Automatic Camera Calibration for Traffic Understanding

Markéta Dubská[1]
idubska@fit.vutbr.cz
Jakub Sochor[1]
isochor@fit.vutbr.cz
Adam Herout[12]
herout@fit.vutbr.cz

[1] Graph@FIT
Brno University of Technology, CZ
[2] click2stream, Inc.

Abstract

We propose a method for fully automatic calibration of traffic surveillance cameras. This method allows for calibration of the camera – including scale – without any user input, only from several minutes of input surveillance video. The targeted applications include speed measurement, measurement of vehicle dimensions, vehicle classification, etc. The first step of our approach is camera calibration by determining three vanishing points defining the stream of vehicles. The second step is construction of 3D bounding boxes of individual vehicles and their measurement up to scale. We propose to first construct the projection of the bounding boxes and then, by using the camera calibration obtained earlier, create their 3D representation. In the third step, we use the dimensions of the 3D bounding boxes for calibration of the scene scale. We collected a dataset with ground truth speed and distance measurements and evaluate our approach on it. The achieved mean accuracy of speed and distance measurement is below 2%. Our efficient C++ implementation runs in real time on a low-end processor (Core i3) with a safe margin even for full-HD videos.

1 Introduction

Automatic visual surveillance is useful in organization of traffic – for collecting statistical data [22], for immediate controlling of traffic signals [21], for law enforcement [17, 30], etc. Existing systems typically require manual setup, often involving physical measurements in the scene of interest [13]. Our goal is to process traffic data fully automatically, without any user input. This includes assessment of camera intrinsic parameters, extrinsic parameters in relation to the stream of traffic, and scale of the ground plane which allows for measurement in the real world units – Fig. 1.

Some existing works in automatic traffic surveillance require user input in the form of annotation of the lane marking with known lane width [32] or marking dimensions [4], camera position [24, 32], average vehicle size [6, 29] or average vehicle speed [25]. A common feature of virtually all methods is detection of the vanishing point corresponding to the direction of moving vehicles (full camera calibration requires three orthogonal vanishing points [2, 5, 8]). A popular approach to obtaining this VP is to use road lines [9, 26, 35] or lanes

© 2014. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

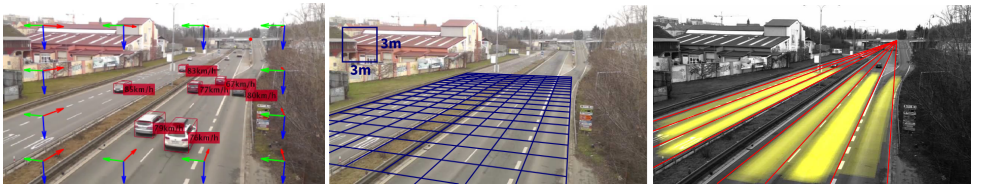

Figure 1: We automatically determine 3 orthogonal vanishing points, construct vehicle bounding boxes *(left)*, and automatically determine the camera scale by knowing the statistics of vehicle dimensions. This allows us to measure dimensions and speed *(middle)* and analyze the traffic scene *(right)*.

[7, 12, 26], more or less automatically extracted from the image. These approaches typically require a high number of traffic lanes and a consistent and well visible lane marking. Another class of methods disregard the line marking on the road (because of its instability and impossible automatic detection) and observe the motion of the vehicles, assuming straight and parallel trajectories in a dominant part of the view. Schoepflin and Dailey [25] construct an activity map of the road with multiple lanes and segment out individual lanes. Again, this approach relies on observing a high number of traffic lanes – high-capacity motorways and similar settings. Other researchers detect vehicles by a boosted detector and observe their movement [19], or analyze edges present on the vehicles [34]. Beymer et al. [1] accumulate tracked feature points, but also require the user to provide defined lines by manual input. Kanhere and Birchfield [18] took a specific approach for cameras placed low above the street level. Once the calibration and scene scale is available, the road plane can be rectified and various applications such as speed measurement can be done [3, 4, 14, 24, 36].

In our approach, we assume that the majority of vehicles move in approximately straight, mutually parallel trajectories (experiments verify that our method is tolerant to a high number of outliers from this assumption). Also, the trajectories do not have to be approximately straight across their whole span – only a significant straight part is sufficient. This makes our approach easily and reliably applicable on a vast majority of real traffic surveillance videos. The calibration of internal and external parameters of the camera is achieved by first computing three orthogonal vanishing points which define the vehicle motion [11].

Similarly to others [25, 26], we assume a pinhole camera with principal point in the image center. The principal point would be difficult (or impossible) to obtain otherwise, because the camera cannot move and no calibration pattern can be used. At the same time, this assumption does not harm the targeted applications (speed/distance measurement, traffic lane identification, ...). Unlike previous works, we do not assume exactly horizontal scene horizon [12, 14, 25]. We find this assumption too limiting and we deal with it by properly finding the second vanishing point defining the scene (Sec. 2.1). We assume zero radial distortion of the camera, but our previous work [11] offers a solution for automatic radial distortion compensation.

