear. This needs to be considered if the task is to count pedestrians or objects that do not have restricted motions. When the blob appears in the frame and later on disappears, this is when then counting task is performed [27].

Supervised counting that is based on regression does not require any segmentation or tracking of every object. Main steps present in the regression-based algorithms are the pre-processing part where background subtraction is conducted. Further steps are detection of unclassified vehicles, feature extraction and cascaded regression [28]. Regression is widely used in the area of image processing. It is applied to predict the travel time, forecast traffic flow and also detect any traffic incidents. Regression is applied in order to count all types of vehicles. However, if three classes of vehicles are counted independently, then three regressions are applied for three classes. Thus, a three-level cascaded regression is used to count three types of vehicle. There are plenty types of regression where the simplest method is the linear regression. Other regression techniques are nonlinear regression with a Gaussian process applied and Poisson regression. Benefits of using regression based to perform the counting task are that prediction stage is usually very fast. The estimation of the vehicle boundary is not necessary when performing counting using regression. However, regression method does not perform well under complex conditions such as at the urban traffic.

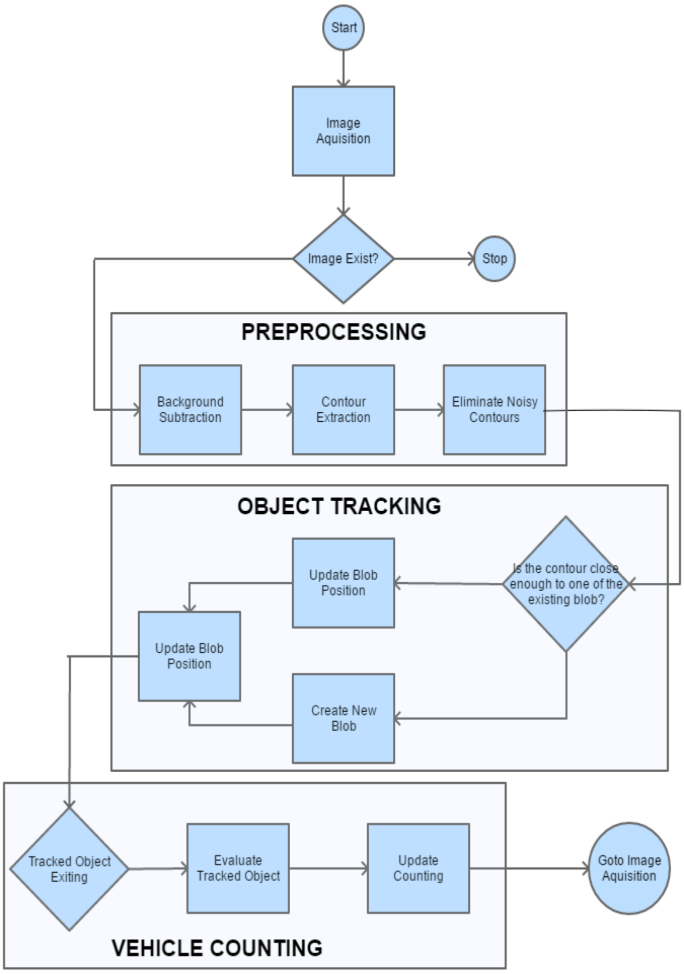

Fig. 1 - The flow chart of proposed counting algorithm namely ECTracking method.

3. Proposed Method

A method has been proposed to address the problem of vehicle counting in the context of free bi-directional flow entrance. Our proposed method involves in vehicle tracking and usage of vehicle positions for recognition of entrance and exit actions.

3.1. An Overview of The System

As an end-to-end framework, the system accepts CCTV streams and videos as input and produces the information of objects passed through a user-defined Region of Interest (ROI) line as output. This functionality makes our system suitable for the task of vehicle counting for entrance points of residential areas.

System consists of two major components (See Fig. 1):

- Image pre-processing and contour detection for creation of object envelopes
- Tracking of object envelopes using exhaustive search on envelope candidate areas

Object envelopes will be referred as blobs in the remaining sections of our paper. Each detection of entering and exiting actions to the designated physical area by vehicles will be referred as counting.

