Archives of
Transport System
Telematics

Volume 10

Issue 3

September 2017

A new method to estimate dimensions of vehicle using a single camera

M. BAZAN, T. JANICZEK, K. HALAWA, A. WYKRZYKOWSKI, P. ŁABUDA
WROCŁAW UNIVERSITY OF SCIENCE AND TECHNOLOGY, Faculty of Electronics, ul. Janiszewskiego 11/17, 50-372 Wrocław, Poland
EMAIL: marek.bazan@pwr.edu.pl

ABSTRACT

Abstract. The capability of a telematic vision system of estimating dimensions of vehicles is used for such tasks as vehicle classification or preselection of vehicles that violate local vehicle size limitations. Also in some European countries dimensions of heavy vehicles must obey some global regulations. Furthermore, vehicle size estimation allows us to determine the structure of traffic and can be very useful for advanced traffic flow control. Many existing Intelligent Transportation Systems consist of a large number of video cameras located in various places e.g., the ITS in Wrocław uses more than 1400 cameras. In this paper we propose a new method developed by the ArsNumerica Group and CyberTech Scientific Circle for the precise estimation of vehicle sizes using a single camera. The method does not require the entering of measurements such as the distance between lane lines or the height of the camera above the roadway. Only one vehicle's dimensions are used for calibration. The proposed method is easy to implement and may be applied with the OpenCV library which is free both for academic and commercial use. The method is tested on real-world video streams. The obtained results are shown and analyzed in the paper.

KEYWORDS: vehicle dimensions estimation, single camera vehicle sizes estimation, calibration

1. Introduction

Presently, a large number of cameras are used for road traffic control or monitoring. For instance, in the Wroclaw ITS (Intelligent Transportation System), more than 1000 cameras were applied, located at major junctions and main roads. Frequently, on the basis of automatic vision image analysis, the traffic intensity is determined, incidents and accidents are detected [1], interval-based speed measurements are taken, the settings of traffic lights are selected with use of various algorithms [2, 3] and the travelling times are forecast [4, 5]. The more and more frequent use of cameras for investigation of traffic results from the fact that they may provide diversified information and, in contrary to the inductive loops, they do not need to build in any devices into the road pavement. The images showing the road traffic may be processed easily with the OpenCV library [6,7], the license for which makes it possible to use it free of charge both in commercial and scientific projects.

On the basis of images from the cameras, it is also possible to estimate the vehicle dimensions [8],that are useful for, among others, classification of the vehicles, pre-selection of the vehicles violating local regulations concerning the permissible overall dimensions, the determination of the traffic structure, e.g., the share of lorrys, vans, cars and other vehicles. One of the relatively easy ways for determination of an object's dimensions consists in the use of stereoscopic images and in calibration through showing a definite geometric pattern. In the OpenCV library, there are pre-prepared procedures for calibration of stereo cameras in such a way. However, in a traffic scene, this is usually impractical since one would need a very large calibration pattern left alone on the road [8]. There is a high number of methods that make use of the knowledge of camera heights and distance, their focal length, etc. Some methods need specification of the dimensions of the immobile objects located in the images, e.g., roadway widths, broken line lengths, the distances between the points laying on one line [9,12], etc. There also exist the camera auto-calibration algorithms based on a complicated detection of vanishing points [10-11], which generates many problems. In addition, in the study [10], for auto-calibration, the statistical data concerns the automobiles sold and their dimensions.

In the present paper, a roadside camera calibration method has been proposed that does not need knowledge of the camera distance, its mounting height, focal length nor any other camera characteristics, roadway dimensions and the sizes of other immobile objects visible in the image. The method enables easy calibration on the basis of the dimensions of any automobile. The dimensions are available, in general, in the manufacturers' catalogues.

The rest of the paper is organised as follows. Section 2 describes the proposed method. In section 3 the obtained results are shown. Discussion of all the results, method limitations and conclusions are at the end of the paper.

2. Proposed method

2.1. Background subtraction

The process of estimating dimensions of vehicles in a video must begin with background subtraction and, subsequently, grouping the resulting foreground points into blobs. These tasks are essential to any algorithm pertaining to acquiring data from a video, however selection and implementation of a Background Subtraction method are beyond the scope of this paper. Quality of dimensions' estimates depends on this initial subtraction, as well as differentiating between moving objects and shadows cast by them. Study [15] presents methods for removing shadows. Our method requires a set of points, grouped into blobs, representing moving objects on the scene.