Once the camera intrinsic and extrinsic calibration (up to scale) defined by three orthogonal vanishing points is determined, we propose to construct 3D bounding boxes of the vehicles based on the assumption of flat ground plane. The dimensions of the 3D bounding boxes of a number of observed cars (experiments show that after processing approximately 50 cars, the scale is within 2 % from the final value) can be used for adaptation of the scale to a known distribution of car dimensions. The proposed 3D bounding boxes are easily constructed and their construction is computationally cheap. At the same time, they provide some 3D insight into the scene observed by a stationary camera, unavailable to existing ap-

proaches mentioned earlier. We are showing that once the camera calibration including scale is computed, our method allows for reasonably accurate measurement of vehicle speed and various dimensions in the scene, including 3D dimensions of passing vehicles. The bounding boxes can be used for other tasks as well – we are showing improved analysis of traffic lanes directly obtained from the geometry of the bounding boxes.

2 Traffic Analysis from Uncalibrated Cameras

Section 2.1 reviews our camera calibration algorithm [11]. Based on it, we propose to construct 3D bounding boxes of observed vehicles (Sec. 2.2). The dimensions of bounding boxes are statistically domain-adapted to known distribution of vehicle dimensions (Sec. 2.3) in order to obtain the scene-specific scale.

2.1 Camera Calibration from Vehicle Motion

In order to make this paper self-contained, we summarize here our calibration method [11]. It enables recovering of the focal length of the camera and its orientation with respect to the stream of traffic. It detects two originally orthogonal directions – 1st in the direction of the traffic and 2nd which is perpendicular to the 1st direction and parallel to the road. Assuming that the camera's principal point is in the center of the projection plane, the 3rd orthogonal direction and the focal length can be calculated. The detection method uses Hough transform based on the parallel coordinates [10], which maps the whole 2D projective plane into a finite space referred to as the *diamond space* by a piecewise linear mapping of lines.

For the detection of the 1st vanishing point, feature points are detected and tracked by KLT tracker in the subsequent frame. Successfully detected and tracked points exhibiting a significant movement are treated as linear fragments of vehicle trajectories. All these lines vote in the *diamond space* accumulator. The most voted point is considered to be the first vanishing point. Figure 2 (*left*) shows the tracked points accumulated to the diamond space.

The second vanishing point corresponds to the direction parallel to the road (or the ground plane) and is perpendicular to the first direction. Again, the diamond space [10] is

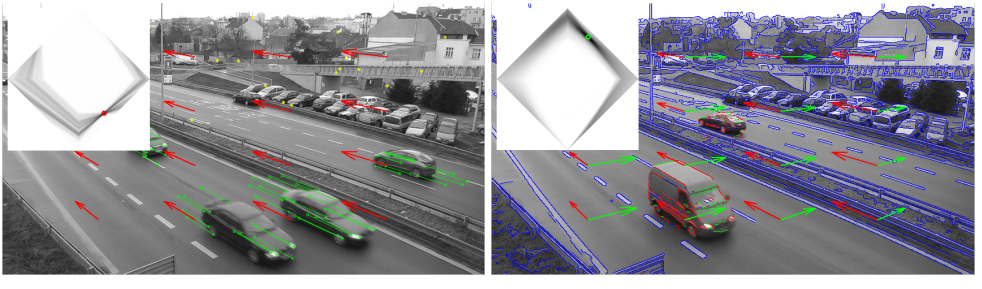

Figure 2: (*left*) Tracked points used for estimation of the 1st VP. Points marked by green exhibit a significant movement and they are accumulated. Points marked by yellow are stable points and do not vote. The accumulated diamond space is in the top left corner. (*right*) Accumulation of the 2nd vanishing point. Blue edges belong to the background. Red edges are omitted from voting because of their vertical direction or direction towards the first VP. Green edges are accumulated to the diamond space (in the top left corner; green circle marks the maximum).

used for its detection. Many edges on the vehicles coincide with the second vanishing point and thus we let them vote in the accumulation space. An edge background model is used in order to select only edges on moving objects – probable vehicles. The model is updated by each frame to deal with shadows and other slow changes. The edge background model stores for each pixel the confidence score of occurrence of an oriented edge (eight bins are used to store likelihoods for different orientations). The edges passing the background test are further processed and filtered. The first vanishing point is known from the previous processing and edges supporting this VP are excluded from accumulation. Also the edges with approximately vertical direction are omitted from voting, based on the assumption of scene horizon being approximately horizontal (with a high tolerance, e.g. $\pm 45^\circ$). This condition can be disregarded when the first VP is detected to be close to infinity. In such a case, edges supporting the second VP are allowed to have vertical direction. Figure 2 (*right*) shows the edge background model, omitted and accumulated edges together with the diamond space.

