Traffic Surveillance Camera Calibration by 3D Model Bounding Box Alignment for Accurate Vehicle Speed Measurement

Jakub Sochor[a,*], Roman Juránek[a], Adam Herout[a]

[a] *Brno University of Technology, Faculty of Information Technology, Centre of Excellence IT4Innovations, Božetěchova 2, 612 66 Brno, Czech Republic*

Abstract

In this paper, we focus on fully automatic traffic surveillance camera calibration which we use for speed measurement of passing vehicles. We improve over a recent state-of-the-art camera calibration method for traffic surveillance based on two detected vanishing points. More importantly, we propose a novel automatic scene scale inference based on matching bounding boxes of rendered 3D models of vehicles with detected bounding boxes in the image. The proposed method can be used from an arbitrary viewpoint and it has no constraints on camera placement. We evaluate our method on recent comprehensive dataset for speed measurement BrnoCompSpeed. Experiments show that our automatic camera calibration by detected two vanishing points method reduces the error by 50 % compared to the previous state-of-the-art method. We also show that our scene scale inference method is much more precise (mean speed measurement error 1.10 km/h) outperforming both state of the art automatic calibration method (error reduction by 86 % – mean error 7.98 km/h) and manual calibration (error reduction by 19 % – mean error 1.35 km/h). We also present qualitative results of automatic camera calibration method on video sequences obtained from real surveillance cameras on various places and under different lighting conditions (night, dawn, day).

Keywords: speed measurement, camera calibration, fully automatic, traffic surveillance, bounding box alignment, vanishing point detection

1. Introduction

Surveillance systems pose specific requirements on camera calibration. Their cameras are typically placed in hardly accessible locations and the optics is focused to

*Corresponding author at BUT FIT, Božetěchova 2, 612 66 Brno, Czech Republic. Jakub Sochor is a Brno Ph.D. Talent Scholarship Holder — Funded by the Brno City Municipality.

Email addresses: isochor@fit.vutbr.cz (Jakub Sochor), ijuranek@fit.vutbr.cz (Roman Juránek), herout@fit.vutbr.cz (Adam Herout)

larger depths, making the common pattern-based calibration approaches (such as classical Zhang (2000)) unusable. That is why many solutions displace markers to the observed scene and/or measure existing geometric features (Sina et al., 2013; Do et al., 2015; You and Zheng, 2016; Luvizon et al., 2016). These approaches are laborious and inconvenient both in terms of camera setup (manually clicking on the measured features in the image) and in terms of physically visiting the scene and measuring the distances.

In our paper, we focus on *precise* and at the same time *fully automatic* traffic surveillance camera calibration including scene scale for speed measurement. The proposed speed measurement method needs to be able to deal with significant viewpoint variation, different zoom factors, various roads and densities of traffic. If the method should be applicable for large-scale deployment, it needs to run fully automatically without the necessity to stop the traffic on the road for its installation or for performing calibration measurements.

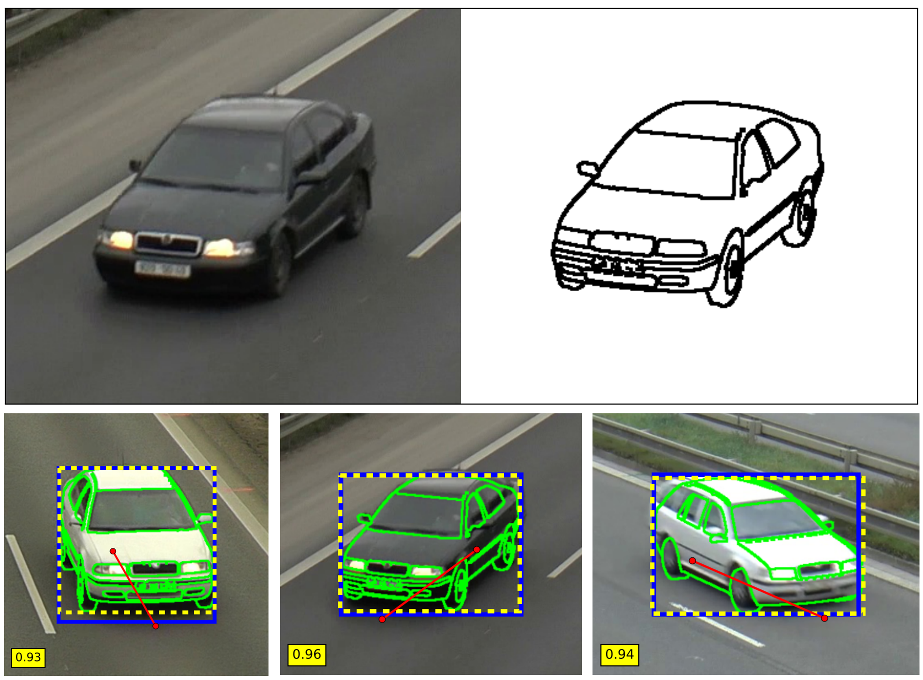

Figure 1: Example of detected vehicles and 3D model bounding box aligned to the vehicle detection bounding box. **top:** detected vehicle and corresponding 3D model (edges only), **bottom:** examples of aligned bounding boxes with shown 3D model edges (green), its bounding box (yellow) and vehicle detection (blue).

Our solution uses camera calibration obtained from two detected vanishing points. However, it is only possible to reconstruct the rotation matrix and intrinsic parameters from the vanishing points, therefore it is still necessary to obtain the scene scale. We propose to detect vehicles on the road by Faster-RCNN (Ren et al., 2015), classify them into a few common fine-grained types by a CNN (Krizhevsky et al., 2012) and use bounding boxes of 3D models for the known classes to align the detected vehicles.

The vanishing point-based calibration allows for full reconstruction of the viewpoint on the vehicle and the only free parameter in the alignment is therefore the scene scale. Figure 1 shows an example of the 3D model and the aligned images. Our experiments show that our method (mean speed measurement error 1.10 km/h) significantly outperforms existing automatic camera calibration method by Dubská et al. (2014) (error reduction by 86 % – mean error 7.98 km/h) and calibration obtained from manual measurements on the road (error reduction by 19 % – mean error 1.35 km/h). This is important because in the previous approaches, the automatism always compromised the accuracy, forcing the system developer to trade off between them. Our work shows that manual calibration (though laborious, thorough, and carried out according to state-of-the-art approaches) is inferior to the fully automatic approach based on computer vision methods.

Existing solutions for traffic surveillance camera calibration (Dailey et al., 2000; Schoepflin and Dailey, 2003; Cathey and Dailey, 2005; Grammatikopoulos et al., 2005; He and Yung, 2007b; Maduro et al., 2008; Sina et al., 2013; Nurhadiyatna et al., 2013; Dubská et al., 2014; Lan et al., 2014; Luvizon et al., 2014; Dubská et al., 2015; Do et al., 2015; Luvizon et al., 2016; You and Zheng, 2016) (see Section 2 for detailed analysis) usually have limitations for real world applications. They are either limited to some viewpoints (zero pan, second vanishing point at infinity), or they require some per-installed-camera manual work. To our knowledge, there is only one work (Dubská et al., 2014) which does not have these limitations and therefore we compare our results with this solution. For a brief description of the method, see Section 2; a more comprehensive review can be found in a recent dataset paper BrnoCompSpeed by Sochor et al. (2016b).

The key contributions of this paper are:

- Improved camera calibration method by detection of two vanishing points – camera calibration error reduced by 50 %.
- Novel method for scene scale inference significantly outperforming automatic traffic camera calibration methods (error reduced by 86 %) and also manual calibration method (error reduced by 19 %) in automatic speed measurement from a monocular camera.
- The results show that the automatic (zero human input) method can perform better than laborious manual calibration, which is generally considered accurate and treated as the ground truth. This finding can be important also in other fields than only in traffic surveillance.

2. Related Work

The camera calibration algorithm (obtaining intrinsic and extrinsic parameters of the surveillance camera) is critical for the accuracy of vehicle speed measurement by a single monocular camera, as it directly influences the speed measurement accuracy. There is a very recent comprehensive review of the traffic surveillance calibration methods (Sochor et al., 2016b), so for detailed information we refer to the review and we include only a brief description of the methods.

Several methods (He and Yung, 2007b; Cathey and Dailey, 2005; Grammatikopoulos et al., 2005) are based on detection of vanishing points as an intersection of road markings (lane dividing lines). Other methods (Dubská et al., 2014; Dubská et al., 2015; Schoepflin and Dailey, 2003; Dailey et al., 2000) use vehicle motion to calibrate the camera. Then there is also a set of methods which use some form of manually measured dimensions on the road plane (Maduro et al., 2008; Nurhadiyatna et al., 2013; Sina et al., 2013; Luvizon et al., 2014, 2016; Do et al., 2015; Lan et al., 2014).

