Noname manuscript No.
(will be inserted by the editor)

Automatic Camera Calibration by Landmarks on Rigid Objects

Vojtěch Bartl · Jakub Špaňhel · Petr Dobeš · Roman Juránek · Adam Herout

Received: date / Accepted: date

Abstract This article presents a new method for automatic calibration of surveillance cameras. We are dealing with traffic surveillance and therefore the camera is calibrated by observing vehicles; however, other rigid objects can be used instead. The proposed method is using *keypoints* or *landmarks* automatically detected on the observed objects by a convolutional neural network. By using fine-grained recognition of the vehicles (calibration objects), and by knowing the 3D positions of the landmarks for the (very limited) set of known objects, the extracted keypoints are used for calibration of the camera, resulting in internal (focal length) and external (rotation, translation) parameters and scene scale of the surveillance camera. We collected a dataset in two parking lots and equipped it with a calibration ground truth by measuring multiple distances in the ground plane. This dataset seems to be more accurate than the existing comparable data (GT calibration error reduced from 4.62 % to 0.99 %). Also, the experiments show that our method overcomes the best existing alternative in terms of accuracy (error reduced from 6.56 % to 4.03 %) and our solution is also more flexible in terms of viewpoint change and other.

1 Introduction

Camera calibration is an important step in the majority of machine vision applications. In various surveillance scenarios, calibration including scale (to tell the position in world units, like meters, not in image units) is of high importance. While research works treat camera calibration as a solved problem by showing a checkerboard to the experimental camera, practical applications often require a calibration procedure that is automatized and suitable for large scene scales. There is a large amount of surveillance cameras around the world and the possibility to calibrate them (in order to be usable in machine vision applications) without the necessity of physical presence is essential. Zhang [65] popularized calibration by inserting a suitable pattern of known properties; in his case, a planar printed checkerboard is used, but arbitrary planar and non-planar [39, 66] objects have been used since. However, in surveillance of real-world scenes, especially with large numbers of processed cameras, it is extremely inconvenient to calibrate the cameras by showing them markers and by making additional distance measurements in the scene (e.g. in the midst of the traffic lanes of a highway).

The goal of our work is to develop fully automatic calibration algorithms for surveillance, providing the internal camera parameters, camera's rotation and translation with regard to the ground plane, and also the scene's scale so that measurements can be done in the world units (meters). We focus on traffic surveillance, thus in this article, we are using vehicles as objects of know properties ("markers"), but the algorithms presented here work with any other suitable rigid objects.

The camera projects every point $\mathbf{x}$ in the world 3D homogeneous coordinates $\mathbf{x} = (x, y, z, 1)^\top$ to its 2D screen image $\mathbf{x}' = (u, v, 1)^\top$:

$$\lambda\, \mathbf{x}' = \mathbf{K}\,[\mathbf{R}|\mathbf{t}]\,\mathbf{x}. \tag{1}$$

By calibration we mean obtaining internal camera parameters $\mathbf{K}$, the camera rotation matrix $\mathbf{R}$ and its translation vector $\mathbf{t}$ that best model such a projection. The

✉ V. Bartl
Brno University of Technology, Faculty of Information Technology, Centre of Excellence IT4Innovations
Tel.: +420-54114-1285
E-mail: ibartl@fit.vutbr.cz

