Vehicle Counting at Road Intersections Using Video Data and Zone-Based Method with Additional Contour

O Ibrayeva¹*, A Zhulev¹, V Shepelev¹, A Burzev²

¹South Ural State University, Lenina avenue, 76, Chelyabinsk, Russia
²Kuzbass State Technical University named after T.F. Gorbachev, Ilyich st., 32a, Belovo, Russia
*Email: ibrievaol@susu.ru

Abstract. The problem of counting vehicles and determining the directions of their movement at an intersection is considered. A new zone-based method for counting vehicles which uses an additional inner contour is proposed. The presence of an additional inner contour helps to partially expand zones of interest and increases our chances of catching a vehicle in time. YOLO and CSRT tracker are used for detection and tracking of objects. Full flowchart of the working process including detection, tracking and counting, is given. The algorithm for inner contour constructing is proposed and it is shown that this contour presence reduces the error in counting vehicles.

1. Introduction

Recently, due to the increase in the number of roads and vehicles, the task of controlling traffic with a smart transport system has become increasingly important. A large number of studies in this area are based on the analysis of video data by computer vision systems. The output values of this analysis are, for example, the class of the object, its speed, trajectory, number of objects, the presence of traffic jams and accidents, etc. The most challenging task is the real-time video analysis, taking into account the presence of shadows, bright sun, overlapping, rain, snow, fog. Despite numerous works and already existing solutions, we can say that the task of accurately calculating traffic, taking into account all these difficulties, is still far from complete.

Usually, in video analysis of traffic, the background subtraction (BS) method and its various modifications are used [0-0]. However, this method fails in the case of heavy traffic conditions. The presence of shadows and overlapping objects also prevent good object detection. There are various modifications of the BS method [0, 0], which try to solve these problems, but even they do not always cope. The BS method is even less suitable when, in addition to detecting objects, we also need to classify them, and we can do this either by comparing the relative size of the objects or their Histogram of Gradient. Truly, only deep convolutional networks can cope with the classification problem [0], however, they require a lot of time for their work. The YOLO network [0] perfectly solves this problem.
by detecting all objects of the frame "at a glance" (You Only Look Once), and providing a speed of 45 frames / seconds. In this paper, we have chosen YOLO method of detection and identification of objects.

YOLO detector is not enough for the task of counting vehicles. We have to track objects on each frame, to determine whether they are old or new in order not to count the same object several times. It is important not to lose or mix up previously found objects. The problem of tracking objects in a video sequence is extremely relevant and many works have been devoted to it, for example, [0-0]. This problem is also complicated by the possible overlap of objects, changes in illumination, fast speed of objects, similarities with the background, etc. In particular, trackers based on the discriminative correlation filter (DCF) method [0-0] have shown great performance in all standard benchmarks. Discriminative correlation methods learn a filter with a pre-defined response on the training image. The latter is obtained by slightly extending the region around the target to include background samples. However, the published DCF methods are not without drawbacks.

The Discriminative Correlation Filter with Channel and Spatial Reliability [0] overcomes both the problems of circular shift enabling an arbitrary search (and training) region size and the limitations related to the rectangular shape assumption. It shows state-of-the-art performance on standard benchmarks while running close to real-time on a single CPU and implemented in OpenCV as CSRT tracker. Our work combines the object detection unit with the CSRT tracker to count the traffic data.

There is a considerable amount of research in general purpose detection and tracking, but relatively little work on the application to large scale transportation tasks such as counting. Most works use counting systems based on a line crossing approach. Vehicles are counted every time their bounding boxes cross a user defined line [0]. In [0] a polygon is drawn in the scene and each side of the polygon corresponds to an entering or exiting virtual gate. In [0] counting is performed by determining the number of trajectories that are close to a source or a sink (entering and exiting points in the scene). Some works count vehicles when they are present in special zones [0].