An important attribute of the calibration methods is whether they are able to work automatically without any manual per-camera calibration input. Only two methods (Dailey et al., 2000; Dubská et al., 2014) are fully automatic and both of them use mean vehicles' dimensions for the camera calibration. Another attribute which is important for real-world deployment is whether the camera can be placed at an arbitrary position above the road, which is not true for some methods as they assume to have zero pan or another form of constraints.

Regarding fine-grained vehicle classification, there are several approaches. The first one is based on detected parts of vehicles (Krause et al., 2015; Simon and Rodner, 2015; Fang et al., 2016), another approach is based on bilinear pooling (Lin et al., 2015; Gao et al., 2016). There is also an approach based on Convolutional Neural Networks (CNN) and input modification (Sochor et al., 2016a). For an object detection, it is possible to use boosted cascades (Dollár et al., 2014), HOG detectors (Dalal and Triggs, 2005), or Deformable Parts Models (DPMs) (Felzenszwalb et al., 2010). Also, there was a recent advancement in object detection based on CNNs (Girshick et al., 2014; Ren et al., 2015; Liu et al., 2016).

Several authors dealt with alignment of 3D models and vehicles and used this technique for gathering data in the context of traffic surveillance. Lin et al. (2014) propose to jointly optimize 3D model fitting and fine-grained classification, Hsiao et al. (2014) align edges formulated as Active Shape Model (Cootes et al., 1995; Li et al., 2009). Krause et al. (2013) propose to use synthetic data to train geometry and viewpoint classifiers for 3D model and 2D image alignment. Prokaj and Medioni (2009) use detected SIFT features (Lowe, 1999) to align 3D vehicle models and the vehicle's observation. They use the alignment mainly to overcome vehicle appearance variation under different viewpoints. However, in our case as the precise viewpoint on the vehicle is known (Section 4.3), such alignment does not have to be done. Thus we adopt much simpler and more efficient method based on 2D bounding boxes – simplifying the procedure considerably without sacrificing the accuracy.

When it comes to camera calibration in general, various approaches exist. Widely used method by Zhang (2000) uses a calibration checkerboard to obtain intrinsic and extrinsic (relative to the checkerboard) camera parameters; Liu et al. (2012) use controlled panning or tilting with stereo matching to calibrate the camera. Correspondences of lines and and points are used by Chaperon et al. (2011). Yu et al. (2009) focus on automatic camera calibration for tennis videos from detected lines on the tennis court.

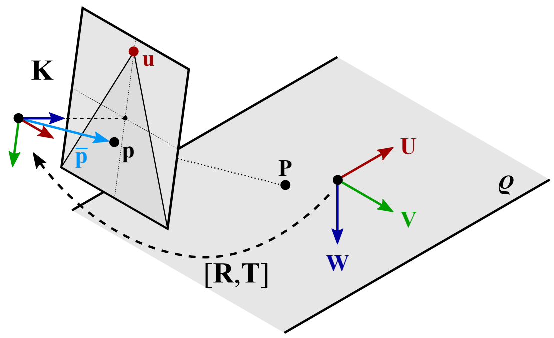

Figure 2: Camera model and coordinates. Points denoted by small letters represent points in image space while points in the world space on the road plane ρ are represented by capital letters. The representation stays the same for both finite and ideal points.

3. Traffic Camera Model

The main goal of camera calibration in the application of speed measurement is to be able to measure distances on the road plane between two arbitrary points in meters (or different units), therefore we focus only on camera model which enables to measure distance between two points on the road plane.

For convenience and better comparison of the methods, we adopt traffic camera model and notation proposed in previous papers (Dubská et al., 2014; Dubská et al., 2015); however, to make the paper self-contained, we briefly describe the model and notation. For our camera model, we assume to have zero pixel skew and principal point $\mathbf{c}$ in the center of image.

Homogeneous 2D image coordinates are referenced by bold small letters $\mathbf{p} = [p_x, p_y, 1]^T$, points on the image plane $\overline{\mathbf{p}} = [p_x, p_y, f]^T$ in 3D, where f is the focal length, are denoted by small bold letters with overline. Finally, other 3D points (on the road plane) are denoted by bold capital letters $\mathbf{P} = [P_x, P_y, P_z]^T$.

Figure 2 shows the camera model and its notation. For convenience, we assume that the origin of the image coordinate system is in the center of the image; therefore, the principal point $\mathbf{c}$ has 2D homogeneous coordinates $[0, 0, 1]^T$ (3D coordinates of the center of camera projection are $[0, 0, 0]^T$). As it is shown, the road plane is denoted by $\boldsymbol{\rho}$. We also encode vanishing points in the following way. The first one (in the direction of vehicles' flow) is referenced as $\mathbf{u}$; the second vanishing point (which has direction perpendicular to the first one and is on the road plane) is denoted by $\mathbf{v}$; and the third one (direction perpendicular to the road plane) is $\mathbf{w}$.

Using the first two vanishing points $\mathbf{u}$, $\mathbf{v}$ and the principal point $\mathbf{c}$, it is possible to compute focal length f, the third vanishing point $\mathbf{w}$, road plane normalized normal vector $\mathbf{n}$, and mainly the road plane $\boldsymbol{\rho}$. However, the road plane is computed only up to scale (as it is not possible to recover the distance to the road plane only from the vanishing points) and therefore, we add arbitrary value $\delta = 1$ as the constant term in

Equation (6).

$$
\begin{aligned}
f &= \sqrt{-\mathbf{u}^T \cdot \mathbf{v}} &\quad (1)\\
\overline{\mathbf{u}} &= [u_x, u_y, f]^T &\quad (2)\\
\overline{\mathbf{v}} &= [v_x, v_y, f]^T &\quad (3)\\
\overline{\mathbf{w}} &= \overline{\mathbf{u}} \times \overline{\mathbf{v}} &\quad (4)\\
\mathbf{n} &= \frac{\overline{\mathbf{w}}}{\|\overline{\mathbf{w}}\|} &\quad (5)\\
\boldsymbol{\rho} &= \left[\mathbf{n}^T, \delta\right]^T &\quad (6)
\end{aligned}
$$

With known road plane $\boldsymbol{\rho}$, it is possible to compute 3D coordinates $\mathbf{P} = [P_x, P_y, P_z]^T$ of an arbitrary point $\mathbf{p} = [p_x, p_y, 1]^T$ by projection on the road plane using the following equations:

$$
\begin{aligned}
\overline{\mathbf{p}} &= [p_x, p_y, f]^T &\quad (7)\\
\mathbf{P} &= -\frac{\delta}{\left[\overline{\mathbf{p}}^T, 0\right] \cdot \boldsymbol{\rho}} \overline{\mathbf{p}} &\quad (8)
\end{aligned}
$$

It is possible to measure distances on the road plane directly with 3D coordinates $\mathbf{P}$; however, as the road plane is shifted into a predefined distance by the constant term, the distance $\|\mathbf{P}_1 - \mathbf{P}_2\|$ between points $\mathbf{P_1}$ and $\mathbf{P_2}$ is not directly expressed in meters (or other units of distance). Therefore, it is necessary to introduce another calibration parameter referenced as the scene scale λ, which converts the distance $\|\mathbf{P}_1 - \mathbf{P}_2\|$ from pseudo-units on the road plane to meters by scaling the distance to $\lambda\|\mathbf{P}_1 - \mathbf{P}_2\|$.

Using the assumption of the principal point in the center of the image and zero pixel skew, it is necessary for a calibration method to compute two vanishing points ($\mathbf{u}$ and $\mathbf{v}$ in our case) together with the scene scale λ, yielding 5 degrees of freedom. Methods how to convert these camera parameters to standard intrinsic and extrinsic camera model $\mathbf{K}\,[\mathbf{R}\;\mathbf{T}]$ were discussed before in several papers (Zhang et al., 2013; Fung et al., 2003; Zheng and Peng, 2014), therefore we refer to them.

4. Camera Calibration and Vehicle Tracking

We adopted the calibration method by Dubská et al. (2014), which give the image coordinates of the vanishing points and scene scale information. We improved the method by more precise detection of the vanishing points, and we infer the scene scale by using 3D models of frequently passing cars.

Our method measures the speed of passing cars detected by Faster-RCNN (Ren et al., 2015) and tracked by a combination of background subtraction and Kalman filter (Kalman, 1960) assisted by the detector. This method, more sophisticated than the previous method (Dubská et al., 2014), gives less false positives and a comparable recall rate. In case of very dense flow when vehicles overlap each other in camera image (which does occur rarely even in real conditions), our method would miss some of the cars as we target free-flow conditions. In the following text, we describe in detail the components of the method and evaluate it in Section 5.