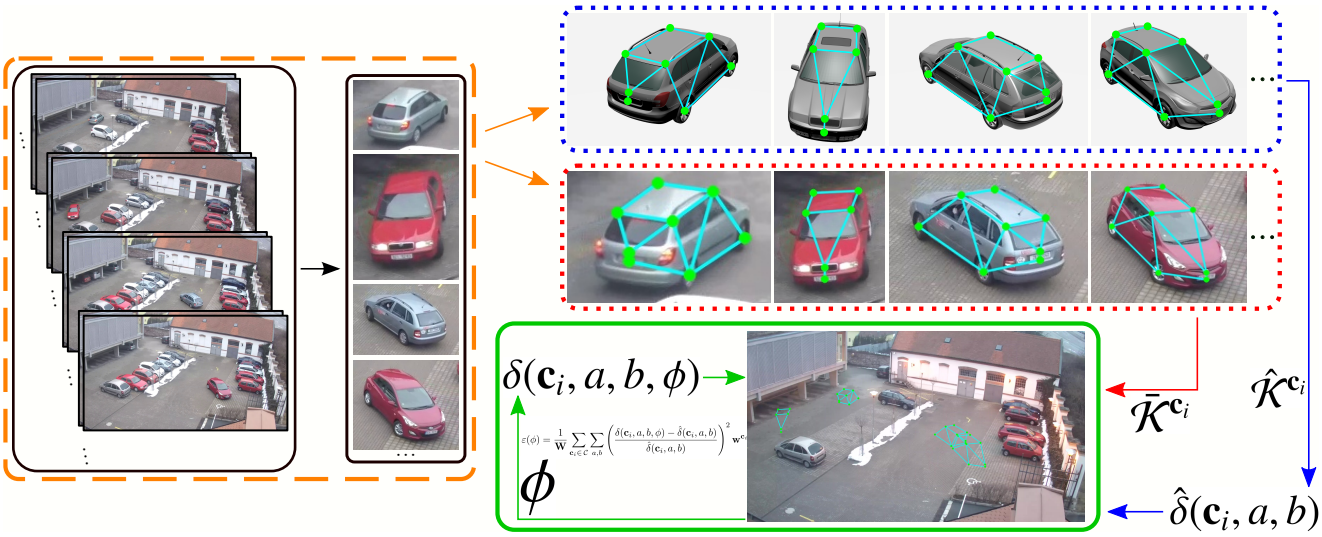

Fig. 1 Overview of the proposed approach. Vehicles observed in the input video (dashed – orange) are classified (to obtain the exact make & model) and processed by a landmark detector (middle dotted – red). For visible landmarks $\bar{\mathcal{K}}^{\mathbf{c}_i}$, their 3D positions $\hat{\mathcal{K}}^{\mathbf{c}_i}$ are obtained from a CAD model (top dotted – blue). Pairwise distances from the known 3D positions $\hat{\delta}(\mathbf{c}_i, a, b)$ are compared with the observed 3D distances $\delta(\mathbf{c}_i, a, b, \phi)$ and the camera model ϕ is optimized (solid – green) by a global optimization method to obtain the best solution.

principal point can be assumed in the center of the image (surveillance cameras generally meet this assumption [48, 56]), the pixels are square and the skew is not present, therefore $\mathbf{K}$ only contains one unknown parameter – the focal length. The rotation and translation $[\mathbf{R}|\mathbf{t}]$ have six degrees of freedom. Our approach assumes a planar surface (the road) on which the vehicles are moving. In that case, from the translation vector $\mathbf{t}$, we are only interested in the height of the camera above the ground (since there is no given central point on the ground plane to look for), so in the end, only 5 parameters are sought after.

A similar task is being solved by *PnP* (Perspective-n-Point) methods [1, 21, 27, 29, 68]. However, these algorithms require the knowledge of the focal length f, which is unknown in our task and it must be optimized together with the rest of the calibration parameters. Even *PnPf* (*PnP* with focal length estimation, [44, 67, 69]) methods exist, which are able to find the focal length, but always from one observation of a rigid structure, contrary to robust calibration methods [65] that compute one common $\mathbf{K}$ and individual per-observation $[\mathbf{R}|\mathbf{t}]$ pairs.

In this paper, we propose a practical calibration method which does not require any calibration pattern and relaxes the requirement of straight motion of vehicles (see Figure 1 for an overview of the method). It is thus useful in more versatile scenarios than the previous methods, such as parking lots and road intersections, where vehicles can move in arbitrary and all directions. However, there are still several assumptions that must be satisfied – the cars must move on a common ground plane and there must be a certain number of observations of vehicles known to the system by their particular type (make & model). Our method harnesses car detection, fine-grained classification, and detection of landmarks on the detected cars (Section 2 gives an overview of state-of-the-art approaches providing these).