The real challenge for video analysis is the case of an intersection - there are a lot of vehicles and one need to track the trajectory of each of them. At first glance an obvious approach is to keep track of which areas each car has visited and, based on this, make a conclusion about its direction of movement. However, due to the fast speed, the object may slip through the desired zone and be detected later. Another approach in through trajectories, which are then checked for intersections with a line or zone of interest is also not very successful, because due to the large number of objects, they may be confused, or also start too late due to the late detection of the object. In this paper, we use an approach based on tracking the entry of an object into certain zones. We propose to solve the problem of late object detection by introducing an additional inner contour.

The paper is organized as follows. In the next section, the algorithm for counting vehicles and determining the direction of their movement is described. The algorithm for constructing an additional contour is also given. Two algorithms (with and without an additional contour) are compared in Section 3. Finally, Section 4 contains the main conclusions of the work.

2. Proposed Method

2.1 Algorithm of Inner Contour construction
The direction of movement of the vehicle is determined by its presence in the zones of interest. The correctness of such determining depends on whether the object was detected in time. If the vehicle is going at high speed, YOLO may not be able to detect it in the desired zone and we cannot determine the arrival area of the object. The presence of an additional inner contour helps to partially expands zones of interest (Figure 1) and increases our chances of catching a vehicle in time.
Let us present the algorithm for constructing the inner contour. Given the four vertices
\[ A(x_A, y_A), B(x_B, y_B), C(x_C, y_C), D(x_D, y_D) \]
of the outer contour (Figure 2).

Algorithm for Inner Contour Construction

1. Find the equations of the lines \( AC : \frac{x-x_A}{x_C-x_A} = \frac{y-y_A}{y_C-y_A} \) and \( BD : \frac{x-x_B}{x_D-x_B} = \frac{y-y_B}{y_D-y_B} \).

2. Find the point \( O(x_o, y_o) \) of intersection of two lines \( AC \) and \( BD \) from the system:

\[
\begin{align*}
  x_o & = \frac{1}{\Delta} [x_C-x_A \cdot y_o - y_C-y_A \\
  y_o & = \frac{1}{\Delta} [y_C-y_A \cdot x_o - x_C-x_A]
\end{align*}
\]

We have \( x_o = \frac{\Delta_1}{\Delta}, y_o = \frac{\Delta_2}{\Delta} \) where

\[
\Delta = \begin{vmatrix}
 x_C-x_A & 1 \\
 x_D-x_B & 1
\end{vmatrix}, \Delta_1 = \begin{vmatrix}
 y_C-y_A & 1 \\
 y_D-y_B & 1
\end{vmatrix}, \Delta_2 = \begin{vmatrix}
 x_C-x_A & y_C-y_A \\
 x_D-x_B & y_D-y_B
\end{vmatrix}
\]

3. Find the vectors \( OA, OB, OC, OD \) and the vectors \( \overrightarrow{AA} = OA \cdot k \), \( BB = OB \cdot k \), \( CC = OC \cdot k \), \( DD = OD \cdot k \).

4. Find the vertices \( A', B', C', D' \) of the inner contour using the coordinates of the vertices \( A, B, C, D \)
and vectors \( \overrightarrow{AA}, \overrightarrow{BB}, \overrightarrow{CC}, \overrightarrow{DD} \).

The parameter \( k \) is responsible for how close the inner contour will be to the outer one and selected experimentally. The inner contour should not be too close in the outer, otherwise its meaning will disappear. At the same time, it should not be too far from the outer contour, since in this case again we will not get a good expansion of the zones of interest (because of the four large white rectangles in the Figure 2).
2.1 Zone-Based Method with Additional Inner Contour

The working process of the proposed method is depicted in Figs. 3, 4. We combine YOLO detector to recognize the vehicles in video data, followed by tracking using CSRT tracker from OpenCV. In YOLO working each frame is divided into MxM (ex. 13x13) grids, and each grid predicts B (ex.9) bounding boxes with the confidence scores. Each bounding box is associated with x, y, w, h, confidence score as predictions. Each grid cell predicts C (ex. 80) conditional class probabilities to indicate the likelihood of object in it.