For the purpose of this research, these issues are solved using functions provided by the OpenCV library. After acquiring a global set of points from a MOG2 Background Subtractor, several erosion and dilation transformations are performed to limit the amount of noise resulting from data imperfection.

Next, using the library's contour detection algorithm we can separate the global set of points into several smaller sets of points neighbouring each other, called blobs. Once a set of blobs for the current frame of the video is determined, construction of 3D bounding boxes may begin.

As a first step, a 2D, axis aligned bounding box containing our blob is found. This is also a standard function included into the OpenCV library. Let's mark the coordinates of the top left corner as (x1, y1) and bottom right corner as (x2, y2). It is worth noting here that the following explanation applies to a 2D coordinate system, with the X axis pointing right and Y axis pointing down, as is most common in computer graphics. Repurposing the algorithm to work within a standard Cartesian coordinate system relies mainly on multiplying slopes of all lines by -1. This, together with swapping values of bmax and bmin (explained in the following section) allows us to use this algorithm in a non-standard situation.

2.2. The model & multiple vehicles case

The key assumption on which we base our method is that cars can be represented by rectangular cuboids, all axis aligned to a coordinate system, in which the XY plane corresponds to the surface of the road, with the X axis pointing in the general direction of movement and Z axis pointing upwards. Furthermore, we assume that representation of these cuboids on the video follows the rules of a Cavalier projection. A non-right angle between the X axis and the non-aligned axis of the projection is marked as α and is the first of the configuration parameters that the user must provide.

At this point, additional steps are required in cases of multiple cars overlapping, as standard methods of background subtraction do not provide any solution. To solve this problem two sets of straight line are generated.

The first set consists of parallel straight lines generated from the slope-intercept form y = mx +b, for x ranging from x_1 to x_2 + y_1, slope m calculated as a tangent of provided angle α, and b equal to y_2/mx. The second set is generated in a similar manner, the only difference being the slope of lines, equal to 1/m.

For each line in the first set, the number of foreground pixels on the line is calculated, and, if the number of points differs significantly between two consecutive lines, the line is added to a list of lines possibly containing a split point. The procedure is then repeated for the second set. The points where lines from both lists intersect are calculated. All points that lie within the blob (foreground points) are discarded, and the remaining points are sorted by their x coordinate. The final set of lines is generated as a line crossing each pair of two consecutive points on the list. The blob is then split in two, one consisting of points that fall below all of the lines, and the second consisting of all points not meeting that criterion.

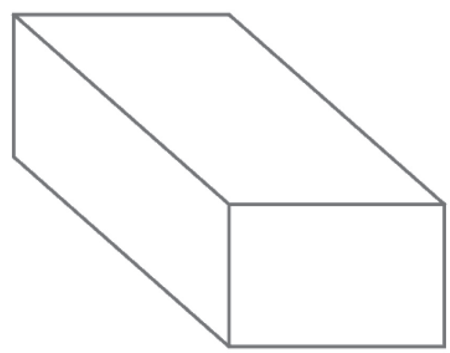

Fig. 1. Cavalier projection [own study]

As seen in Figure 1, a rectangular cuboid in a Cavalier projection consists of three pairs of parallel segments of the same length. To find these segments, we first find pairs of parallel lines tangent to the blob. Vertical and horizontal lines are trivial to find, as they correspond to boundaries of a 2D axis aligned bounding box obtained earlier. The final pair of lines is firstly defined in the slope-intercept form y = mx +b. The value of slope m was previously calculated and is the same here. Iterating over all points belonging to the blob value of intercept b for each point can be calculated. We mark the maximal and minimal value of the intercept as b_{max} and b_{min} respectively. Lines of slope m and intercepts b_{max} and b_{min} are tangent to the blob.

2.3. Calculating pixel dimensions

After calculating all six lines, we define points A through F as intersections of certain pairs of lines. These definitions vary in respect to the sign of the tangent of provided angle α and can be seen in Figures 2 and 3.