2.2 Construction of 3D Bounding Boxes

The next step of our approach is construction of 3D bounding boxes of the observed vehicles (see Fig. 3 (IV) for an example). We assume that vehicle silhouettes can be extracted by background modeling and foreground detection [27, 37]. Detection of foreground blobs for vehicles can be done reliably, including removal of shadows [15]. Further we assume that the vehicles of interest are moving from/towards the first vanishing point (Sec. 2.1). In fact, all detected foreground blobs in the input video are filtered by this criterion, which leads to disposal of invalid blobs.

Our approach is based on an observation, that vehicle blobs tend to have some edges very stable and reliable. Refer to Fig. 3 for an illustration where the detected blob of the car is colored and rest of the image is desaturated. In the given situation, red lines pass through the 1st VP and they are tangent to the vehicle's blob. Green lines are blob's tangents coinciding with the 2nd VP; blue tangents pass through the 3rd VP. The two tangents corresponding to the VP are lines with minimal and maximal orientation passing thought the VP and the points from convex hull of the blob.

Because the blobs are not accurate and the cars are not exactly boxes, the fitting of the bounding box is ambiguous, i.e. the order in which the tangents and their intersections are extracted matters. We propose the following order, which appears to be the most stable one. Firstly, point A is constructed as the intersection of the lower red and green tangent. Then, points B and C are defined by intersections of the lower green tangent with right blue and the lower red with left blue, respectively, Fig. 3 (I). Constructed line segments AB and AC define the shorter and the longer side of the box base. Point D lies on the intersection of the upper green tangent and the left blue tangent. Together with the line passing through point A and

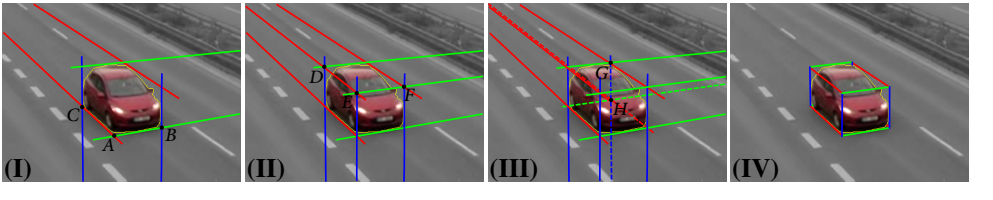

Figure 3: Construction of vehicle's 3D bounding box. **(I)** Tangent lines and their relevant intersections A, B, C. **(II)** Derived lines and their intersections E, D, F. **(III)** Derived lines and intersection H. **(IV)** Constructed bounding box.

the 3rd VP it uniquely defines point E, Fig. 3 (II). Point E can be also constructed using point F – leading to an alternative position of point E. We choose point E with the larger distance $|AE|$, which ensures that the whole blob will be enclosed in the bounding box. With known F and D, point G is the intersection of the line through D and 2nd VP with line through F and 1st VP, Fig. 3 (III).

When the configuration of the vanishing points with respect to the center of the foreground blob is different from the one discussed in the previous paragraphs, the set and order of used tangent lines and points slightly changes. The change is self-evident and follows the principles sketched above. Figure 4 shows other possible orientations of the bounding box with respect to different configurations of VPs.

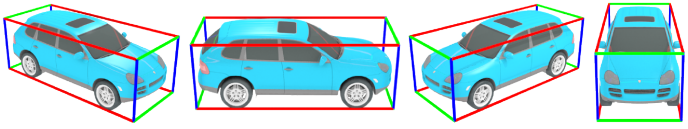

Figure 4: Different bounding boxes depending on positions of the vanishing points with respect to the camera. Because of rounded corners of the car, the edges of the bounding box would not fit tight to the car. However, in most cases, at least one dimension fits tight and this is enough to find the scale.

Because the roof and sides of the car are typically somewhat bent, the detected bounding box can be slightly smaller that in reality. However, we count with this inaccuracy in the domain adaptation procedure and prefer the best matching pair of bounding box sides for further computation. The experiments show that the final accuracy is not harmed by the slightly diminished detected bounding boxes (Sec. 3). In order to be able to determine the vehicles dimensions accurately, shadows need to be removed from the detected foreground objects. Elaborate shadow removal exceeds the scope of our work, but it has been addressed by other researchers [31, 33]. In our work, we assume only the presence of soft shadows and we use the method of Horprasert et al. [15] for their removal.