3.2. Implementation Details

In our implementation, the Java bindings of famous computer vision library OpenCV library [22] have been used for capturing the RGB frames (image acquisition) from video or CCTV cameras, background subtraction, contour extraction.

For the blob creation, the code of Chris Dahms [23] have been ported for object tracking and improved the suggested counting method by using our proposed method.

3.2.1. Pre-processing: Image Acquisition and Foreground Object Detection

As the first step of our algorithm, a frame from the RGB video stream has been acquired. Then, background subtraction is applied on it to extract the foreground objects exist on the scene. Then further elimination is applied to these contours to remove the redundant ones, e.g. overlapping and intersecting contours. Pre-processing part ends with the removal of these redundant contours (Algorithm 1).

Algorithm 1: Preprocessing Phase for vehicle counting system

Input : Source image, S
Output : Set vehicle candidate, VC
1: $k \leftarrow s$ //Convert S to grayscale, k.
2: $imgdiff \leftarrow imgdiffgaussian[i] - imgdiffgaussian[i-1]$ //Get image difference from two sequence gaussian frame images
3: $imgbin \leftarrow imgdiff$ //Subtract object from background using binary threshold such that $t = 50$.
4: Apply dilation and erosion using (3×3)kernel
5: $VC \leftarrow CCL(imgbin)$ //Set vehicle candidate, VC by extracting contour using connected component labelling.

3.2.2. Memory Allocation: Blob Creation

The main function of the post-processing part is to match the contours (foreground object envelope candidates) with the blobs. At the program start, after the analysis of the first frame, all of the existing contours will be stored as individual blobs. For remaining frames, another set of measures are being taken to ensure the continuation of these blobs.

3.2.3. Object Tracking: Location Projection for Blobs

A projection of next center location will be calculated for each blob upon creation and after each update to their location history. If the location history contains only one location, then the projection will be equal to the current position, for blobs with multiple locations in their history, a projection will be made based on the average of their displacement vector.

3.2.4. Object Tracking: Location Projection for Blobs

After the calculation of projection, the existing blobs is matched with eligible contours based on their locations. Referring to Algorithm 2, this is done via comparison of locations between contours and projections; with a degree of tolerance according to its object size, aspect ratio, object width, object height, and object corner size. In case of a match, these new contours were added to the location history of the corresponding blob.

Algorithm 2: Vehicle Tracking

Input : Set vehicle candidate, VC and vehicle location memory, VL
Output : Vehicle location memory, VL
1: For all objects in VC do
2: If$(object_{size} > 400 \cap aspect\ ratio\ object > 0.2) \cup (aspect\ ratio\ object < 4.0) \cap (object\ width > 30) \cup (object\ height > 30) \cap (corner\ object\ size > 60)$ then
3:         if object=object in VL
4:             Update object track in VL
5:         endif
6:     else
7:         Add object track in VL
8:     end
9: end.

3.2.5. Vehicle Counting: Detection Through Comparison of First and Last Locations in the Blob History

As shown in Algorithm 3, upon its disappearance, the locations of the blob were compared to understand its direction.

Algorithm 3 : Vehicle Counting

```
Input: vehicle.l.size
Output: Incount, OutCount
0: function VehicleCount(vehicles):
1: t = vehicles.l.size()
```

```
for v in vehicles do
  if v.hasDisappeared
     if v.l[0].y < crossingLine.y and v.l[t-1].y > crossingLine.y
              InCount++;
     endif
     else if v.l[0].y > crossingLine.y and v.l.last.y < crossingLine.y
              OutCount++;
     endif
  endif
end
```

The locations to be compared are the first location in its location history, namely the appearance location and the last location, which can also be named as disappearance location. The parameters of this comparison can change based on the scene that the program is configured for. If the counting has been configured on vertical axis of the scene, last location with a lower vertical position compared to the first location of the vehicle would be counted as an entering action. If the vertical value of the last location is higher, however, that would be counted as exiting action. For each vehicle v, a set of locations l are sorted by their appearance time t, a flow direction D_v can be recognized and used to count the vehicles entering and exiting follows:

$$InCount = \sum_{v=1}^{k} D_v = \begin{Bmatrix} 1 & l_{0_y} > l_{t-1_y} \\ 0 & l_{0_y} \leq l_{t-1_y} \end{Bmatrix} \quad (1)$$

$$OutCount = \sum_{v=1}^{k} D_v = \begin{Bmatrix} 1 & l_{0_y} < l_{t-1_y} \\ 0 & l_{0_y} \geq l_{t-1_y} \end{Bmatrix} \quad (2)$$