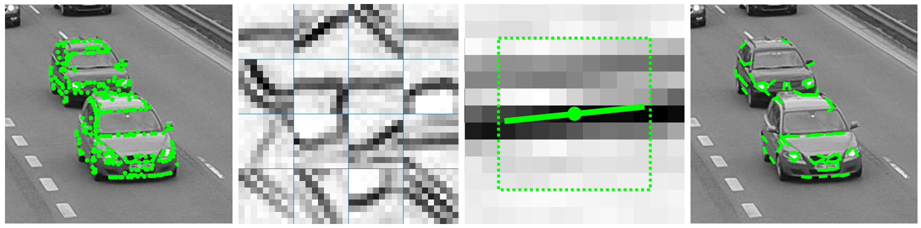

Figure 3: Visualization of edgelet detection. From left to right – Seed points $\mathbf{s}_i$ as local maxima of image gradient (foreground mask was used to filter interesting areas); Patches gatherded around the seed points from whose is computed the edge orientation; Detail of an edgelet and its orientation superimposed on the gradient image; Top 25% of edgelets detected in the image.

4.1. Vanishing Point Estimation by Edgelets

We adopted the algorithm proposed by Dubská et al. (2015) (based on detection of two orthogonal vanishing points) for the detection of the first vanishing point and propose to use a similar algorithm for detecting the second vanishing point. However, we improved the detection of the second vanishing point by using edgelets instead of image gradients used in the previous paper (Dubská et al., 2015). This change, although subtle, improves the calibration and speed measurement considerably, as the results in Section 5.3 show.

We start with detection of vanishing points from which the camera rotation with respect to the road can be estimated. The first vanishing point $\mathbf{u}$ is estimated from the movement of the vehicles by a form of cascaded Hough Transform (Dubská et al., 2015) of lines formed by tracking points of interest on the moving vehicles. This is a more stable approach than finding closest point to the lines in an algebraic way, because it is more robust to tracking noise and it is not influenced by vehicles that change lane (and therefore vanishing point of their movement is different that the rest of the vehicles). Similarly to Dubská et al. (2015), we use Min-eigenvalue point detector (Shi and Tomasi, 1994) and KLT tracker (Tomasi and Kanade, 1991).

For the detection of the second vanishing point $\mathbf{v}$, we use edges on passing vehicles as many of them coincide with $\mathbf{v}$. This step heavily relies on correct estimation of the orientation of the edges. The angle can be easily computed from gradients, but angles close to $k\pi/2$ are almost impossible to accurately recover on small neighborhoods We estimate edge orientation from a larger neighborhood by analysis of the shape of image gradient magnitude – edgelets, the detection process is shown in Figure 3.

Edgelets are detected by the following algorithm. Given an image $\mathbf{I}$, first, we find seed points $\mathbf{s}_i$ as local maxima of gradient magnitude of the image $\mathbf{E} = \|\nabla \mathbf{I}\|$, keeping only the strong ones with magnitudes above a threshold. From 9×9 neighborhood of each seed point $\mathbf{s}_i = [x_i, y_i, 1]^T$, matrix $\mathbf{X}_i$ is formed:

$$\mathbf{X}_i = \begin{bmatrix} w_1(m_1 - x_i) & w_1(n_1 - y_i) \\ w_2(m_2 - x_i) & w_2(n_2 - y_i) \\ \vdots & \vdots \\ w_k(m_k - x_i) & w_k(n_k - y_i) \end{bmatrix} \tag{9}$$

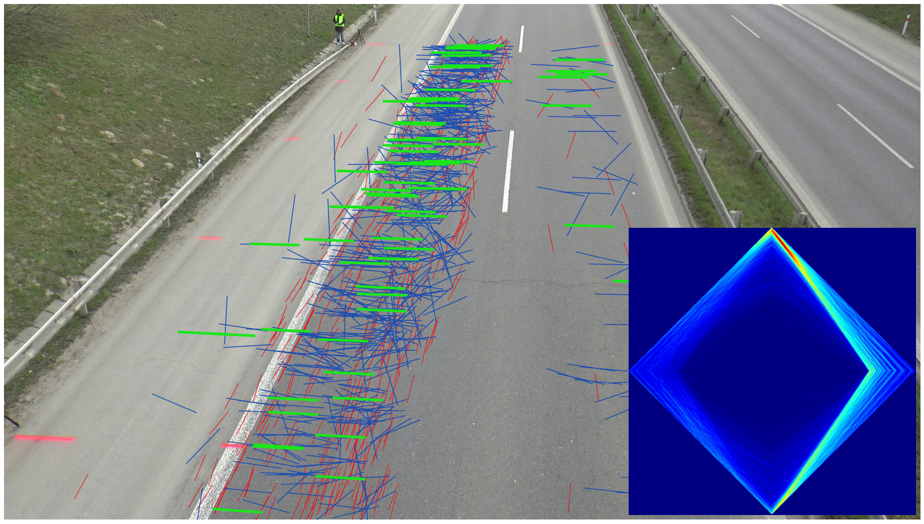

Figure 4: Visualization of edges gathered from a video – **(red)** edges that pass close to the first vanishing point, **(blue and green)** edges accumulated to the Diamond Space, and **(green)** edges supporting the detected second vanishing point. The corresponding Diamond Space is shown in bottom-right corner.

where $[m_k, n_k, 1]^T$ are coordinates of the neighboring pixels ($k = 1 \dots 81$) and w_k is their gradient magnitude from $\mathbf{E}$, i.e. for 9×9 neighborhood, the size of $\mathbf{X}_i$ is 81×2. Then, from (10), singular vectors and values of $\mathbf{X}_i$ can be computed as:

$$\mathbf{W}_i \mathbf{\Sigma}_i^2 \mathbf{W}_i^T = \mathrm{SVD}\left(\mathbf{X}_i^T \mathbf{X}_i\right), \tag{10}$$

where

$$\mathbf{W}_i = [\mathbf{a}_1, \mathbf{a}_2] \tag{11}$$

$$\mathbf{\Sigma}_i = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}. \tag{12}$$

Edge orientation is then the first singular column vector $\mathbf{d}_i = \mathbf{a}_1$ from (11), and edge quality is the ratio of singular values $q_i = \frac{\lambda_1}{\lambda_2}$ from (12). Each edgelet is then represented as a triplet $\mathcal{E}_i = (\mathbf{s}_i, \mathbf{d}_i, q_i)$.

We gather the edgelets from the input video (see Figure 4), keeping only the strong ones which do not coincide with already estimated $\mathbf{u}$, and accumulate them to the Diamond Space accumulator (Dubská and Herout, 2013). The position of the global maximum in the accumulator is taken as the second vanishing point $\mathbf{v}$. It should be noted that in this step, additional filtering can be applied – e.g. mask the Diamond Space to find only plausible solutions (avoid imaginary focal length from Equation (1)), or to find solutions within a certain range of focal lengths or horizon inclinations (when known in advance). This may improve the robustness of the second vanishing point estimation.

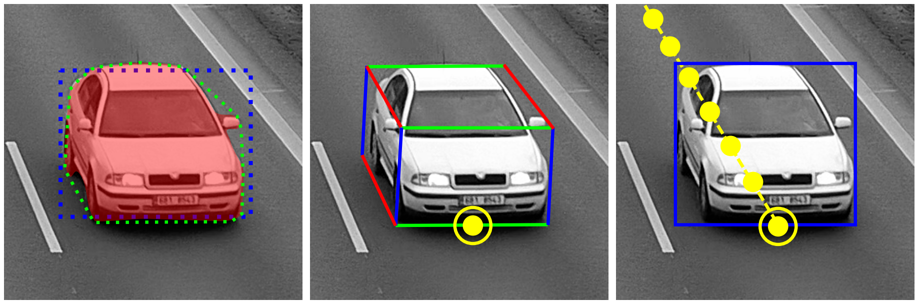

Figure 5: Car detection and tracking. From left to right: Car detected by FRCN (blue), its foreground mask and convex hull (green); 3D bounding box constructed around the convex hull and tracking point on the bottom front edge; Car bounding box (from the convex hull) tracked by Kalman filter.

4.2. Vehicle Detection and Tracking

During the speed measurement, passing cars are detected in each frame by the Faster-RCNN (FRCN) detector (Ren et al., 2015) but any detector can be used as well (e.g. ACF, LDCF (Dollár et al., 2014)). We trained the detector on COD20K dataset (Juránek et al., 2015) containing approximately 20 k car instances for training from views of surveillance nature. The detection rate of the detector is 96 % with 0.02 false positive detections per image on the test part of COD20K dataset. The detector yields a coarse information about locations of cars in the image (bounding boxes are not precisely aligned). We use a simple heuristic to remove detections that would lead to imprecise tracking and ultimately to wrong speed estimation – those that are slightly occluded by other detections and that are farther from the camera. Therefore we track only cars that are fully visible.

For the tracking, we use a simple background model that builds a background reference image by moving average. In the foreground image, compact blobs are detected and the FRCN detections are used to group those blobs that correspond to one car. From each group of blobs, convex hull and its 2D bounding box is extracted. Finally, we track the 2D bounding box of the convex hull using Kalman filter to get the movement of the car. For an example, see Figure 5.