2.3 Statistical Domain Adaptation of Vehicle Dimensions

Having the bounding box projection, it is directly possible to calculate the 3D bounding box dimensions (and position in the scene) up to precise scale (Figure 5). We consider a three-dimensional coordinate system with camera coordinates $O = [p_x, p_y, 0]$, center of the projection plane $P = [p_x, p_y, f]$ (where $[p_x, p_y]$ is the principal point) and three orthogonal directions derived from the detected vanishing points in the image. Firstly, plane $\wp$ parallel to the road ground plane is constructed – its orientation is known since the direction of the 3rd VP is perpendicular to this plane; its distance from the camera is chosen arbitrarily. Figure 5 shows two possible placements of the plane and the influence of such placement – the closer the plane is to the camera, the smaller the objects appear. The detected corners of the bounding box (points A, B, C, E) are projected to the plane:

$$\begin{aligned} &A_w = \wp \cap \overleftrightarrow{OA}, \quad B_w = \wp \cap \overleftrightarrow{OB}, \quad C_w = \wp \cap \overleftrightarrow{OC}, \\ &E_w = p_E \cap \overleftrightarrow{OE};\ p_E \perp \wp \wedge A_w \in p_E. \end{aligned} \tag{1}$$

When the world coordinates of the bounding box corners are known, it is possible to determine the (somehow scaled) dimensions of the box: $(l, w, h) = (|A_w C_w|, |A_w B_w|, |A_w E_w|)$.

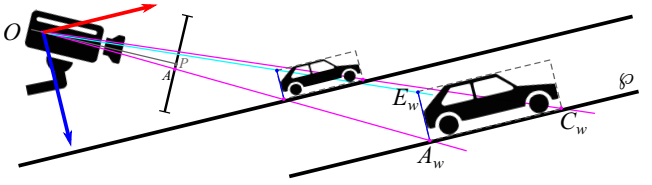

Figure 5: Calculation of the world coordinates. Plane ℘ is parallel to the road and it is derived from the detected VPs. Its distance is selected arbitrarily and the precise scale is found later, Fig. 6. The camera is placed in $O = [p_x, p_y, 0]$ and world points of the base of the bounding box are intersections of plane ℘ with rays from O through points A, C (constructed earlier in the projection plane). Other points are intersections of rays from O through projected points and rays perpendicular to ℘ passing through points A_w, B_w, C_w, H_w.

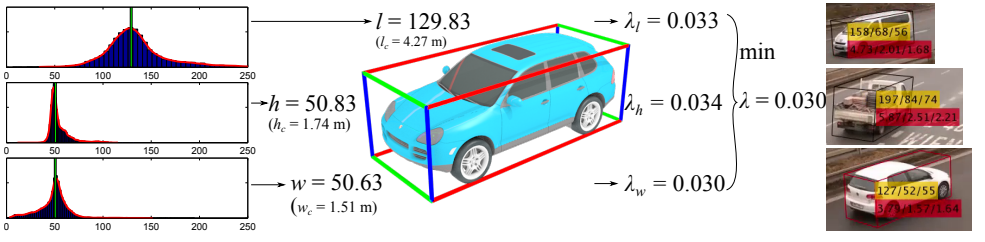

Figure 6: Calculation of scene scale λ. *(left)* Median (green bar) for each dimension is found (l, w, h) in the measured data. *(middle)* Scales are derived separately based on known median car size (l_c, w_c, h_c) as $\lambda_l = l_c/l; \lambda_w = w_c/w; \lambda_h = h_c/h$. The final scale is the minimum from these three scales. *(right)* Examples of relative size of the vehicles (yellow) and real dimensions in meters after scaling by factor λ (red).

Scale factor λ must be found so that the actual metric dimensions are defined as $(\mathbf{l}, \mathbf{w}, \mathbf{h}) = \lambda(l, w, h)$. For this purpose, we collect statistical data about sold cars and their dimensions and form a histogram of their bounding box dimensions. Relative sizes of the cars (l, w, h) are accumulated into a histogram as well. Histograms confirm the intuitive assumption that vehicles have very similar width and height (peaks in histograms are more significant) but they differ in their length. By fitting the statistics of known dimensions and the measured data from the traffic, for each dimension we obtain a scale (Fig. 6). In an ideal case, all these scales are equal. However, because different influences of perspective and rounded corners of the cars (Fig. 4), they are not absolutely the same. For the final scale λ, we choose the smallest of the scales. The motivation here is that the detected bounding boxes tend to be smaller (and therefore the scale λ is greater) because cars are not perfectly boxed and from specific views, some edges of the bounding box did not fit tightly to the car (see Fig. 4).