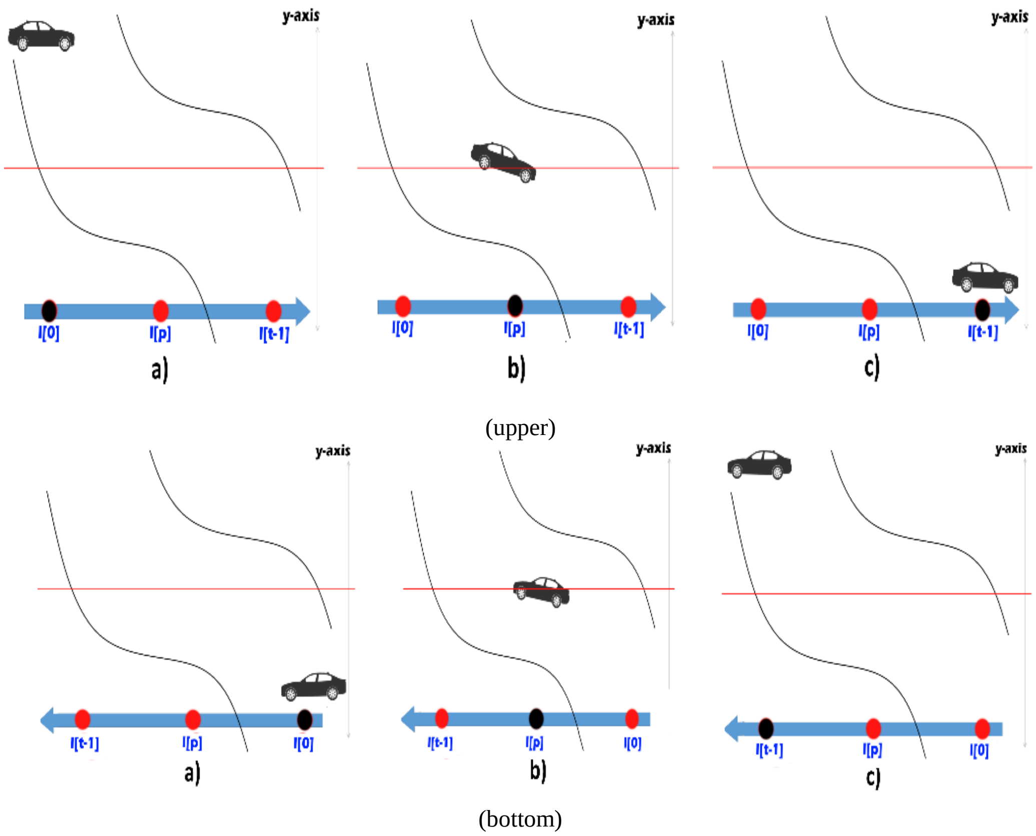

Fig. 2 - Illustrations of incoming and outgoing vehicles. Note that locations of vehicle are stored in memory in compliance with the timeline, *t*. Each location will be stored in array *l* with an index based on their appearance time from *a* to *c* which correspond to entering ($l[0]$), line passing ($l[p]$) and exiting ($l[t-1]$). To detect the whether there was an entrance or an exit, final vertical position of the vehicle is compared with the starting position. The vehicle is classified as entering if there is a decrease in vertical position and it is classified as exiting if there is an increase.

4. Experimental Results and Discussion

For a more robust comparison between the proposed algorithms, some established methods are needed. Therefore, to demonstrate the performance of the proposed algorithm, a real vehicle traffic flow data was used in these experiments. This dataset contains various vehicle types at different viewpoints. The evaluation measures and the performance results of our proposed algorithm compared to other systems will be explained in the following subsections.

4.1. Experimental Setup

The algorithm has been implemented on the real vehicle traffic flow at Desa Surada residential area. To test the proposed algorithm, a camera i.e. CCTV camera has been installed at front of the main road. The main road consists of bi-directional inflow and outflow traffics.