For each tracked car, we extract a reference point for speed measurement. The convex hull is used to construct the 3D bounding box (Dubská et al., 2014) and take the center of the bottom-front edge – the reference point located in the ground/road plane. Each track is represented by a sequence of bounding boxes and reference points both constructed from the convex hull.

4.3. Scale Inference using 3D Model Bounding Box Alignment

Previous state-of-the-art automatic method for scale inference in traffic surveillance by Dubská et al. (2014) used three-dimensional bounding boxes built around the vehicle and mean dimensions of vehicles to compute the scale. However, this approach has two main drawbacks. The obvious one is in the usage of mean dimensions of vehicles. Although, the more important is not that much apparent and that is that the constructed bounding box is too tight around the vehicle and the tightness is largely influenced

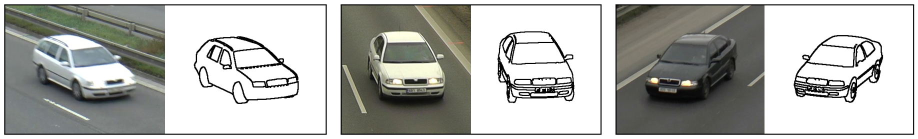

Figure 6: Examples of used 3D models (showing only edges) render under the same viewpoint as the corresponding real vehicle on the road. The left image show model which we will refer as Combi and the other two images show 3D model Sedan. Both the models are for Skoda Octavia mk1 which is common on the observed streets.

by the particular viewpoint direction. This causes systematic errors in the calibration depending on the camera location with respect to the road.

We propose to use a different approach to the scale inference, overcoming the mentioned imprecisions. We use fine-grained types of the vehicles (i.e. make, model, variant, model year) and for a few (two in our experiments) common types we obtained 3D models which are rendered to the image and align them to the real observed vehicles to obtain the proper scale.

As it is necessary to know the precise vehicle classes (up to model year) for our scale inference method, we used BoxCars dataset with such images (Sochor et al., 2016a) and we also collected some other training data from videos related to papers by Dubská et al. (2014); Dubská et al. (2015). The classification of vehicles is done only into a few most common fine-grained vehicle types on roads in the area plus one class for all the others vehicles. The full training dataset contained ∼23 k tracks and ∼92 k images of vehicles. We used a CNN (Krizhevsky et al., 2012) for the classification itself. The classification accuracy on the validation set (∼7 k of images) was 0.97. As only single instances of vehicles are classified by the CNN, we use mean probability over all of the detections belonging to one vehicle track.

For each vehicle, we also build a 3D bounding box around it (Dubská et al., 2014) to obtain the center $\mathbf{b}$ of the vehicle's base in image coordinates. To obtain the viewpoint vector $\boldsymbol{\phi}$, we first compute the rotation matrix $\mathbf{R}$ which has columns equal to normalized $\overline{\mathbf{u}}$, $\overline{\mathbf{v}}$, and $\overline{\mathbf{w}}$ and then it is possible to compute the 3D viewpoint vector as $\boldsymbol{\phi} = -\mathbf{R}^T\overline{\mathbf{b}}$. The minus sign is necessary as we need the viewpoint vector going from the vehicle to the camera, not the opposite one.

When the viewpoint vector to the vehicle, the vehicle's class, and its position on the screen are determined, we render the obtained 3D model for the class of the vehicle, under the same viewpoint and on the same position in the image remaining the only unknown to be the scale of the vehicle (distance between the vehicle and the camera) which should be rendered. Examples of the two used 3D models are shown in Figure 6. Therefore, we render images of the vehicle in multiple different scales and match the bounding boxes of the rendered vehicles with the bounding box detected in the video by using the Intersection-over-Union (IoU) metric. Examples of such matches can be found in Figure 7. We also added points to the base of the 3D models representing front $\mathbf{f}$ and rear $\mathbf{r}$ of the vehicle (red lines in Figure 7). Finally, for all vehicle instances i and scales j, these points are projected on the road plane obtaining $\mathbf{F}_{ij}$ and $\mathbf{R}_{ij}$ and they are used to compute scale λ_{ij} (Eq. (13)) for the vehicle instance, where l_{t_i} is the real world length of the type t_i. For all these combinations, the IoU matching metric

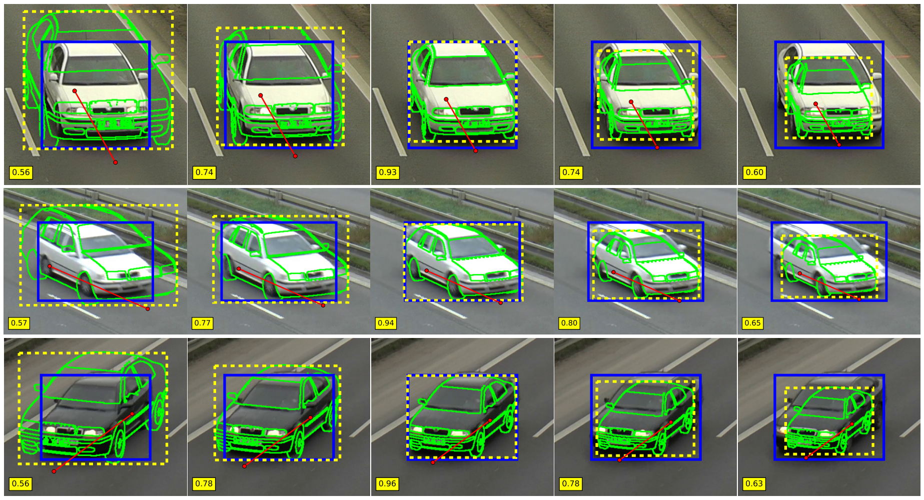

Figure 7: Development of IoU (yellow boxes) metric for different scales (**left to right**), vehicle types and viewpoints (**top to bottom**). The left two images show larger rendered vehicle, the middle one show the best match, and the right two images show smaller rendered vehicle. The rendered vehicle is shown only in a form of edges with yellow rectangle as bounding box of the rendered model and blue rectangle denotes the detected vehicle bounding box.

m_{ij} is computed.

$$\lambda_{ij} = \frac{l_{t_i}}{\|\mathbf{F}_{ij} - \mathbf{R}_{ij}\|} \tag{13}$$

To obtain the final scale λ^* for the camera, all the scales λ_{ij} are taken into account together with metrics m_{ij} with m_{ij} larger then a predefined threshold (we used 0.85 in our experiments) to eliminate poor matches. Finally, we compute λ^* according to Equation (14) with probability $p\left(\lambda \,|\, (\lambda_{ij}, m_{ij})\right)$ computed by kernel density estimation with discretized space.

$$\lambda^* = \arg\max_\lambda \; p\left(\lambda \,|\, (\lambda_{ij}, m_{ij})\right) \tag{14}$$

In order to further improve the scale inference, we use several training videos from BrnoCompSpeed dataset (Sochor et al., 2016b) and train scale correction linear regression $\lambda^*_{reg} = \alpha\lambda^* + \beta$, where we use manually obtained scales as the ground truth values for the training. Even though this step is not necessary, it improves the scale acquisition furthermore. This improvement is caused by imprecise geometry of the obtained 3D models.

We have also experimented with an alignment metric based on matching of edges of detected and rendered vehicles by distance transform based metric, but the speed measurement did not improved further. The biggest problem with this method is that most of the edges on the vehicles are blurry and therefore not detected at all. However, the vehicle detector (Ren et al., 2015) is able to detect the vehicles properly and in most cases accurately. Also, the proposed algorithm using just the bounding boxes is much

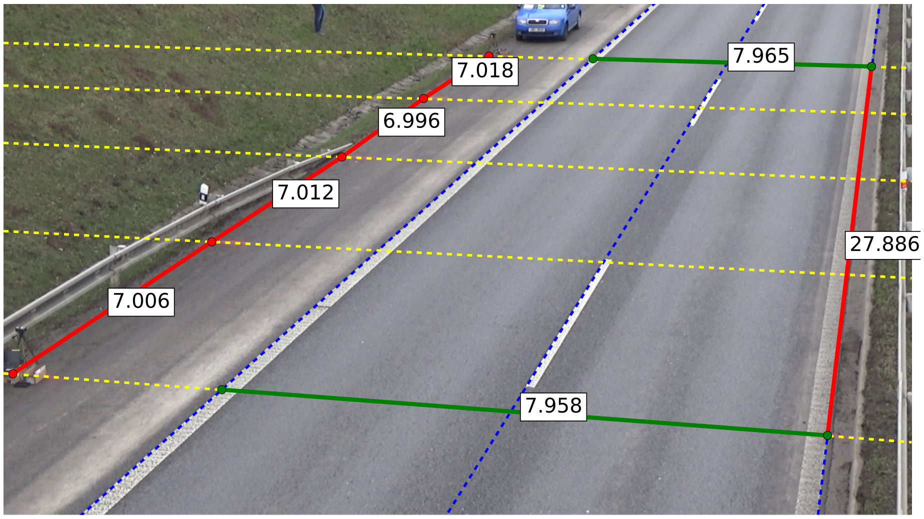

Figure 8: An example of manually measured distances between markers on the road plane. Other examples can be found in the original BrnoCompSpeed publication (Sochor et al., 2016b). Blue lines denote the lane dividing lines, lines perpendicular to the vehicles direction are shown in yellow. Finally, measured distances between two points towards the first (second) vanishing point are shown by red (green) color.

more efficient in terms of storage (it is possible to store just the bounding boxes, not the images) and computation.