3 Experimental Evaluation

Our method presented here allows for automatic obtaining camera intrinsic and extrinsic parameters, including the scene scale on the ground plane. This allows for multiple applications, previously unavailable without entering human calibration input. This section evaluates the accuracy relevant to the most straightforward applications: Distance measurements, speed measurements (Sec. 3.1), and analysis of traffic lanes (Sec. 3.2). Section 3.3

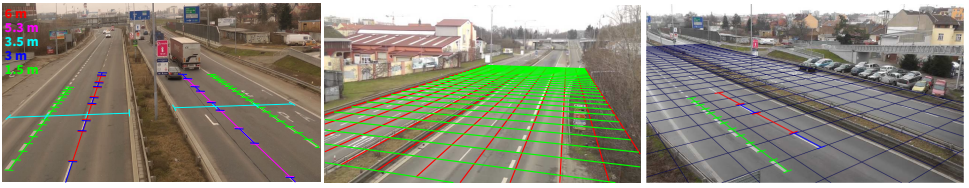

Figure 7: *(left)* Scene with measured ground truth distances used for accuracy evaluation. *(middle)* Grid projected to the road (i.e. ground plane). The size of the squares is $3m \times 3m$. *(right)* Different view of a scene with detected ground plane with $3m \times 3m$ squares and some of the measured ground truth distances.

shows that the algorithm is capable of running in real time on a low-end processor. Figure 9 shows example images of achievable results.

3.1 Distance & Speed Measurement

Figure 7 *(middle)* shows a uniform grid with square $3m \times 3m$ placed over the ground plane. We measured several distances on the road plane, Fig. 7 *(left)*, and evaluated error in distance measurements by our approach. This evaluation is similar to the work of Zhang et al. [34]; however, we evaluate the absolute dimension in meters, while Zhang et al. evaluate relative distances supposed to be equal. They report average error of measurement "less then 10%". Our average error is 1.9% with worst case 5.6%. Table 1 shows results on five videos observing the same scene from different viewpoints.

When measuring the vehicle speed (Tab. 2), we take into account corner A of the bounding box, which lies directly on the road (this is an arbitrary choice – any point from the box base can be used). Vehicles in the video are tracked and their velocity is evaluated over the whole straight part of the track. It is also possible to calculate instant speed of the vehicle as the distance vehicle passes between subsequent video frames, but it is not perfectly stable because of inaccuracies in detection of the bounding box and image discretization. It should be noted that once the camera is calibrated including the scale, for computing the average speed of a vehicle, its blob segmentation does not need to be very precise, because even though a part of the vehicle is missing, the speed measurements are still accurate.

The average speed of the vehicle was $75\frac{km}{h}$ and therefore 2% error causes $\pm 1.5\frac{km}{h}$ deviation. A similar evaluation was provided by Dailey [6] who used distribution of car lengths for

	1.5 m	3 m	3.5 m	5.3 m	6 m	all
v1	2.0/3.3 (29)	2.1/3.9 (7)	4.5/5.5 (3)	3.1/5.6 (5)	2.1/2.4 (3)	2.3 /5.6 (47)
v2	1.6/2.3 (15)	1.3/2.4 (7)	1.3/2.3 (3)	3.3/3.3 (2)	0.7/.17 (3)	1.5/ 3.3 (30)
v3	1.9/3.5 (13)	2.5/3.2 (6)	1.0/1.6 (3)	2.7/3.0 (3)	2.7/3.3 (3)	2.1/ 3.3 (28)
v4	1.0/1.9 (13)	1.8/3.5 (6)	2.3/3.1 (3)	3.7/5.3 (3)	0.9/2.0 (3)	1.6/ 5.3 (28)
v5	2.4/3.6 (15)	1.0/2.5 (6)	0.9/1.7 (3)	1.5/2.5 (3)	1.1/1.7 (3)	1.7/ 3.6 (30)
all	1.8/3.6 (85)	1.7/3.9 (32)	2.0/5.5 (15)	2.8/5.6 (16)	1.5/3.3 (15)	**1.9**/**5.6**(163)

Table 1: Percentage error of absolute distance measurements (5 videos). The error is evaluated as $|l_m - l_{gt}|/l_{gt} * 100\%$, where l_{gt} is ground truth value and l_m is distance measured by presented algorithm. For each video and each distance we evaluate the average and worst error. The number in parentheses stands for the number of measurements of the given length. The bold numbers are average and worst error over all videos and all measurements.

	v1 (5)	v2 (3)	v3 (5)	v4 (5)	v5 (5)	v6 (5)	all (28)
mean	2.39	2.90	1.49	1.65	1.31	2.58	**1.99**
worst	3.47	3.63	3.18	3.77	2.40	4.26	**4.26**

Table 2: Percentage error in speed measurement (6 videos). For obtaining the ground truth values, we drove cars with cruise control and get the speed from GPS. The error is evaluated as $|s_m - s_{gt}|/s_{gt} * 100\%$, where s_{gt} is speed from GPS and s_m is speed calculated by presented algorithm. The number in parentheses stands for the number of evaluated measurements.