The camera is calibrated from locations of detected landmarks in the image and the knowledge of their 3D coordinates in the object local coordinate system of the recognized cars. We detect cars in the incoming images (either video or individual images) by Faster R-CNN detector [47]. We classify the detected cars by their make & model and keep those detections belonging to the 9 most frequent models in the data. We use our previously published recognizer [52, 54] achieving state-of-the-art results in fine-grained vehicle recognition; it is based on CNN with modified input with *unwrapped* bounding boxes of the cars and other modifications. Extraction of landmarks in the image is based on a fully convolution neural network proposed by Newell et al. [41]. For each landmark, their *Stacked Hourglass Network* predicts a probability map, whose maximum determines the location. This design of the convolutional neural network was used by Wang et al. [61] for localizing landmarks on vehicles (OIF, *Orientation Invariant Features*). Wang et al. collected and labeled a dataset and trained a model localizing the said landmarks. The accuracy differs for different landmarks and the resolution of both the input data (256×256 px) and

of the regressed map (64×64 px) suggest the limitations of the localization accuracy.

The calibration itself then takes the observed recognized cars with the landmarks detected as its input and finds the best combination of parameters by global optimization. The optimal set of calibration parameters must best match the observed landmarks on each observed vehicle to their known 3D locations (unique per vehicle type).

The main contributions of this article are the following:

- Method for calibrating the surveillance camera ($\mathbf{K}$ $[\mathbf{R}|\mathbf{t}]$) automatically only by observing vehicles — "**Calibration from Landmarks**".
- State-of-the-Art accuracy in the task of traffic camera calibration (considerably outperforming *AutoCalib* [3] in accuracy and versatility).
- Evaluation of the limits of the OIF landmark extractor [61] for the purpose of camera calibration.
- A publicly available dataset from two parking lots with vehicle observations, recognized vehicle models, detected landmarks, and ground-truth ground plane calibration.

2 Background and Related Work

Existing methods for camera calibration applicable in traffic surveillance are mostly based on manual measurements [37, 38, 42], markers [6, 10, 17, 20, 62], vehicle movement [9, 12, 48, 53], or other principles like optical flow, recognition of cars or license plates [3, 13, 28, 36].

With the exceptions of methods [3, 12, 53], traffic calibration solutions require known dimensions in the scene (e.g. width of lanes, length of dashed markings, height of camera, etc.) and thus they cannot be used in a fully automatic manner. The calibration is often constrained to a limited range of viewpoints and it supports only straight motion of vehicles. Maduro et al. [38] assume a known angle of the camera and a known width of the traffic lanes. Marker-based methods either use special markers or horizontal road markings. Cathey and Dailey [6] detect the vanishing point of the lane marking with a known lane width. Grammatikopoulos et al. [17] assume a camera with zero roll and they detect the vanishing point of the road markings. He and Yung [20] calibrate the camera from a pattern formed by dashed line markings on the road. Do et al. [10] use an equilateral triangle with known dimensions drawn on the road.

Solutions based on vehicle movement typically detect vanishing points in the direction towards the vehicle motion (first VP) and in directions perpendicular to it (second and third VP). Schoepflin and Dailey [48] use a background model to detect lane boundaries in the activity map. The intersection of lanes is assumed to be the vanishing point of vehicle motion. The second vanishing point is detected from vehicle edges. One known length in the scene is required for full calibration.

Dubská et al. [12] proposed a fully automatic method for calibration. It assumes straight movement of cars on the road; this condition is usually met on freeways, but it cannot be assumed in parking lots, roundabouts and similar scenarios. Their method uses a particular form of cascaded Hough Transform [11] to search for vanishing points and the scene scale is inferred from the mean size of observed vehicles. Sochor et al. [53] extended this method by more accurate detection of vanishing points and scale inference. They use fine-grained recognition of vehicles and align the bounding box of a known 3D geometry to the observations in the image. The accuracy of this method is sufficient for speed measurement with mean error of 1.1 kph.

Our method is to some extent similar to *AutoCalib* [3], which also observes passing vehicles and forms the calibration. Their camera calibration is optimized by minimizing the re-projection error of known average 3D positions of detected keypoints on the vehicles from the rear side. However, *AutoCalib* must be given the focal length f as its input and it is limited to a coherent view of the vehicles (roughly from the rear). Our goal is to make the calibration algorithm fully automatic (not requiring f), to refrain from any assumptions about the vehicle's direction or viewpoint, and to make it accurate enough to be actually usable (the accuracy of *AutoCalib* reported by the authors in their article is 8.98 %).