4.4. Speed Measurement of Tracked Cars

The speed measurement itself is done by following the methodology proposed by Sochor et al. (2016b). Given a tracked car with reference points $\mathbf{p}_i$ and timestamps t_i for each of the reference point, where $i = 1 \dots N$, the speed v is calculated from Equation (15) by projecting the reference points $\mathbf{p}_i$ to the ground plane $\mathbf{P}_i$ (see Equation (8)).

$$v = \operatorname*{median}_{i=1\dots N-\tau} \left(\frac{\lambda^*_{reg} \|\mathbf{P}_{i+\tau} - \mathbf{P}_i\|}{t_{i+\tau} - t_i} \right) \tag{15}$$

The speed is computed as the median value of speeds between consequent time positions. However, for stability of the measurement, it is better not to use the next frame, but the time position several video frames apart. This is controlled by constant τ and for all our experiments, we use $\tau = 5$ (the time difference is usually $0.2\,\mathrm{s}$).

5. Experiments and Results

To evaluate our proposed methods for camera calibration and scene scale inference, we use a very recent dataset BrnoCompSpeed (Sochor et al., 2016b) which contains over 20 k of vehicles with precise ground truth speed from multiple locations. The dataset also contains markers on the road with known dimensions between them. For an example of such road markers, see Figure 8. The ground truth distances can be used for either calibration or evaluation of distance measurement on the road plane. It is also possible to evaluate the accuracy of vanishing points estimation with the markings (Sochor et al., 2016b). In the following text we will refer to various methods for camera calibration which are defined as:

- **ITS15** – Automatic camera calibration method as described by Dubská et al. (2015). Brief outline of the method is in Sections 2 and 4.1.
- **Edgelets** – Camera calibration method proposed in this paper, Section 4.1.
- **ManualCalib** – We use known distances (Figure 8) on the road for manual calibration of the camera. In agreement with the previous papers (Cathey and Dailey, 2005; Grammatikopoulos et al., 2005; He and Yung, 2007a) we use intersection lanes dividing lines (blue dashed lines in Figure 8) for estimation of the first vanishing point $\mathbf{u}$. As there are usually more than just two lane dividing lines, we use least squares minimization to obtain the intersection for multiple lines. Formally, given lines $\mathbf{l}_i$ with normalized normal vectors, we compute vanishing point $\mathbf{u}$ by solving $\mathbf{A}\mathbf{u} = -\mathbf{b}$ in a least squares manner, where rows of $\mathbf{A}$ contain transposed normal vectors of the lines and rows of $\mathbf{b}$ contain constant terms of the lines.

  The second vanishing point $\mathbf{v}$ can be obtained in the same manner (as the intersection of yellow dashed lines in Figure 8, since they are perpendicular to the vehicle flow on the road). However, we found out that it is more accurate and robust to use the intersection only as a first guess and then use measured distances on the road to optimize the vanishing point position using Equation (16).

  $$\mathbf{v} = \arg\min_{\mathbf{v}} \left(\sum_{(\mathbf{p}_1, \mathbf{p}_2, d) \in \mathcal{D}_2} |\lambda \|\mathbf{P}_1 - \mathbf{P}_2\| - d| \right), \tag{16}$$

  where set $\mathcal{D}_2$ contains image endpoints and distances measured on the road towards the second vanishing point (green line segments in Figure 8) and scale λ is computed for the given vanishing points $\mathbf{u}, \mathbf{v}$ by Equation (17). It should be noted that the computation of 3D coordinates $\mathbf{P}_i$ of image point $\mathbf{p}_i$ depends on the vanishing points (see Equation (8) for details). The optimization itself is done by grid search.

  The usage of standard manual methods based on calibration patterns (e.g checkerboards) proposed by Zhang (2000) is impractical as it would require a large checkerboard (more than $10\,\mathrm{m}^2$) which would be placed on the road.

We also define method names for different approaches for scale inference:

- **BMVC14** – Scale inference method proposed by Dubská et al. (2014). Brief outline of the method is in Section 2.
- **BBScale + reg** – Our method for scale calibration using bounding box matching (Section 4.3) with scale correction regression.

- **ManualScale** – Scale computed from manually measured distances between markers towards the first vanishing point on the road. The scale is computed as the mean value of Equation (17) from a set of endpoints and distances $(\mathbf{p}_{i,1}, \mathbf{p}_{i,2}, d_i)$ towards the first vanishing point (red line segments in Figure 8).

$$\lambda = \mathbb{E}\left[\frac{d_i}{\|\mathbf{P}_{i,1} - \mathbf{P}_{i,2}\|}\right] \tag{17}$$

- **SpeedScale** – Scale is computed from the ground truth speed measurements and it minimizes the speed measurement error for given camera calibration. It can be understood as the lower error bound for the given camera calibration method. The scale is computed as the mean value of Equation (18) where set $\mathcal{M}$ contains pairs of ground truth speed $\hat{v}_i$ and measured speed v_i. It is assumed that scale $\lambda = 1$ was used for computation of speeds v_i.

$$\lambda = \mathbb{E}\left[\frac{\hat{v}_i}{v_i}\right] \tag{18}$$

If not stated otherwise, the evaluation was done on BrnoCompSpeed – Split C (the split contains more than 10 k of vehicle tracks for evaluation) as our method requires parameter tuning for the scale correction regression. For each metric, we report mean, median, and 99 percentile error for both absolute units ($err = |\hat{r} - r|$) and relative units ($err = |\hat{r} - r|/\hat{r} \cdot 100\%$), where $\hat{r}$ denotes the ground truth measurement, and r represents the measured value.

5.1. Evaluation of VP Estimation – Camera Calibration Error

To evaluate the camera calibration itself (the obtained vanishing points), we follow the evaluation metric proposed with BrnoCompSpeed dataset (Sochor et al., 2016b). The evaluation measures the difference between ratios of distances between markings towards the first vanishing point (red lines in Figure 8) and the distances between markers towards the second vanishing point (green lines in Figure 8). As the ratio does not depend on scale, this metric considers only the camera calibration in the form of two detected vanishing points.

As we do not require any parameter tuning for the camera calibration method, we report the results on all videos in BrnoCompSpeed dataset (including extra session0). The results (reported in Table 1) show that our automatic calibration method Edgelets outperforms calibration method ITS15 almost twice in mean error. However, even though the automatic calibration results were improved by our proposed method, the results of manual calibration are still better. Also, it should be noted that the same distances which were used to obtain the manual calibration were evaluated by the calibration error metric based on distance ratios; this gives the manual calibration an advantage in the comparison.

The significant improvement of our method is caused by more precise acquisition of the $\mathbf{v}$; position of $\mathbf{u}$ stays the same for our method as for ITS15 calibration method.

Table 1: Errors of distance measurement ratios (see Section 5.1 for details). The first row for each calibration method contains absolute errors; the relative errors in percents are in the second row.

system	mean	median	99 %
Edgelets (ours)	0.09	0.04	0.49
	6.45	3.38	39.08
ITS15	0.18	0.05	1.36
	11.74	5.25	61.03
ManualCalib	0.02	0.01	0.15
	1.80	1.26	10.98

Table 2: Distance measurement errors on the road plane for different calibrations. Only distances towards the first vanishing point (red in Figure 8) were used for this evaluation. The first row for each calibration method contains absolute errors in meters; the relative errors in percents are in the second row.

system	mean	median	99 %
Edgelets + BBScale + reg (ours)	0.26	0.17	1.08
	2.33	2.06	5.49
ITS15 + BMVC14	1.23	0.81	5.40
	9.62	10.65	21.07
Edgelets + ManualScale (ours)	0.10	0.06	0.57
	0.98	0.62	4.46
ITS15 + ManualScale	0.25	0.14	1.54
	2.11	1.66	8.07
ManualCalib + ManualScale	0.10	0.08	0.32
	1.08	0.65	3.59

5.2. Evaluation of Distance Measurement on Road

The next step is to evaluate the camera calibration together with the obtained scale. To do that we use manual annotation of distances on the road plane which are going towards the first or the second vanishing point respectively (red and green in Figure 8).