We used YOLO trained on COCO dataset and it can predict 80 classes such as bicycle, car, truck, motorcycle, person etc. The confidence score of each bounding box and class predictions are combined to estimate the object YOLO is applied to the image every 15 frames in our 25 FPS video. The class "person" is deleted from the detection process. Each detected object is attributed by its ID, centroid and the bounding box. The detected objects are tracked across the video frames by CSRT tracker.
The presented method uses both inner and outer contours. In order to investigate how much the inner contour improves the efficiency of the method, we compare it with the method which uses only one outer contour.
The flowchart for this method is very similar and we did not give it. It does not contain calculating the position of the inner contour control points in block “Configuration” and setting the object position relative to the inner contour in the block “Counting”. The next section compares these two methods.

3. Experimental analysis

To test the robustness of vehicle counting, 3 scenarios lasting 5 minutes, including day, night and rain are selected from the real video from the intersection of our city. To evaluate the quality of the counting result, the Absolute Percentage Error (APE) was used. Here $N$ and $GT$ are respectively the number of vehicles counted automatically and the ground-truth number of vehicles.

The APE of the vehicle counting in our methods without and with the additional contour is listed in Table 1. All possible turning movements counts were considered.

The value $APE = 100\%$ is taken in those cases when no objects were found in this particular direction ($N=0$), but in fact the cars passed there ($GT\neq0$). There were cases when $GT=0$ in the reviewed video files. In these cases, we always had $N=0$ and $APE = 0\%$. Of course, for the cases when $GT=N$ we also have $APE = 0\%$.

The last line shows the average $APE$ for the video file. As one can see, the average $APE$ decreases significantly in the case of the algorithm with additional inner contour.
Table 1. The Absolute Percentage Error values for Algorithms without/with Inner Contour.

| Zones of interest | Algorithm without inner contour | Algorithm with inner contour |
|-------------------|--------------------------------|------------------------------|
|                   | 26.07.19 06.08.19 02.08.19     | 26.07.19 06.08.19 02.08.19     |
|                   | Day Night Rain Day Night Rain Day Night Rain |
| Top-Top           | 0 0 0 0 0 0 0 0 |
| Top-Right         | 0 0 0 100 0 100 0 100 |
| Top-Bottom        | 33.33 100 48.15 0 54.55 22.22 |
| Top-Left          | 0 100 0 100 0 100 |
| Right-Left        | 0 0 0 100 0 0 |
| Right-Right       | 0 0 0 100 0 100 |
| Right-Bottom      | 41.18 7.14 43.75 17.65 0 28.125 |
| Right-Left        | 56.41 16.67 83.33 51.28 16.67 81.82 |
| Bottom-Top        | 16.67 76.92 58.82 5.56 61.54 35.29 |
| Bottom-Right      | 32 20 10.20 4 0 10.20 |
| Bottom-Bottom     | 0 0 0 0 0 0 |
| Bottom-Left       | 0 60 23.08 0 60 11.54 |
| Left-Top          | 0 0 25 0 0 0 |
| Left-Right        | 14.81 48.39 35 11.11 40.91 25 |
| Left-Bottom       | 11.11 8.33 6.90 0 7.69 0 |
| Left-Left         | 0 0 0 0 0 0 |
| Average           | 12.84 27.34 39.64 5.60 21.34 27.97 |

4. Conclusions
In the paper, the zone-based method for counting vehicles is proposed. The method uses a new idea of expanding zones of interest due to the additional inner contour. The algorithm for the inner contour construction is also presented. The advantage of the proposed counting method is shown experimentally.

The absolute percentage error in vehicles counting when using the outer and inner contours is significantly less than when we use only the outer one. Note that the error values obtained in themselves are quite large, but this is due to insufficient network and tracker performance. Our goal was to demonstrate the advantage of using an inner contour and, as we can see, other things being equal, this method works better. As for improving the network and tracker, we will devote further work to this.