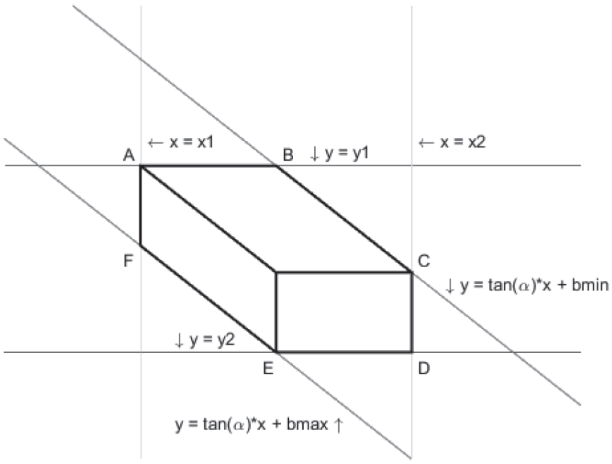

Fig. 2 Point definitions for tan(α) > 0 [own study]

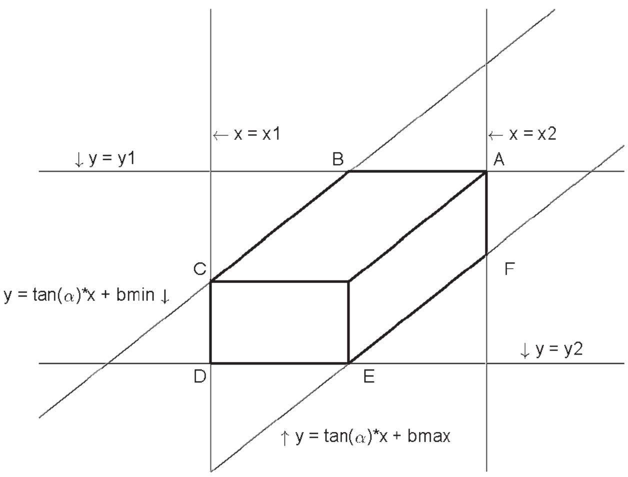

Fig. 3 Point definitions for tan(α) < 0 [own study]

As cars are not all rectangular cuboids, it is not guaranteed that parallel segments of our 3D bounding box will share the same length. Therefore, a method is needed to determine unique pixel dimensions for every 3D bounding box. We propose selecting the lower of the segments representing length (Distance between points E and F), the lower of the segments representing width (D and E), and a segment representing height that does not share a common point with the selected length segment (C-D). We argue that these segments depend the least on the silhouette of a car, and therefore provide the best estimate of the dimensions within our model. These dimensions are marked as x_p, y_p, z_p.

2.4. Assigning blobs to vehicles

We have a way to measure dimensions and construct bounding boxes around blobs extracted from a single frame. At this point further improvement of our detection can be achieved by discarding all blobs that do not meet certain requirements, such as a sufficient length-to-width ratio or bounding box's area. The requirements depend on the implementation. Working with a video, we need to find a way to track a car over multiple frames and correctly differentiate it from other cars. This is done by keeping track of the list of vehicles present. Each frame, all blobs that meet the requirements are assigned a point in the centre of their 2D bounding box. If that point belongs within the 2D bounding box of any vehicle on the list, it is assumed that the blob represents that vehicle. The vehicle's bounding box is updated to reflect its movement between frames. If a blob's centre point does not belong in any of the previous vehicles, it is treated as a new vehicle and added to the list.

As the final step, we define a measurement area within the video's coordinate system. For each frame, where the lowest point of the blob's 3D bounding box (marked as E in Figures 2 and 3) is within the measurement area, measured pixel dimensions are added to the vehicle list of measurements. The measurements are then sorted with respect to a volume calculated from them, and a median of the list is chosen to represent the vehicles' final pixel dimensions. The measurement area depends on the implementation, but should be relatively small. It allows us to further remove imperfections within the bounding box's construction or background subtraction, as choosing a median will filter out any gross errors.

2.5. Calibration

To calibrate the system, we have to manually recognise a car on the video and read its pixel measurements. Comparing them with the dimensions available in the manufacturer's catalogue, a scale between real-life measurements in centimetres and measurements in pixels for each dimension is found, and marked as s_x, s_y, s_z, following our axis naming convention. Applying these scales to each measured vehicle allows us to calculate the estimate of its real-life dimensions.

3. Results

During a combined length of 4 minutes of footage our algorithm recognised 38 vehicles, 17 of which we identified manually to look up their dimensions in the manufacturers' catalogues. Due to the quality of our footage and implemented background subtraction method, few cars were recognised wrongly or not at all.

The interquartile means of relative errors in all dimensions are as follows: 11.14% for length, 21.94% for width and 19.74% for height.

Below, example results are shown:

Fig. 4. Ford Mondeo IV Kombi. On the left: calculated bounding box [own study]

Car identified as a Ford Mondeo IV Kombi, with relative errors of 0.17%, 28.02%, and 30.80% respectively. This is an example of relatively good calculation of length and height, with excellent length estimation quality.