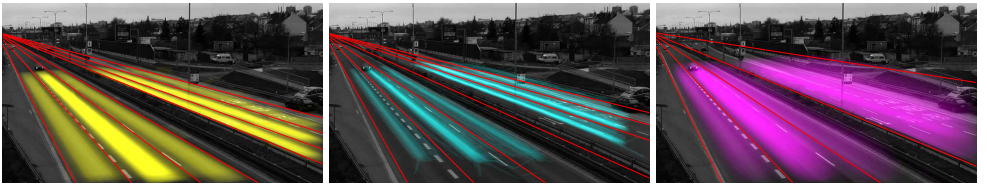

Figure 8: Traffic lane segmentation. *(left)* Our approach based on 3D bounding boxes. Lanes are correctly segmented even for side views. *(middle)* Method using trajectories of the centers of blobs [16]. *(right)* Method based on activity map [28].

scale calculation and reached average deviation $6.4\frac{km}{h}$ (assuming on-screen vertical motion and measuring the projected lengths of cars in pixels) or by Grammatikopoulos [13] whose solution has reported accuracy $\pm 3\frac{km}{h}$ but requires manual distance measurements.

3.2 Detection of Traffic Lanes

Having the 3D vehicle bounding boxes, it is also possible to obtain accurate segmentation of traffic lanes, even from views where cars from one lane overlap ones from another. Existing methods accumulate trajectories of the blobs [16], the whole blobs, pixels different to background model [25, 28] or lines on the road [20]. All these methods tend to fail when the camera views the road from side. In our approach, for each vehicle's trajectory we accumulate a filled quad strip with quad vertices $A_i, B_i, A_{i+1}, B_{i+1}$, where i denotes points in i-th video frame. After accumulation, minima are found on the line perpendicular to road direction (i.e. line passing through the 2^{nd} VP) and these are set to be lanes' borders. Accumulation of the above mentioned quad is suitable for finding the borders between the lanes. In some cases, centers of lanes (locations with dominant vehicle movement) are of interest – in that case, only trajectories of a center point in the vehicle base (e.g. $(A_i + B_i)/2$) are accumulated. Figure 8 shows a comparison of different lane segmentation methods with our approach based on projection of "correct" bounding boxes.

3.3 Computational Speed

We created an efficient C++ implementation of the proposed algorithm and evaluated the computational speed on 195 minutes of video (using Intel i3-4330 3.50 GHz processor and 8 GB DDR3 RAM). The measured framerates also include reading and decompression of videos (considerable load for full-HD videos). It should be noted that optimal framerate for running the detection/tracking algorithm is around 12.5 FPS, because the cars must move measurably from one frame to the next one. Therefore, "real-time processing" in this case means running faster than 12.5 FPS. The results in Tab. 3 show that the system can work in real time with a safe margin.

resolution	low traffic intensity	high traffic intensity
854×480	116.93 FPS	93.79 FPS
1920×1080	24.98 FPS	19.64 FPS

Table 3: Results of processing speed measurement. High traffic: ~ 40 vehicles per minute; low traffic: ~ 3.5 vehicles per minute. It should be noted that the system uses video streams with ~ 12.5 FPS; and therefore, it can run safely in real time even for full-HD video with the high traffic intensity.

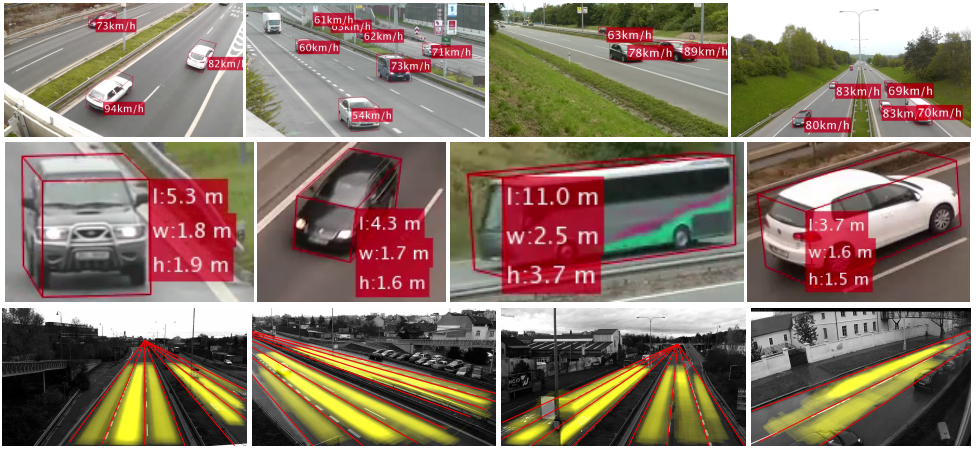

Figure 9: Examples of achieved results (see supplementary video for further illustration). (1^{st} *row*) Different scenes with measured vehicle speed. (2^{nd} *row*) Cropped out vehicles with estimated dimensions. (3^{rd} *row*) Road lanes detected using 3D bounding boxes.