There is different type of vehicles such as car, motorcycle, van, bicycle, lorry and bus are entering to Desa Surada. In order to operate a successful surveillance, a small guard house or security post keeps all the hardware and software that required by CCTV.

The distance between CCTV and the main road is approximately more or less 3 meters. The security post is air-conditioned to maintain a temperature between 20 and 25 degrees Celsius. The interface of ezCam v1.0 is illustrated in Fig. 3.

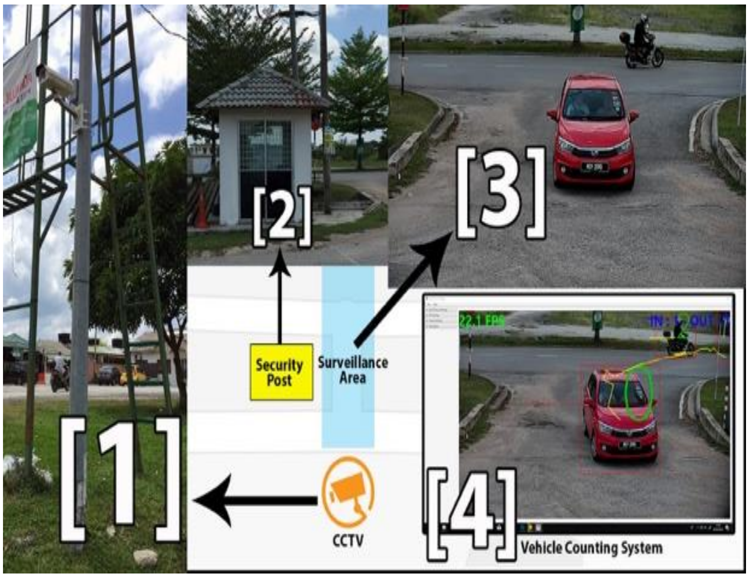

Fig. 3 - The figure shows the physical setup and some part of vehicle counting system user interface. (1) The CCTV is installed at front of the main entrance to Desa Surada; (2) The server and CCTV are kept in a small guard house located in Desa Surada; (3) Main entrance of Taman Desa Surada under CCTV surveillance; (4) A vehicle is detected by the system at the main entrance.

4.2. Evaluation Methods

In the experiments four evaluation measures have been used namely the F-Measure, recall, precision and accuracy. These measures are standardized, and they will enable us to compare our proposed algorithm with other systems.

4.2.1. F-Measure

For evaluating the proposed system performance, the F-measure has been computed as follows:

$$F - Measure = \frac{(2 \times Recall \times Precision)}{(Recall + Precision)} \quad (1)$$

Where,

$$Recall = \frac{TP}{(TP + FN)} \quad (2)$$

$$Precision = \frac{TP}{(TP + FP)} \quad (3)$$

$$Accuracy = \frac{TP}{(TP+FP+TN+FN)} \quad (4)$$

Where TP = True Positive, FN = False Negative, TN = True Negative and FP = False Positive.

Table 1 shows a total number of vehicles entered to Desa Surada for 83 min and 40sec. The number of vehicles is taken at three different durations. Based on this table then the F-Measure is calculated.

Table 1 - CCTV Recording Datasets tabulation obtained from entrance at Taman Desa Surada, Seksyen 8, Bangi, Selangor

Filename	Duration	Vehicle IN	Vehicle OUT	Total Count
cctv001.mp4	27min 29sec	78	53	131
cctv002.mp4	28min 09sec	44	62	106
cctv003.mp4	28min 02sec	64	38	102
Total	83min 40sec	186	153	339

Table 2 shows the F-measure value for a proposed counting method namely Exploited Contour Tracking (ECTracking). The results show that the proposed method outperformed other approach such as in Chirs Dahms. The proposed method achieved F-Measure and accuracy of 98.0% and 97.0% respectively. The ECTracking gives better precision results compared to Chris Dahms method due to its decision rules of taking vehicle entry point and exit point used to validate and count the vehicles. In contrast, the counting error in Chris Dahms method might be due its sensitivity in counting every object that passing through the virtual counting line without validating it. Thus, the validating scheme that is used in ECTracking is quite efficient in reducing the counting error.