Fine-Grained Vehicle Recognition

Recently, a number of methods for fine-grained recognition of various objects were published [14, 30, 31, 50], even for vehicles in particular [32, 34, 43, 52, 54, 64].

The task of fine-grained recognition (classification) benefits from extra image information provided by parts of classified objects. However, it cannot be assumed that the location of such parts is known in advance nor that the location is the same for all objects of the same type. Simon and Rodner [50] proposed a method how to deal with this problem (during training & test time) by automatic discovery and localization of such parts.

Lin et al. [31] and Gao et al. [14] approach this problem differently by using *Bilinear Pooling*. Lin et al. [31] use a bilinear classifier [45] to classify features extracted by convolutional layers from a CNN. Gao et al. [14] improved this idea and proposed the method for *Com-*

pact Bilinear Pooling reducing the number of features used while preserving the accuracy of classification. The method proposed by Lin et al. [30] uses three different CNNs for localization, alignment, and classification of images for general object recognition. Sochor et al. [54] show that these general fine-grained methods are not accurate enough for the task of vehicle fine-grained recognition and specialized approaches must be used.

A considerable group of existing fine-grained recognition algorithms are specialized on classification of vehicles. Some of them are limited to frontal/rear images of vehicles: Pearce and Pears [43] use the detection of license plates to localize the front/rear part of the vehicle for feature extraction, as these parts are usually very discriminative for the purpose of recognizing the vehicles. Directly extracted features from frontal images of cars and exploiting common structure of the vehicle's frontal mask are presented in the work by Zhang [64]. A more complex method based on optimizing/fitting vehicle 3D CAD model to image data for fine-grained classification was proposed by Lin et al. [32].

State-of-the-art results in this field are achieved by methods based on Convolutional Neural Networks. Liu et al. [34] proposed to use *Deep Relative Distance* trained on the re-identification task to extract more discriminative feature vectors, and *Coupled Clusters Loss* function during training. On the other hand, Sochor et al. [54] improve the classification accuracy using an "unwrapped" version of the 3D bounding box of detected vehicles to 2D plane as additional input of CNN for fine-grained recognition and other modifications.

Detection of Objects and Vehicles

Convolutional Neural Networks dominate also in the task of object detection by their accuracy [8, 15, 16, 18, 35, 46, 47, 58]. All of these networks can be divided into three main *meta-architectures* based on their behavior. The *feature extractor* is the first part of a detection network and it is common for all meta-architectures. Feature extractor can be any of available CNNs (e.g. VGG-16 [51], Inception v2 [23], Inception v3 [59], ResNet-101 [18], MobileNet [22], etc.).

The first meta-architecture is covered by the term *Single Shot Detector* (SSD). Although the term SSD was used as the name of the detector published by Liu et al. [35], it can denote the whole class of detectors which use a single feed-forward CNN to directly predict classes without a second stage classification operation processing the proposed boxes. Typical representatives of this group (aside the original SSD detector) are also YOLO [46] or Multibox [58] and the Region Proposal Network (RPN) stage of the Faster R-CNN [47], which are used to predict class-independent box proposals.

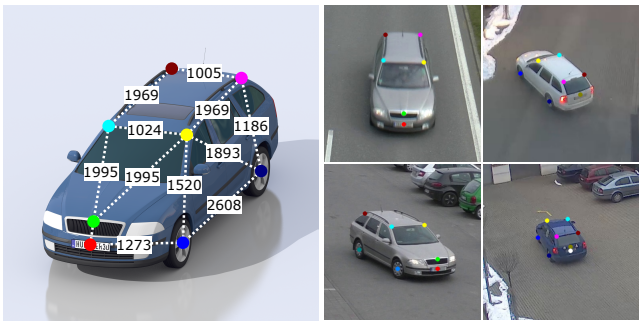

Fig. 2 *left:* Distances $\hat{\delta}$ (in millimeters) in the 3D model of *Skoda Octavia mk2* car. Only a small subset of all the distances in the set $\hat{\mathcal{K}}^{c_i}$ is shown. *right:* Examples of detected landmarks on actual vehicles of the same type.