4 Conclusions and Future Work

We presented a method for understanding traffic scenes observed by stable roadside cameras. Our method is fully automatic – no user input is required during the whole process. Experimental results show that the mean error of speed and distance measurement is below 2 % (worst 5.6 % for distance and 4.3 % for speed). This outperforms existing approaches and provides sufficient accuracy for statistical traffic analysis. Besides measurement, our approach can facilitate other traffic analysis task, as shown on the case of traffic lane segmentation. The algorithm works in real time with a safe margin. Our measurements show that the system is able to process 93 FPS of normal video input. The extracted bounding boxes can be used for various traffic analyses – on the example of traffic lane segmentation we are showing its benefits for traffic scene understanding.

We are exploring ways how to use the bounding boxes for facilitating various computer vision tasks. Their knowledge can improve and/or speed up scanning window-based detection and recognition algorithms. Our bounding boxes can serve as a starting point for fitting of detailed 3D models to the visual data [23]. We are also working on a multi-camera system resistant to mutual occlusions of vehicles – the bounding boxes constructed by multiple cameras can be cheaply fused into one stream of results.

Acknowledgements This work was supported by TACR project V3C, TE01020415.

References

[1] D. Beymer, P. McLauchlan, B. Coifman, and J. Malik. A real-time computer vision system for measuring traffic parameters. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 1997. doi: 10.1109/CVPR.1997.609371.

[2] Bruno Caprile and Vincent Torre. Using vanishing points for camera calibration. *International Journal of Computer Vision*, 4(2):127–139, 1990.

[3] F.W. Cathey and D.J. Dailey. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In *Intelligent Vehicles Symposium*, pages 777–782, 2005. doi: 10.1109/IVS.2005.1505199.

[4] F.W. Cathey and D.J. Dailey. Mathematical theory of image straightening with applications to camera calibration. In *Intelligent Transportation Systems Conference*, 2006. doi: 10.1109/ITSC.2006.1707413.

[5] Roberto Cipolla, Tom Drummond, and Duncan P Robertson. Camera calibration from vanishing points in image of architectural scenes. In *British Machine Vision Conference, BMVC*, 1999.

[6] D.J. Dailey, F.W. Cathey, and S. Pumrin. An algorithm to estimate mean traffic speed using uncalibrated cameras. *IEEE Transactions on Intelligent Transportation Systems*, 1(2):98–107, 2000. ISSN 1524-9050. doi: 10.1109/6979.880967.

[7] Bart De Schutter and Bart De Moor. Optimal traffic light control for a single intersection. *European Journal of Control*, 4(3):260–276, 1998.

[8] J. Deutscher, M. Isard, and J. MacCormick. Automatic camera calibration from a single manhattan image. In *European Conference on Computer Vision, ECCV*, pages 175–188. 2002. ISBN 978-3-540-43748-2. doi: 10.1007/3-540-47979-1-12. URL http://dx.doi.org/10.1007/3-540-47979-1-12.

[9] Rong Dong, Bo Li, and Qi-mei Chen. An automatic calibration method for PTZ camera in expressway monitoring system. In *World Congress on Computer Science and Information Engineering*, pages 636–640, 2009. ISBN 978-0-7695-3507-4. doi: 10.1109/CSIE.2009.763. URL http://dx.doi.org/10.1109/CSIE.2009.763.

[10] Markéta Dubská and Adam Herout. Real projective plane mapping for detection of orthogonal vanishing points. In *British Machine Vision Conference, BMVC*, 2013.

[11] Markéta Dubská, Adam Herout, Jakub Sochor, and Roman Juránek. Fully automatic roadside camera calibration for traffic surveillance. *Accepted for publication in IEEE Transactions on Itelligent Transportation Systems*.

[12] George S. K. Fung, Nelson H. C. Yung, and Grantham K. H. Pang. Camera calibration from road lane markings. *Optical Engineering*, 42(10):2967–2977, 2003. doi: 10.1117/1.1606458. URL http://dx.doi.org/10.1117/1.1606458.

[13] Lazaros Grammatikopoulos, George Karras, and Elli Petsa. Automatic estimation of vehicle speed from uncalibrated video sequences. In *Proceedings of International Symposium on Modern Technologies, Education and Profeesional Practice in Geodesy and Related Fields*, pages 332–338, 2005.

[14] Xiao Chen He and N. H C Yung. A novel algorithm for estimating vehicle speed from two consecutive images. In *IEEE Workshop on Applications of Computer Vision, WACV*, 2007. doi: 10.1109/WACV.2007.7.

[15] T. Horprasert, D. Harwood, and L. S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. IEEE ICCV*, volume 99, pages 1–19, 1999.

[16] Jun-Wei Hsieh, Shih-Hao Yu, Yung-Sheng Chen, and Wen-Fong Hu. Automatic traffic surveillance system for vehicle tracking and classification. *Intelligent Transportation Systems, IEEE Transactions on*, 7(2):175–187, 2006.

[17] Shunsuke Kamijo, Yasuyuki Matsushita, Katsushi Ikeuchi, and Masao Sakauchi. Traffic monitoring and accident detection at intersections. *Intelligent Transportation Systems, IEEE Transactions on*, 1(2):108–118, 2000.