Fig. 5. Mercedes Sprinter 2006. On the left: calculated bounding box [own study]

Car identified as Mercedes Sprinter 2006, with relative errors of 10.9%, 48.9%, and 12.57% respectively. This is an example of good height estimation at the expense of width estimation quality.

4. Conclusion

The new method for estimation of vehicles' dimensions with the use of a single roadside camera has been described. This method utilises, in a novel manner, the calibration on the basis of vehicle dimensions published in manufacturers' catalogues. Such an approach has many significant advantages, among others, it does not require measurement of a camera's mounting height, knowledge of its distance to the vehicle nor lane width etc. In addition, acquiring any static object is unnecessary. The proposed method has been tested on real-world videos. While interquartile means were quite small, maximal recorded relative error was 28.83% for length, 35.54% for height and 54.22% for width. It is therefore advised to use this method primarily for length determination. Estimation in our method depends on properly marked bounding boxes. Therefore, some aspects, such as shadows of vehicles must be taken into account as they may seriously corrupt blobs and as a result bounding boxes may be too big. Examples provided in section 3 show how inherent flaws of the assumed model translate to errors. While length estimation shows no major issues, a correlation can be seen between height and width errors. Depending on the proportions of the car and the angle between the road and the camera, our method tends to "favour" one dimension over the other. Therefore, it is important that the camera angle is neither close to zero nor the right angle. However, in general, the magnitude of height error for good width estimations is considerably lower than the magnitude of width error in cases where height estimation was precise.

It is also worth noting that modern ITS systems collect information from many cameras, which allows for the easy selection of the best angled one. When video from the already mounted camera is utilised, the implementation cost of the proposed method is minute. The method is easy to calibrate and implement in real-world scenarios.

Acknowledgements

The work in this paper was partially financed from grant 0401/0230/16.

Bibliography

[1] BUCH, N., VELASTIN, S.A., ORWELL, J.: A Review of Computer Vision Techniques for the Analysis of Urban Traffic, IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 3, Sept. 2011

[2] ZHANG, J., et al.: Data-driven intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, vol. 12 no.4, pp. 1624-1639.

[3] BAZAN, M., et al.: Detection of vehicles moving in the wrong direction, Archives of Transport System Telematics. 2016, vol. 9, no 1, pp. 3-9.

[4] BAZAN, M., et al.: Green wave optimisation, Archives of Transport System Telematics. 2016, vol. 9, no. 3, pp. 3-8.

[5] LÄMMER, S., HELBING, D.: Self-control of traffic lights and vehicle flows in urban road networks. Journal of Statistical Mechanics: Theory and Experiment 2008.04 (2008): P04019.

[6] BAZAN, M. et al.: Road traffic predictions across major city intersections using multilayer perceptrons and data from multiple intersections located in various places, IET Intelligent Transport Systems. 2016, vol. 10, no 7, pp. 469-475

[7] BAZAN, M., et al.: Estimation of travel time in the city based on intelligent transportation system traffic data with the use of neural networks, Dependability Engineering and Complex Systems : proceedings of the Eleventh International Conference on Dependability and Complex Systems DepCoS-RELCOMEX, Springer.

[8] KAEHLER, A., BRADSKI, G.: Learning OpenCV 3 Computer Vision in C++ with the OpenCV Library, O'Reilly Media, 2016

[9] [On-line] OpenCV Documentation Available: http://opencv.org/documentation.html [date of access: 20.01.2016]

[10] MASOUD, O., PAPANIKOLOPOULOS, N.: Using geometric primitives to calibrate traffic scenes, in Proc. IEEE/RSJ Intl. Conf. Intelligent Robots and Systems, vol. 2, 2004, pp. 1878-1883.
[11] FEI-YUE, W.: A simple and analytical procedure for calibrating extrinsic camera parameters. IEEE Transactions on Robotics and Automation 20.1 (2004): 121-124.
[12] DUBSKÁ, M., HEROUT, A., SOCHOR, J.: Automatic Camera Calibration for Traffic Understanding. BMVC. 2014
[13] NEERAJ, K., BIRCHFIELD, S., SARASUA, W.: Automatic camera calibration using pattern detection for vision-based speed sensing." Transportation Research Record: Journal of the Transportation Research Board 2086 (2008): 30-39.
[14] KUNFENG, W., et al. Research on lane-marking line based camera calibration. Vehicular Electronics and Safety, 2007. ICVES. IEEE International Conference on. IEEE, 2007.
[15] WANG, J.M., et al.: Shadow detection and removal for traffic images. In Networking, Sensing and Control, 2004 IEEE International Conference on (Vol. 1, pp. 649-654). IEEE.