First, we evaluated the distance measurement only towards the first vanishing point as it is the direction in which the vehicles are going and it is more important for speed measurement. The results are shown in Table 2 for different combinations of calibrations and scale estimations. The table shows several things. First, our fully automatic method for camera calibration (Edgelets) and scale inference (BBScale + reg) significantly outperforms the previous automatic method ITS15 + BMVC14. Second, when we use our automatically computed calibration and scale obtained with manual annotations, we achieve almost the same results as ManualCalib + ManualScale, which required much more manual work and effort than our automatic system.

When we evaluated the same metric with all the distances, the results are similar (see Table 3). Again, our method significantly outperforms the previous automatic method. Considering the calibrations with manually obtained scale, our system has a

Table 3: Distance measurement errors on the road plane for different calibrations. Each segment of table represent level of supervision in the calibration. The first row for each calibration method contains absolute errors in meters and the relative errors in percents are in the second row.

system	**mean**	**median**	**99 %**
Edgelets + BBScale + reg (ours)	0.34	0.18	2.29
	3.47	2.28	30.49
ITS15 + BMVC14	1.17	0.72	5.82
	9.79	9.00	55.89
Edgelets + ManualScale (ours)	0.24	0.10	2.60
	2.66	1.00	34.75
ITS15 + ManualScale	0.57	0.20	5.43
	5.84	2.07	52.19
ManualCalib + ManualScale	0.07	0.04	0.30
	0.84	0.50	3.47

little bit higher error then the manual calibration. However, this is caused by the fact that the manual calibration is optimized directly to this metric by Equation (16).

To summarize the distance measurement results, our method significantly outperforms previous automatic state of the art for speed measurement – the mean error for distance measurement in the direction of vehicles' flow (which is important for speed measurement) was reduced by 79 % (1.23 to 0.26).

5.3. *Evaluation of Speed Measurement*

The most important part of the evaluation is the speed measurement itself. We used the same vehicle detection and tracking system (see Section 5) so that the results for different calibrations and scales are directly comparable.

We show both quantitative results in the form of Table 4 and plots with cumulative error histograms in Figure 9. The table and the figures are divided into several parts where we compare similar levels of supervision.

The first level of supervision is fully automatic; in the second level, known ground truth dimensions on the road plane are used. In the third and final level of supervision, we use known ground truth speeds to form the lower error bound for different calibration methods.

Regarding the first level of supervision, our system Edgelets + BBScale + reg significantly outperforms (1.10 km/h mean error) the previous automatic method ITS15 + BMVC14 (7.98 km/h mean error) and we reduce the mean speed measurement error by 86 %. Another important fact is that our fully automatic method for camera calibration and scale inference also outperforms manual calibration and scale inference (1.35 km/h mean error) while the error is reduced by 19 %. This improvement is important as in the previous approaches, the automatism always compromised the accuracy, forcing the system developer to trade off between them. Our work shows that our proposed fully automatic method based on computer vision methods is superior to the manual calibration.

Table 4: Evaluation of speed measurement errors; all the systems are different only in the calibration and scale inference, with the same tracking of vehicles. Each segment represents one level of supervision in the calibration (automatic, known ground truth distances on road, known ground truth speeds). The first row for each calibration method contains absolute errors in km/h; the relative errors in percents are in the second row.

system	**mean**	**median**	**99 %**
Edgelets + BBScale + reg (ours)	1.10	0.97	3.05
	1.39	1.22	4.13
ITS15 + BMVC14	7.98	8.18	18.58
	10.15	11.45	19.22
Edgelets + ManualScale (ours)	1.04	0.83	3.48
	1.31	1.04	4.61
ITS15 + ManualScale	1.44	1.17	5.43
	1.76	1.50	6.16
ManualCalib + ManualScale	1.35	0.95	4.84
	1.64	1.18	5.40
Edgelets + SpeedScale (ours)	0.52	0.35	2.57
	0.66	0.44	3.71
ITS15 + SpeedScale	0.80	0.57	3.70
	0.99	0.72	4.68
ManualCalib + SpeedScale	0.56	0.38	2.73
	0.71	0.48	3.63

When it comes to the second and the third level of supervision, the results follow the same trend with our calibration outperforming all of them (manual and automatic). The fact that manual calibration is better on the calibration metric (Section 5.1) and distance measurement (Section 5.2), while our method outperforms the manual calibration on the speed measurement task, is caused by that the manual calibration use the same data which are then used for the evaluation of the calibration metric and distance measurement. The achieved accuracy is very close to meeting the standards for speed measurements accuracy required for enforcement. The accuracy is definitely comparable to measurements achievable by radars (Sochor et al., 2016b), while being considerably cheaper, more flexible, and passive.

5.4. 3D Model Selection Sensitivity Analysis

We have also evaluated how using different 3D models of vehicles influences the speed measurement results. The results are shown in Table 5 and Figure 9 (bottom right). We have tested several combinations of used vehicles: use of only one of the models (Combi, Sedan) or both of them together (Combi + Sedan), forming the first segment of the table. It shows that using both the models significantly improves the results, as the errors in geometry of the 3D models cancel out. We consider using only a few fine-grained models as beneficial as it is not necessary to obtain more 3D models and training data for fine-grained recognition.

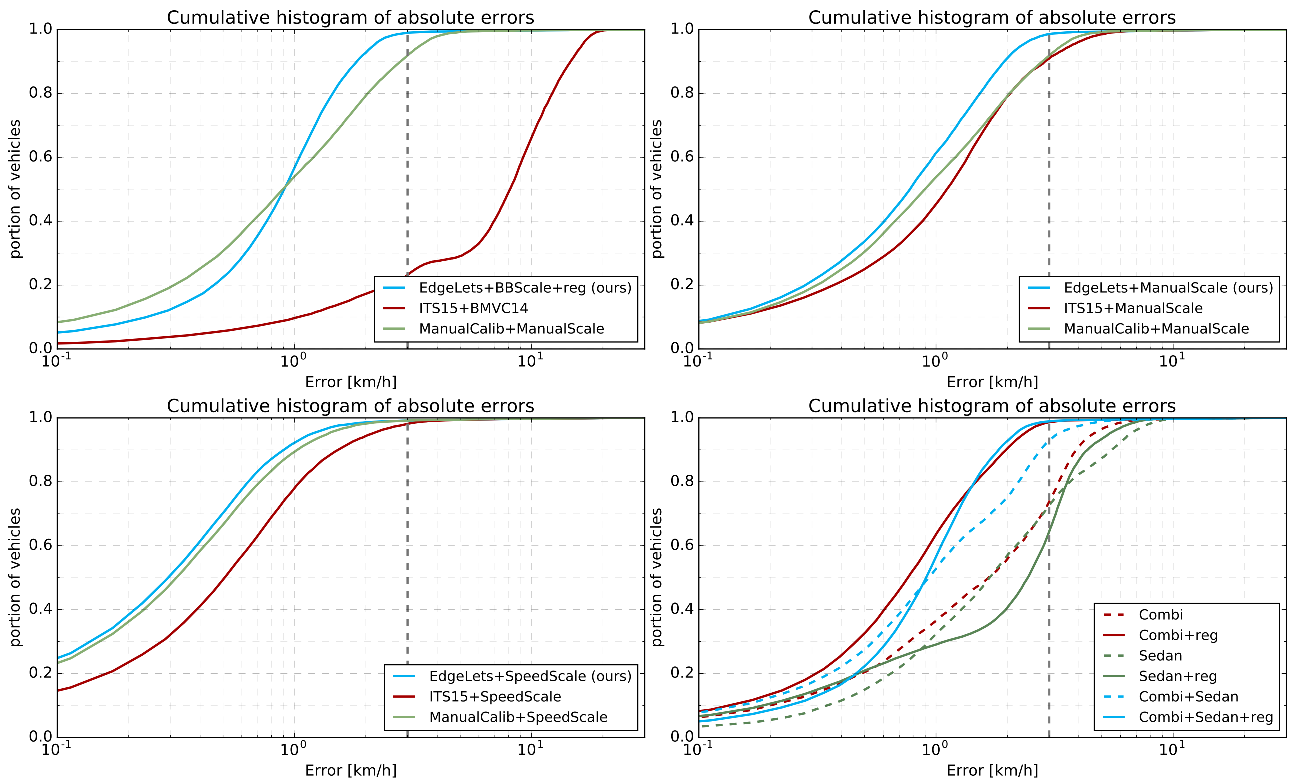

Figure 9: Evaluation of speed measurement – cumulative histograms of errors. The gray dashed vertical lines represent 3 km/h error. **top left:** comparison of automatic methods and a manual method for camera calibration, **top right:** calibrations obtained with known ground truth distances on the road plane, **bottom left:** calibrations with scale obtained by minimizing the speed measurement error, thus forming a lower bound error for speed measurement with given camera calibration and tracking algorithm, **bottom right:** analysis of influence of different aspects of used 3D car models evaluated on speed measurement, see Section 5.4.