The second meta-architecture is represented by *Faster R-CNN* [47]. It is the evolution of R-CNN [16] and Fast R-CNN [15]. In this setting, detection is divided into two stages. In the first stage, called the *Region Proposal Network* (RPN), features are extracted from the image by an intermediate level of feature extractor and they are used to predict class-independent box proposals. In the second stage, these box proposals are used to crop extracted features from the feature map which are passed to the following levels of the feature extractor to predict classes and class-related boxes. A number of works have been based on the Faster R-CNN meta-architecture since 2015 [2, 7, 18, 35, 49, 63] including SSD and R-FCN.

The last meta-architecture is called *Region-based Fully Convolutional Networks* (R-FCN) [8] and it follows the idea of the Faster R-CNN. However, crops are taken from the last layer preceding the prediction step, instead of cropping the features from the same level where region proposals were predicted. This step reduces the per-region computation which leads to faster prediction then in the case of Faster R-CNN, while accuracies are comparable.

3 Calibration from Landmarks

Our calibration method is based on detection and classification of vehicles in the video frames. Each video frame is processed by a neural network for detection and localization of vehicles (Faster R-CNN by Ren et al. [47], in our case trained on the COD20k dataset published by Juránek et al. [24]). The detected vehicles are classified into fine-grained classes (make & model & submodel & model year) by using our previously published vehicle classifier [54]. For the most common models, landmarks are localized in vehicles' positions by another neural

network by Wang et al. [61]. This neural network localizes 20 different landmarks in the image with a vehicle present. Some landmarks can be occluded in the given frame; the network also decides about the landmarks' visibility and only visible landmarks are further used.

It should be noted that although we describe the individual tasks of car detection, recognition and landmark localization as decoupled, in a production implementation, they could and should be merged to a single neural network predicting for each car its type and locations of the landmarks at the same time, similarly to Mask R-CNN [19] or the Panoptic Feature Pyramid Networks [26], which predict a binary mask for each detection.

Our solution is based on 2D-3D correspondences and thus the landmarks must be as precise as possible. Some of the localized landmarks have an ambiguous 3D position – for example headlight or fog lamp are hard to localize in the 3D space. Therefore, not all 20 keypoints are used, only the 12 most usable landmarks (4 wheels, 2 license plates, 2 logos, 4 corners of vehicle top).

The video is transformed into a set of cars' observations:

$$\mathcal{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_N\} \tag{2}$$

and for each car $\mathbf{c}_i$, a set of 2D landmark locations detected by the neural network [61] is available:

$$\bar{\mathcal{K}}^{\mathbf{c}_i} = \left\{\bar{\mathbf{k}}_1^{\mathbf{c}_i}, \ldots, \bar{\mathbf{k}}_K^{\mathbf{c}_i}\right\}. \tag{3}$$

Only vehicles of several (9 in our case) most common car models are included in $\mathcal{C}$, for which precise 3D CAD models are available (details of the particular dataset are in Section 4). These 3D models were manually processed to obtain the accurate 3D positions of the landmarks. Contrary to *AutoCalib* [3], these locations differ for different recognized vehicle models. For each of the observed vehicles $\mathbf{c}_i$, the correct 3D positions in the vehicle's local coordinate system are available:

$$\hat{\mathcal{K}}^{\mathbf{c}_i} = \left\{\hat{\mathbf{k}}_1^{\mathbf{c}_i}, \ldots, \hat{\mathbf{k}}_K^{\mathbf{c}_i}\right\}. \tag{4}$$

These precise 3D positions of the landmarks define the reference 3D distances of the landmarks (pairs of landmarks, identified by indices a and b) as:

$$\hat{\delta}(\mathbf{c}_i, a, b) = \left|\hat{\mathbf{k}}_a^{\mathbf{c}_i}, \hat{\mathbf{k}}_b^{\mathbf{c}_i}\right|, \tag{5}$$

which is the Euclidean distance of any two potential landmarks a and b in the model coordinate system. An example of these distances $\hat{\delta}(\mathbf{c}_i, a, b)$, together with a few samples of the detected 2D landmarks $\bar{\mathcal{K}}^{\mathbf{c}_i}$ can be seen in Figure 2.