Table 2 - F-measure, Precision and Recall for ezCam v1.0 vehicle counting system

Database	Vehicle In		Vehicle Out	
Method	Chris Dahms	ECTracking	Chris Dahms	ECTracking
TP	159	170	140	148
FP	5	0	3	0
TN	0	0	0	0
FN	22	16	10	5
Total	186	186	153	153
Precision	97%	100%	98%	100%
Recall	88%	91%	93%	97%
F-measure	92%	96%	96%	98%
Accuracy	85%	91%	92%	97%

Table 3 shows that the ECTracking based method obtained a higher average score (94 percent) compared to Chris Dahms method (88 percent). The ECTracking-based method employs foreground object detection method and a mechanism for object filtering approach. In this case, it takes contour information for each streaming frame and count vehicle or object only that permits with consistent size and gradient of motion directly after intersecting with the virtual counting line. In contrast, Chris Dahms method does not consider any object rejection whereby all contour information that intersects with the virtual counting line are relevant to count as vehicle. Therefore, the ECTracking based method can help to improve significantly the accuracy in counting vehicle in the vehicle counting system ($p < 0.05$). However, it takes slightly longer computing time compared to Chris Dahms method due to its complexity.

Table 3 - Average accuracy and standard deviation of ezCam v1.0 for vehicle counting system

Method	Chris Dahms	ECTracking
Vehicle in	85%	91%
Vehicle out	92%	97%

Average Accuracy	88%	94%
Standard deviation	4%	4%

5. Conclusion

This paper introduces ezCam v1.0 with an alternative way to do vehicle counting in a free flow entrance. The challenge of this algorithm is to track the orientation of the vehicles and count them according to in or outflow. Our proposed heuristic rule namely ECTracking mimicks the concept of human thinking whereby the inner or outer flows is only countable as the contour image and its central point are consistently appearing in each frame and intersecting to the virtual counting line. After testing on self-collection datasets at Surada Residence, our experimental results have proven that our proposed ECTracking method outperformed the classic way namely line-based method.

Acknowledgement

This research is supported by the Prototype Development Research Grant Scheme (PRGS) MoE Grant PRGS/1/2016/ICT02/UKM/02/1 from the Ministry of Higher Education Malaysia and AP2017-005-2 Grant from Universiti Kebangsaan Malaysia.