The second segment of the table shows the performance of the system with scale correction regression to overcome the inaccuracies of the 3D models. The results show that for model Combi, the error significantly decreases. However, for the Sedan model, the results stay more or less the same. This paradox is caused by the smaller number of training data for Sedan version as for some training videos, no Sedan vehicle was detected. The results also show that if we use both models, the performance drop is not that significant (1.10 to 1.38) and therefore, it is possible to use the scale inference without the scale correction regression.

5.5. *Vehicle Detection and Tracking Evaluation*

As we use a different vehicle detection and tracking method than Dubská et al. (2014), we evaluate also this part of the solution. We compare the methods on all

Table 5: Analysis of influence of different aspects of used 3D car models. It shows that it is best to use both models. The second segment of the table also shows that it is useful to use scale correction regression as described in Section 4.3. The first row for each 3D model combination method contains absolute errors in km/h; the relative errors in percents are in the second row.

system	**mean**	**median**	**99 %**
Sedan	2.39 2.82	1.74 2.14	8.67 7.74
Combi	2.03 2.48	1.72 2.14	6.51 5.94
Combi + Sedan	1.38 1.70	0.99 1.23	5.18 4.94
Sedan + reg	2.43 2.97	2.49 3.17	7.26 6.56
Combi + reg	1.03 1.33	0.82 1.04	3.29 4.49
Combi + Sedan + reg	1.10 1.39	0.97 1.22	3.05 4.13

Table 6: Evaluation of differences between vehicle detection and tracking proposed by Dubská et al. (2014) and our detection and tracking method. FPPM denotes the number of False Positives Per Minute, recall was computed as mean recall across all videos and speed error denotes mean speed measurement error.

method	**FPPM**	**recall**	**speed error [km/h]**
Dubská et al. (2014)	9.77	0.885	1.46
ours	1.91	0.863	1.21

videos of BrnoCompSpeed (including extra session0) with exactly the same calibration (ManualCalib + ManualScale) to isolate the influence of vehicle detection and tracking.

We report the number of False Positives Per Minute and mean recall in vehicles counting. The results can be found in Table 6 and as the table shows, our method significantly reduces the number of false positives with essentially the same recall.

A tracked vehicle is matched to the ground truth if it passes through the correct lane and the time difference of pass through the measurement line (yellow line in Figure 8 which is closest to the camera) compared to the ground truth is less then $0.2\,\mathrm{s}$. This threshold is used by Sochor et al. (2016b) to safely match the vehicles as a higher threshold could lead to mismatches between the detected track and ground truth.

As we use the same calibration, we can also compare directly the speed measurement error which is influenced (with the same calibration) only by the tracking. As the table shows, our tracking method yields slightly reduced speed measurement error for the same scale and camera calibration.

For the tracking and speed measurement, we use the point at the front of the vehicle on the road plane (using 3D bounding box), which is geometrically correct, as the point is on the road plane. We evaluated how the choice of the tracking point influence the measurement error, comparing to naive solution which takes center of the bottom edge of the 2D bounding box for the tracking, and found out that the difference to the correct solution was negligible.

5.6. Camera Calibration on Real Surveillance Cameras

The automatic calibration from vehicle movement can be justly suspected of requiring idealized conditions and to be sensitive to bad lighting, etc. In order to verify the usability of our camera calibration method in real-world conditions, we obtained data from surveillance cameras in production use at 9 different locations. The videos were captured both at day and night conditions. The data are rather of a poor quality ($704 \times 576\,\mathrm{px}$ or $704 \times 288\,\mathrm{px}$) with 6 frames per second and mean length of 40s. As the ground truth calibration is not available for the data, we report only qualitative results in the form of equilateral grid projected on the road plane. Despite the challenging character of the sequences (poor video quality and lighting conditions), we were able to correctly detect the vanishing points, as can be seen in Figure 10 on a few examples, and thus find the camera parameters and its orientation, which is important in many real-world surveillance applications (e.g estimation of vehicle viewpoints or image rectification).

6. Conclusions

We propose a fully automatic method for traffic surveillance camera calibration. It does not have any constraints on camera placement a does not require any manual input whatsoever. The results show that our system decreases the mean speed measurement error by $86\,\%$ (7.98 km/h to 1.10 km/h) compared to previous automatic state-of-the-art method and by $19\,\%$ (1.35 km/h to 1.10 km/h) compared to manual calibration method. This improvement is important as in the previous approaches, the automatism always compromised the accuracy, forcing the system developer to trade off between them. Our work shows that our proposed fully automatic method based on computer vision algorithms is superior to the manual calibration. This result can be important beyond the field of traffic surveillance, since different forms of manual camera calibration are often considered the “ground truth”, but our work shows that automatic calibration from statistics of repeated inaccurate measurements can be more precise, despite requiring no user input.

In the experiments, we also showed that our method is able to calibrate real world traffic surveillance cameras and our proposed method for vehicle detection and tracking significantly reduces the number of false positives compared to the previous method. In the future work, we would like to simplify the system and remove the necessity to render the vehicles by approximation of the bounding box size by a function parametrized by viewpoint and image location.

Acknowledgments

This work was supported by The Ministry of Education, Youth and Sports of the Czech Republic from the National Programme of Sustainability (NPU II); project

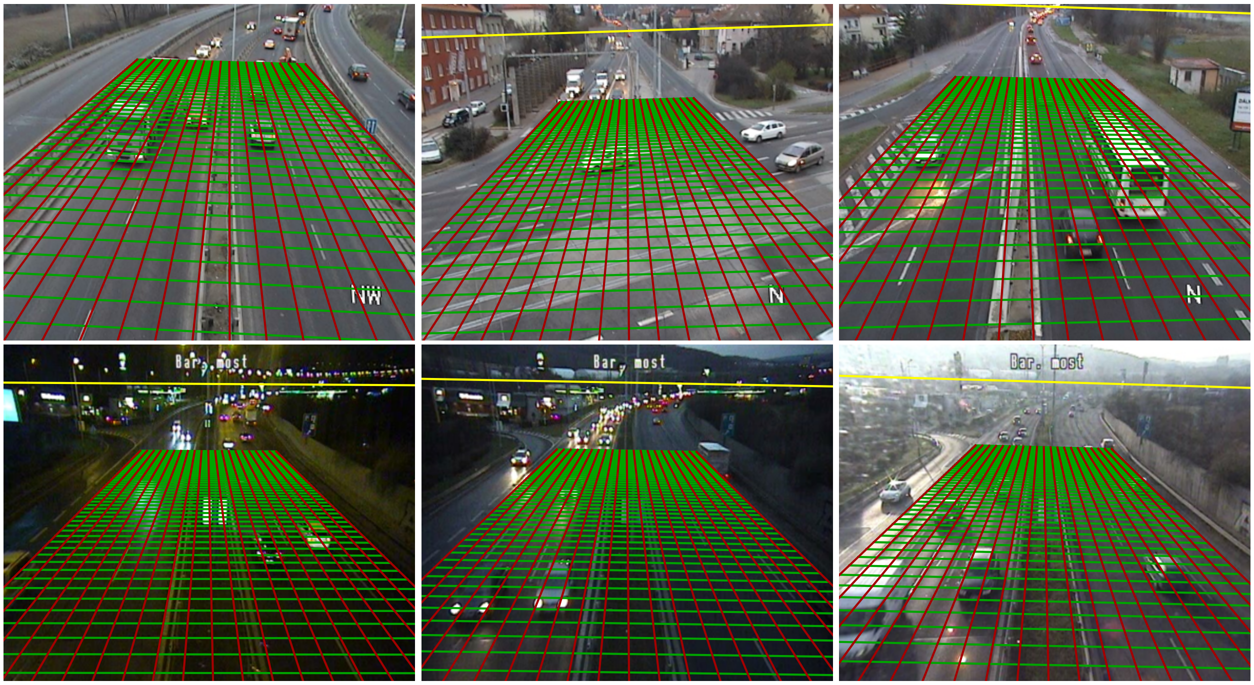

Figure 10: Example of camera calibration (two vanishing points) for real world surveillance cameras. The first row shows different locations, while the second one show the same locations at night, dawn, and daylight. The yellow line denotes the detected horizon (if present inside the frames) and red-green grid is formed by lines going to the first vanishing point (red) and to the second one (green). In an ideal case the grid is perpendicular in the real world and the lines are parallel to the features which define the vanishing points on the ground (e.g. line marking). Also, it should be noted that the method is able to work even on an intersection (top center).

IT4Innovations excellence in science – LQ1602.

We would also like to thank to company CAMEA, s r.o. for providing us data from industrial surveillance cameras.

References

References

Cathey F, Dailey D. A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In: Intelligent Vehicles Symposium. 2005. p. 777–82. doi:10.1109/IVS.2005.1505199.

Chaperon T, Droulez J, Thibault G. Reliable camera pose and calibration from a small set of point and line correspondences: A probabilistic approach. Computer Vision and Image Understanding 2011;115(5):576 –85. URL: http://www.sciencedirect.com/science/article/pii/S1077314210002584.

doi:http://dx.doi.org/10.1016/j.cviu.2010.11.018; special issue on 3D Imaging and Modelling.