For each landmark's projection $\bar{\mathbf{k}}_j^{\mathbf{c}_i}$, it is possible to compute its 3D position in the world coordinate system based on its known height above the ground plane (the Z-coordinate in the 3D model $\hat{\mathbf{k}}_j^{\mathbf{c}_i}$) and based on given calibration parameters ϕ. This reconstructed 3D position vector will be denoted as $\mathbf{k}_j^{\mathbf{c}_i}(\phi)$, as it is a function of the calibration parameters ϕ, consisting of focal length f (forming the intrinsic matrix $\mathbf{K}_\phi$), rotation matrix $\mathbf{R}_\phi$ and translation vector $\mathbf{t}_\phi$.

Starting with the camera projection (1), in our notation:

$$\lambda \begin{bmatrix} \bar{\mathbf{k}}_j^{\mathbf{c}_i} \\ 1 \end{bmatrix} = \mathbf{K}_\phi \left(\mathbf{R}_\phi \mathbf{k}_j^{\mathbf{c}_i}(\phi) + \mathbf{t}_\phi\right), \tag{6}$$

which can be rearranged to

$$\mathbf{R}_\phi^{-1}\mathbf{K}_\phi^{-1}\,\lambda \begin{bmatrix} \bar{\mathbf{k}}_j^{\mathbf{c}_i} \\ 1 \end{bmatrix} = \mathbf{k}_j^{\mathbf{c}_i}(\phi) + \mathbf{R}_\phi^{-1}\mathbf{t}_\phi \tag{7}$$

and further:

$$\mathbf{k}_j^{\mathbf{c}_i}(\phi) = \mathbf{R}_\phi^{-1}\left(\mathbf{K}_\phi^{-1}\,\lambda \begin{bmatrix} \bar{\mathbf{k}}_j^{\mathbf{c}_i} \\ 1 \end{bmatrix} - \mathbf{t}_\phi\right). \tag{8}$$

The projective scale λ can be expressed from eq. (7) by using the Z coordinate known from the CAD model $\hat{\mathbf{k}}_j^{\mathbf{c}_i}$. Only the third component of all the column vectors is used from (7) (operator $[\mathbf{x}]_3$ symbolizes extraction of the third member):

$$\lambda = \frac{\left[\hat{\mathbf{k}}_j^{\mathbf{c}_i}\right]_3 + \left[\mathbf{R}_\phi^{-1}\mathbf{t}_\phi\right]_3}{\left[\mathbf{R}_\phi^{-1}\mathbf{K}_\phi^{-1}\begin{bmatrix} \bar{\mathbf{k}}_j^{\mathbf{c}_i} \\ 1 \end{bmatrix}\right]_3}. \tag{9}$$

For each car $\mathbf{c}_i$ and each pair of landmarks a, b in the world coordinate system $\mathbf{k}_j^{\mathbf{c}_i}(\phi)$, their 3D distance can then be computed as:

$$\delta(\mathbf{c}_i, a, b, \phi) = \left|\mathbf{k}_a^{\mathbf{c}_i}(\phi), \mathbf{k}_b^{\mathbf{c}_i}(\phi)\right|. \tag{10}$$

Although localization of the landmarks works well enough (according to [61], 88.8 % of landmarks are correctly predicted within 3 pixels), it fails significantly in some cases. In particular, it leads to considerable outliers in the detection which can impact the calibration process significantly. For this reason, we propose to compute the re-projection error for each car $\mathbf{c}_i$ and transform it to a weight, controlling the impact of the given sample on the whole calibration. The *PnP* solver [29] provides extrinsic camera parameters for each vehicle instance. Given those, the 3D points from $\hat{\mathcal{K}}^{\mathbf{c}_i}$ are projected to the image plane similarly to eq. (6), yielding:

$$\tilde{\mathcal{K}}^{\mathbf{c}_i} = \left\{\tilde{\mathbf{k}}_1^{\mathbf{c}_i}, \ldots, \tilde{\mathbf{k}}_K^{\mathbf{c}_i}\right\}. \tag{11}$$