References

[1] Rodrigue, J. P., Comtois, C., & Slack, B. (2016). The geography of transport systems. Routledge.
[2] Kanacı, A., Zhu, X., & Gong, S. (2017). Vehicle reidentification by fine-grained cross-level deep learning. In British Machine Vision Conference (BMVC) AMMDS Workshop (Vol. 2).
[3] Yang, H., & Qu, S. (2017). Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition. IET Intelligent Transport Systems, 12(1), 75-85.
[4] Wang, K., Li, Z., Yao, Q., Huang, W., & Wang, F. Y. (2007). An automated vehicle counting system for traffic surveillance. In IEEE International Conference on Vehicular Electronics and Safety. ICVES. (pp. 1-6). IEEE.
[5] Taghvaeeyan, S., & Rajamani, R. (2014). Portable roadside sensors for vehicle counting, classification, and speed measurement. IEEE Transactions on Intelligent Transportation Systems, 15(1), 73-83.
[6] Iqbal, M., Yuwono, R. T., Fadlillah, H. M., & Widiyanto, S. (2017). Classification and Counting Using Raspberry Pi. Journal of Engineering and Applied Sciences, 12(8), 2207-2213.
[7] Biswas, D., Su, H., Wang, C., Blankenship, J., & Stevanovic, A. (2017). An automatic car counting system using OverFeat Framework. Sensors, 17(7), 1535.
[8] Yang, T., Pan, Q., Li, J., & Li, S. Z. (2005). Real-time multiple objects tracking with occlusion handling in dynamic scenes. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). (pp. 970-975). IEEE.
[9] Bouaich, S., Mahraz, M. A., Rifïi, J., & Tairi, H. (2018). Vehicle counting system in real-time. In 2018 International Conference on Intelligent Systems and Computer Vision (ISCV), (pp. 1-4). IEEE.
[10] Soleh, M., Jati, G., Sasongko, A. T., Jatmiko, W., & Hilman, M. H. (2017). A real time vehicle counting based on adaptive tracking approach for highway videos. In 2017 International Workshop on Big Data and Information Security (IWBIS), (pp. 93-98). IEEE.
[11] PirahanSiah, F., Abdullah, S. N. H. S., & Sahran, S. (2017). Augmented Optical Flow Methods for Video. In Computational Intelligence for Data Science Application (pp. 7-12).
[12] Kulchandani, J. S., & Dangarwala, K. J. (2015). Moving object detection: Review of recent research trends. In 2015 International Conference on Pervasive Computing (ICPC), (pp. 1-5). IEEE.
[13] Quesada, J., & Rodriguez, P. (2016). Automatic vehicle counting method based on principal component pursuit background modeling. In 2016 IEEE International Conference on Image Processing (ICIP) (pp. 3822-3826). IEEE.
[14] Tao, M., Bai, J., Kohli, P., & Paris, S. (2012). SimpleFlow: A Non- iterative, Sublinear Optical Flow Algorithm. In Computer Graphics Forum (Vol. 31, No. 2pt1, pp. 345-353). Oxford, UK: Blackwell Publishing Ltd.
[15] Prajapati, D., & Galiyawala, H. J. (2015). A review on moving object detection and tracking. Department of Electronics And Communication Engineering, UKA Tarsadia University, Bardoli, Gujarat.
[16] Galvin, B., McCane, B., Novins, K., Mason, D., & Mills, S. (1998). Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms. In Proceedings of the British Machine Vision Conference (BMVC). (pp. 195-204).
[17] Wedel, A., Pock, T., Zach, C., Bischof, H., & Cremers, D. (2009). An improved algorithm for tv-l 1 optical flow. In Statistical and geometrical approaches to visual motion analysis (pp. 23-45). Springer, Berlin, Heidelberg.
[18] Kinjal A. Joshi and Darshak G. Thakore. (2012). A survey on moving object detection and tracking in video surveillance system. International Journal of Soft Computing and Engineering, 2(3), 44-48.

[19] Narendra Chauhan Vahora, Safvan and Nilesh Prajapati. (2012). A robust method for moving object detection using modified statistical mean method. International Journal of Advanced Information Technology, 2(1), 65.
[20] Yannick Benezeth. (2010). Comparative study of background subtraction algorithms. Journal of Electronic Imaging, 19(3).
[21] Wei Li. (2010). Foreground detection based on optical flow and background subtract. In 2010 International Conference on Communications, Circuits and Systems (ICCCAS). (pp. 359-362). IEEE.
[22] G. Bradski. The OpenCV Library. Dr. Dobb's Journal of Software Tools, 2000.
[23] Chris Dahms. Latest commit of the GitHub Repository of the Code OpenCV 3 Car Counting CPP. https://github.com/MicrocontrollersAndMore/OpenCV\3\Car\Counting\Cpp, 2017. Online; accessed 02 May 2018.
[24] Yazan M. Alomari and S.N.H.S Abdullah. (2014). Automatic detection and quantification of wbcs and rbcs using iterative structured circle detection algorithm. Computational and mathematical methods in medicine 2014.
[25] Yilmaz, A., Li, X., & Shah, M. (2004). Object contour tracking using level sets. In Asian Conference on Computer Vision.
[26] Lei, M., Lefloch, D., Gouton, P., & Madani, K. (2008). A video-based real-time vehicle counting system using adaptive background method. In 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems (pp. 523-528). IEEE.
[27] Masoud, O., & Papanikolopoulos, N. P. (2001). A novel method for tracking and counting pedestrians in real-time using a single camera. IEEE transactions on vehicular technology, 50(5), 1267-1278.
[28] Liang, M., Huang, X., Chen, C. H., Chen, X., & Tokuta, A. O. (2015). Counting and classification of highway vehicles by regression analysis. IEEE Trans. Intelligent Transportation Systems, 16(5), 2878-2888.
[29] Lempitsky, V., & Zisserman, A. (2010). Learning to count objects in images. In Advances in neural information processing systems (pp. 1324-1332).