References
[1] C Pompanomchai, T Liamsanguan, V Vannakosit, 2008 Vehicle detection and counting from a video frame, Wavelet Analysis and Pattern Recognition 1 356-361
[2] V Pham, B Lee 2015 Front-view car detection and counting with occlusion in dense traffic flow, International Journal of Control, Automation and Systems 13(5) 1150-1160
[3] H Jang, I Won, D Jeong 2014 Automatic Vehicle Detection and Counting Algorithm, International Journal of Computer Science and Network Security 14(9) 99
[4] P Barcellos, C Bouvie, F Escouto 2015 A novel video based system for detecting and counting vehicles at user-defined virtual-loops, Expert Systems with Applications 42(4) 1845-1856
[5] Y Xia, X Shi, G Song, Q Geng, Y Liu 2016 Towards improving quality of video-based vehicle counting method for traffic flow estimation, Signal Processing 120 672-681
[6] G Salvi 2014 An automated nighttime vehicle counting and detection system for traffic surveillance, Computational Science and Computational Intelligence 1 131-136
[7] C Ren, K He, R Girshick, D McAllester, Faster R 2015 CNN: Towards real-time object detection with region proposal network, Advances in Neural Information Processing System 91-99
[8] J Redmon, S Divvala, R Girshick, A Farhadi 2016 You only look once: Unified, real-time object detection, Proc of the IEEE Conference on Computer Vision and Pattern Recognition 7780460 779-788

[9] Y Wu, J Lim, M Yang 2013 Online object tracking: A benchmark, Comp Vis Patt Recognition 6619156 2411-2418

[10] M Kristan, J Matas, A Leonardis, T Vojir, R Pflugfelder, G Fernandez, G Nebehay, F Porikli, L Cehovin 2016 Novel performance evaluation methodology for single-target trackers, IEEE Trans Pattern Anal Mach Intel 38(11) 2137-2155

[11] P Liang, E Blasch, H Ling 2015 Encoding color information for visual tracking: Algorithms and benchmark, IEEE Trans Image Proc 24(12) 5630-5644

[12] M Mueller, N Smith, B Ghanem 2016 A benchmark and simulator for UAV tracking, Lecture Notes in Computer Science 9905 445-461

[13] M Danelljan, G Hager, F S Khan, M Felsberg 2014 Accurate scale estimation for robust visual tracking, Proc British Machine Vision Conference 1-11

[14] M Danelljan, G Hager, F Shahbaz Khan, M Felsberg 2015 Learning spatially regularized correlation filters for visual tracking, Proc Int Conf Computer Vision 4310-4318

[15] J F Henriques, R Caseiro, P Martins, J Batista 2015 Highspeed tracking with kernelized correlation filters, IEEE Trans Pattern Anal Mach Intell 37(3) 583-596

[16] Y Li, J Zhu, A 2014 scale adaptive kernel correlation filter tracker with feature integration, Proc European Conf Computer Vision 254-265

[17] A Lukezic, T Vojir, L Zajk, J Matas, M Kristan 2018 Discriminative correlation filter tracker with channel and spatial reliability, International Journal of Computer Vision 126(7) 671-688

[18] K Jie, M Liu 2005 Computer vision based real-time information acquisition for transport traffic, Information Acquisition

[19] S Messelodi, C M Modena, M Zanin 2005 A computer vision system for the detection and classification of vehicles at urban road intersections, Pattern Analysis and Applications 8(1) 17-31

[20] R Zhao, X Wang 2013 Counting vehicles from semantic regions, IEEE Transactions on Intelligent Transportation Systems 14(2) 1016-1022

[21] J C Mendes, A G C Bianchi, A R P Junior 2015 Vehicle tracking and origin-destination counting system for urban environment, Proc of the 10th International Conference on Computer Vision Theory and Applications 3 600-607