The particular vehicle instance $\mathbf{c}_i$ is then assigned its individual normalized re-projection error:

$$\epsilon(\mathbf{c}_i) = \sqrt{\frac{\sum_{j=1}^{K} \left| \tilde{\mathbf{k}}_j^{\mathbf{c}_i}, \bar{\mathbf{k}}_j^{\mathbf{c}_i} \right|}{\sum_{j=1}^{K} \left| \tilde{\mathbf{k}}_j^{\mathbf{c}_i}, \mathbf{K}^{\mathbf{c}_i} \right|}}, \tag{12}$$

where $\mathbf{K}^{\mathbf{c}_i}$ is the mean of all the points $\bar{\mathbf{k}}^{\mathbf{c}_i}$. The fraction normalizes the re-projection error so that vehicle instances of different sizes are mutually comparable. The *PnP* computation assumes knowledge of the intrinsic matrix which is for now considered to be known and its estimation is discussed later.

For each vehicle $\mathbf{c}_i$ its normalized re-projection error $\epsilon(\mathbf{c}_i)$ is computed by (12), which defines its *weight*:

$$\mathbf{w}^{\mathbf{c}_i} = \left(\frac{1}{\epsilon(\mathbf{c}_i)} \right)^{\alpha}, \tag{13}$$

where α controls the power of the weight and its effect is studied in Section 5.1.

The total error/cost of the observations in the video given some calibration parameters ϕ can be expressed as follows:

$$\varepsilon(\phi) = \frac{1}{\mathbf{W}} \sum_{\mathbf{c}_i \in \mathcal{C}} \sum_{a,b} \left(\frac{\delta(\mathbf{c}_i, a, b, \phi) - \hat{\delta}(\mathbf{c}_i, a, b)}{\hat{\delta}(\mathbf{c}_i, a, b)} \right)^2 \mathbf{w}^{\mathbf{c}_i}, \tag{14}$$

where $\mathbf{W} = \sum_{\mathbf{c}_i \in \mathcal{C}} \mathbf{w}^{\mathbf{c}_i}$ and thus (14) is the weighted mean of vehicles' reconstruction errors. The process of calibration consists of finding such parameters ϕ that minimize this error function. In our experiments, we find the parameters ϕ by *Differential Evolution* [57] minimizing (14). Any other global optimization method can be used; differential evolution was chosen due to its fast computation and robustness. We experimented with local optimizers as well (gradient descent, AdaGrad [5], Adam [25], L–BFGS [33]), but they failed so the problem appears to be considerably non-linear.

Since available 3D positions of landmarks $\hat{\mathcal{K}}$ are located in the vehicle's (local) coordinate system, projection of these 3D points to the image plane based of the calibration parameters ϕ (by eq. (6)) would project all points near the world coordinate origin (3D coordinates are not available in the world coordinate system). However, detected 2D landmarks' positions $\bar{\mathcal{K}}$ are localized in the whole world coordinate system (whole image plane), and thus these projected position do not correspond to the localized ones. Due to this fact it is necessary to use the reconstruction error computed from detected 2D landmarks' positions $\bar{\mathcal{K}}$ instead of the re-projection error in the image plane.

Fig. 3 Sample images from *BrnoCompSpeed* dataset [55].

As was mentioned before, weights (13) are used in the process of calibration parameters optimization (14). However, the computation of weights needs projected points $\tilde{\mathcal{K}}^{\mathbf{c}_i}$ which are computed by a *PnP* solver and thus the knowledge of the focal length is necessary. Since the focal length value is assumed to be unknown, it must be estimated first, and only after it, the weights $\mathbf{w}^{\mathbf{c}_i}$ can be computed. Therefore, the whole calibration process is twofold; in the first pass, all weights $\mathbf{w}^{\mathbf{c}_i}$ are set to the value 1.0. The estimated value of focal length in the first pass is used for computing the more accurate weights $\mathbf{w}^{\mathbf{c}_i}$ by eq. (12), and these weights are used during the second pass of the calibration process.

The usage of two iterations of computation (and also proper weights) seem to be beneficial, since the error is reduced approximately by 53 % as is described in Section 5.2.

4 Datasets

Two datasets were used for evaluation of the proposed method: *BrnoCompSpeed* dataset published by Sochor et al. [55], which contains recordings of highways and is made publicly available and our novel *BrnoCarPark* dataset with recordings of parking lots. The latter is made public along with publication of this article.