[18] N. K. Kanhere and S. T. Birchfield. Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features. *IEEE Transactions on Intelligent Transportation Systems*, 9(1):148–160, 2008. ISSN 1524-9050. doi: 10.1109/TITS.2007.911357. URL http://dx.doi.org/10.1109/TITS.2007.911357.

[19] Neeraj K Kanhere, Stanley T Birchfield, and Wayne A Sarasua. Automatic camera calibration using pattern detection for vision-based speed sensing. *Journal of the Transportation Research Board*, 2086(1):30–39, 2008.

[20] Andrew HS Lai and Nelson HC Yung. Lane detection by orientation and length discrimination. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 30(4):539–548, 2000.

[21] Stefan Lämmer and Dirk Helbing. Self-control of traffic lights and vehicle flows in urban road networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008 (04), 2008. doi: 10.1088/1742-5468/2008/04/P04019.

[22] J de Ortuzar and Luis G Willumsen. *Modelling transport*. 2011.

[23] A. Ottlik and H.-H. Nagel. Initialization of model-based vehicle tracking in video sequences of inner-city intersections. *International Journal of Computer Vision*, 80(2): 211–225, 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0112-6. URL http://dx.doi.org/10.1007/s11263-007-0112-6.

[24] Tun-Wen Pai, Wen-Jung Juang, and Lee-Jyi Wang. An adaptive windowing prediction algorithm for vehicle speed estimation. In *IEEE Intelligent Transportation Systems*, 2001. doi: 10.1109/ITSC.2001.948780.

[25] T.N. Schoepflin and D.J. Dailey. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. *IEEE Transactions on Intelligent Transportation Systems*, 4(2):90–98, 2003. ISSN 1524-9050. doi: 10.1109/TITS.2003.821213.

[26] Kai-Tai Song and Jen-Chao Tai. Dynamic calibration of Pan–Tilt–Zoom cameras for traffic monitoring. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 36(5):1091–1103, 2006. ISSN 1083-4419. doi: 10.1109/TSMCB.2006.872271. URL http://dx.doi.org/10.1109/TSMCB.2006.872271.

[27] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *Computer Vision and Pattern Recognition*, volume 2, pages 246–252, 1999.

[28] BD Stewart, I Reading, MS Thomson, TD Binnie, KW Dickinson, and CL Wan. Adaptive lane finding in road traffic image analysis. 1994.

[29] Tuan Hue Thi, Sijun Lu, and Jian Zhang. Self-calibration of traffic surveillance camera using motion tracking. In *Proceedings of the 11th International IEEE Conference on Intelligent Transportation Systems*, 2008.

[30] David Vallejo, Javier Albusac, Luis Jimenez, Carlos Gonzalez, and Juan Moreno. A cognitive surveillance system for detecting incorrect traffic behaviors. *Expert Systems with Applications*, 36(7):10503–10511, 2009.

[31] J. M. Wang, Y. C. Chung, C. L. Chang, and S.W. Chen. Shadow detection and removal for traffic images. In *Networking, Sensing and Control, 2004 IEEE International Conference on*, volume 1, pages 649–654 Vol.1, March 2004. doi: 10.1109/ICNSC.2004.1297516.

[32] Kunfeng Wang, Hua Huang, Yuantao Li, and Fei-Yue Wang. Research on lane-marking line based camera calibration. In *International Conference on Vehicular Electronics and Safety, ICVES*, 2007. doi: 10.1109/ICVES.2007.4456361.

[33] Mei Xiao, Chong-Zhao Han, and Lei Zhang. Moving shadow detection and removal for traffic sequences. *International Journal of Automation and Computing*, 4(1):38–46, 2007. ISSN 1476-8186. doi: 10.1007/s11633-007-0038-z. URL http://dx.doi.org/10.1007/s11633-007-0038-z.

[34] Zhaoxiang Zhang, Tieniu Tan, Kaiqi Huang, and Yunhong Wang. Practical camera calibration from moving objects for traffic scene surveillance. *IEEE Transactions on Circuits and Systems for Video Technology*, 23(3):518–533, 2013. ISSN 1051-8215. doi: 10.1109/TCSVT.2012.2210670.

[35] Yuan Zheng and Silong Peng. A practical roadside camera calibration method based on least squares optimization. *Intelligent Transportation Systems, IEEE Transactions on*, 15(2):831–843, April 2014. ISSN 1524-9050. doi: 10.1109/TITS.2013.2288353.

[36] Zhigang Zhu, Bo Yang, Guangyou Xu, and Dingji Shi. A real-time vision system for automatic traffic monitoring based on 2D spatio-temporal images. In *Proceedings of WACV*, 1996.

[37] Z. Zivkovic. Improved adaptive gaussian mixture model for background subtraction. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 28–31 Vol.2, 2004. doi: 10.1109/ICPR.2004.1333992.