Cootes TF, Taylor CJ, Cooper DH, Graham J. Active shape models-their training and application. Computer vision and image understanding 1995;61(1):38–59.

Dailey D, Cathey F, Pumrin S. An algorithm to estimate mean traffic speed using uncalibrated cameras. IEEE Transactions on Intelligent Transportation Systems 2000;1(2):98–107. doi:10.1109/6979.880967.

Dalal N, Triggs B. Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. IEEE; volume 1; 2005. p. 886–93.

Do VH, Nghiem LH, Thi NP, Ngoc NP. A simple camera calibration method for vehicle velocity estimation. In: Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), 2015 12th International Conference on. 2015. p. 1–5. doi:10.1109/ECTICon.2015.7207027.

Dollár P, Appel R, Belongie S, Perona P. Fast feature pyramids for object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence 2014;36(8):1532–45. doi:10.1109/TPAMI.2014.2300479.

Dubská M, Sochor J, Herout A. Automatic camera calibration for traffic understanding. In: BMVC. 2014. .

Dubská M, Herout A. Real projective plane mapping for detection of orthogonal vanishing points. In: Proceedings of the British Machine Vision Conference. BMVA Press; 2013. .

Dubská M, Herout A, Juranek R, Sochor J. Fully automatic roadside camera calibration for traffic surveillance. Intelligent Transportation Systems, IEEE Transactions on 2015;16(3):1162–71. doi:10.1109/TITS.2014.2352854.

Fang J, Zhou Y, Yu Y, Du S. Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture. IEEE Transactions on Intelligent Transportation Systems 2016;PP(99):1–11. doi:10.1109/TITS.2016.2620495.

Felzenszwalb P, Girshick R, McAllester D, Ramanan D. Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 2010;32(9):1627–45. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5255236. doi:10.1109/TPAMI.2009.167.

Fung GSK, Yung NHC, Pang GKH. Camera calibration from road lane markings. Optical Engineering 2003;42(10):2967–77.

Gao Y, Beijbom O, Zhang N, Darrell T. Compact bilinear pooling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. .

Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Computer Vision and Pattern Recognition. 2014. .

Grammatikopoulos L, Karras G, Petsa E. Automatic estimation of vehicle speed from uncalibrated video sequences. In: Proceedings of International Symposium on Modern Technologies, Educationand Profeesional Practice in Geodesy and Related Fields. 2005. p. 332–8.

He X, Yung N. New method for overcoming ill-conditioning in vanishing-point-based camera calibration. Optical Engineering 2007a;46(3).

He XC, Yung NHC. A novel algorithm for estimating vehicle speed from two consecutive images. In: IEEE Workshop on Applications of Computer Vision, WACV. 2007b. doi:10.1109/WACV.2007.7.

Hsiao E, Sinha S, Ramnath K, Baker S, Zitnick L, Szeliski R. Car make and model recognition using 3D curve alignment. In: IEEE WACV. 2014. doi:10.1109/WACV.2014.6836126.

Juránek R, Herout A, Dubská M, Zemčík P. Real-time pose estimation piggybacked on object detection. In: The IEEE International Conference on Computer Vision (ICCV). 2015. .

Kalman RE. A new approach to linear filtering and prediction problems. Transactions of the ASME – Journal of Basic Engineering 1960;(82 (Series D)):35–45.

Krause J, Jin H, Yang J, Fei-Fei L. Fine-grained recognition without part annotations. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2015. .

Krause J, Stark M, Deng J, Fei-Fei L. 3D object representations for fine-grained categorization. In: ICCV Workshop 3dRR-13. 2013. .

Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Pereira F, Burges C, Bottou L, Weinberger K, editors. Advances in Neural Information Processing Systems 25. Curran Associates, Inc.; 2012. p. 1097–105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Lan J, Li J, Hu G, Ran B, Wang L. Vehicle speed measurement based on gray constraint optical flow algorithm. Optik - International Journal for Light and Electron Optics 2014;125(1):289 –95. URL: http://www.sciencedirect.com/science/article/pii/S0030402613009005. doi:http://dx.doi.org/10.1016/j.ijleo.2013.06.036.

Li C, Gatenby C, Wang L, Gore JC. A robust parametric method for bias field estimation and segmentation of mr images. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE; 2009. p. 218–23.

Lin TY, RoyChowdhury A, Maji S. Bilinear cnn models for fine-grained visual recognition. In: International Conference on Computer Vision (ICCV). 2015. .

Lin YL, Morariu VI, Hsu W, Davis LS. Jointly optimizing 3D model fitting and fine-grained classification. In: ECCV. 2014. .

Liu J, Kanazawa A, Jacobs D, Belhumeur P. Dog breed classification using part localization. In: Fitzgibbon A, Lazebnik S, Perona P, Sato Y, Schmid C, editors. ECCV 2012. Springer Berlin Heidelberg; volume 7572 of *Lecture Notes in Computer Science*; 2012. p. 172–85. URL: http://dx.doi.org/10.1007/978-3-642-33718-5_13. doi:10.1007/978-3-642-33718-5_13.

Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. Ssd: Single shot multibox detector. 2016. URL: http://arxiv.org/abs/1512.02325; to appear.

Lowe DG. Object recognition from local scale-invariant features. In: Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Ieee; volume 2; 1999. p. 1150–7.

Luvizon D, Nassu B, Minetto R. Vehicle speed estimation by license plate detection and tracking. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. 2014. p. 6563–7. doi:10.1109/ICASSP.2014.6854869.

Luvizon DC, Nassu BT, Minetto R. A video-based system for vehicle speed measurement in urban roadways. IEEE Transactions on Intelligent Transportation Systems 2016;PP(99):1–12. doi:10.1109/TITS.2016.2606369.

Maduro C, Batista K, Peixoto P, Batista J. Estimation of vehicle velocity and traffic intensity using rectified images. In: Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on. 2008. p. 777–80. doi:10.1109/ICIP.2008.4711870.

Nurhadiyatna A, Hardjono B, Wibisono A, Sina I, Jatmiko W, Ma'sum M, Mursanto P. Improved vehicle speed estimation using gaussian mixture model and hole filling algorithm. In: Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on. 2013. p. 451–6. doi:10.1109/ICACSIS.2013.6761617.

Prokaj J, Medioni G. 3-D model based vehicle recognition. In: IEEE WACV. 2009. doi:10.1109/WACV.2009.5403032.

Ren S, He K, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS). 2015. .

Schoepflin T, Dailey D. Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. Intelligent Transportation Systems, IEEE Transactions on 2003;4(2):90–8. doi:10.1109/TITS.2003.821213.

Shi J, Tomasi C. Good features to track. In: 1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 1994. p. 593–600. doi:10.1109/CVPR.1994.323794.

Simon M, Rodner E. Neural activation constellations: Unsupervised part model discovery with convolutional networks. In: International Conference on Computer Vision (ICCV). 2015. .

Sina I, Wibisono A, Nurhadiyatna A, Hardjono B, Jatmiko W, Mursanto P. Vehicle counting and speed measurement using headlight detection. In: Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on. 2013. p. 149–54. doi:10.1109/ICACSIS.2013.6761567.

Sochor J, Herout A, Havel J. BoxCars: 3D boxes as CNN input for improved fine-grained vehicle recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016a. .

Sochor J, Juranek R, Spanhel J, Marsik L, Siroky A, Herout A, Zemcik P. BrnoCompSpeed: Review of traffic camera calibration and a comprehensive dataset for monocular speed measurement. Intelligent Transportation Systems (under review), IEEE Transactions on 2016b;.

Tomasi C, Kanade T. Detection and Tracking of Point Features. Technical Report; International Journal of Computer Vision; 1991.

You X, Zheng Y. An accurate and practical calibration method for roadside camera using two vanishing points. Neurocomputing 2016;:–URL: http://www.sciencedirect.com/science/article/pii/S092523121630100X. doi:http://dx.doi.org/10.1016/j.neucom.2015.09.132.

Yu X, Jiang N, Cheong LF, Leong HW, Yan X. Automatic camera calibration of broadcast tennis video with applications to 3d virtual content insertion and ball detection and tracking. Computer Vision and Image Understanding 2009;113(5):643 –52. URL: http://www.sciencedirect.com/science/article/pii/S1077314208000416. doi:http://dx.doi.org/10.1016/j.cviu.2008.01.006; computer Vision Based Analysis in Sport Environments.

Zhang Z. A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000;22(11):1330–4. doi:10.1109/34.888718.

Zhang Z, Tan T, Huang K, Wang Y. Practical camera calibration from moving objects for traffic scene surveillance. IEEE Transactions on Circuits and Systems for Video Technology 2013;23(3):518–33. doi:10.1109/TCSVT.2012.2210670.

Zheng Y, Peng S. A practical roadside camera calibration method based on least squares optimization. IEEE Transactions on Intelligent Transportation Systems 2014;15:831–43.