BrnoCompSpeed

This dataset was made for speed measurement of vehicles by a single monocular camera. It contains video recordings of highways captured from the bridge above the highway in traffic surveillance manner, with ground truth measurements on road plane and 20,865 vehicles with ground truth speed (see Figure 3 for a few samples). The dataset was shot for approximately one hour at seven different locations with three cameras at each location, making ≈ 21 hours of recordings in total. However, the dataset only contains straight roads because of its purpose of vehicle speed measurement and thus

Fig. 4 Sample images from the new *BrnoCarPark* dataset (best viewed in digital). The data include 11 videos from parking lots with different traffic density, parking occupancy, etc.

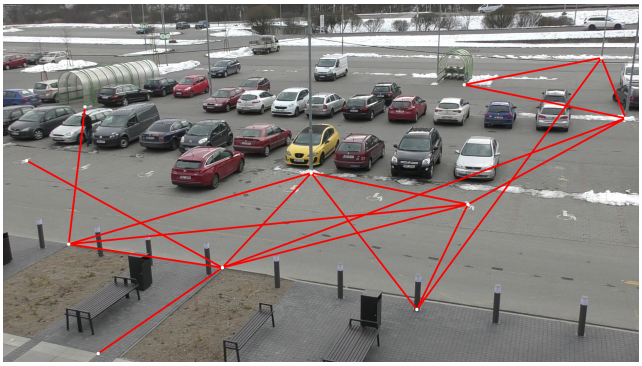

Fig. 5 Example of ground-truth distance measurements in the *BrnoCarPark* dataset.

the movement of the captured vehicles is limited to one direction only.

BrnoCarPark

In order to be able to better evaluate the proposed methodology, a novel challenging calibration dataset *BrnoCarPark* was created. This dataset contains recordings of parking lots with vehicles passing in front of the camera randomly (see Figure 4). Therefore, the cars are not moving in any single dominant direction and the extraction of a single set of vanishing points for the whole scene is impossible. The recordings were captured at two locations from different viewpoints (sessions), during various times of the day and somewhat diverse weather conditions. We make the dataset publicly available on our website[1].

4.1 Ground Truth for Evaluation

Both of the datasets are equipped with ground-truth measurements in the ground plane, which make the evaluation of the calibration algorithms possible. The ground truth data consist of measurements between various points in the real scene's ground plane and corresponding 2D positions of the points in the image plane. The existing *BrnoCompSpeed* dataset is equipped with $4-10$ measurements in each camera view, typically in the direction of the vehicles' movement and in the direction perpendicular to it. For our new *BrnoCarPark* dataset, we chose a number of distinctive points in the camera images and measured distances between them when the parking lot was empty. We used a laser distance measurer with precision of ± 2 mm declared by its manufacturer. For each scene, $8-19$ distance measurements are available. One example of the ground truth distance measurements with marked 2D points in the frame is depicted in Figure 5.

[1] https://medusa.fit.vutbr.cz/traffic

5 Results

As explained in Section 4.1, each of the testing scenes is equipped with real-world measurements of distances between distinguishable points in the ground plane (see Figure 5 for an example):

$$\hat{\mathcal{D}} = \left\{\hat{d}_1, \ldots, \hat{d}_D\right\}. \tag{15}$$

The 2D endpoints of these measurements can be projected from the image plane to the ground plane by the calibration parameters obtained by our method, by using the same technique as described in Section 3, eq. (8). The measurements between the points re-projected in this manner

$$\mathcal{D} = \{d_1, \ldots, d_D\} \tag{16}$$

can be evaluated against the ground truth $\hat{\mathcal{D}}$ by measuring the relative root mean square error:

$$\text{RMSE} = \sqrt{\frac{1}{D}\sum_{i=1}^{D}\left(\frac{d_i - \hat{d}_i}{\hat{d}_i}\right)^2}, \tag{17}$$

similarly to *AutoCalib* [3] and thus we can compare our results with *AutoCalib* results. It should be noted that direct comparison with "classical" calibration approaches [65] is impossible, because a large checkerboard (as wide as several meters) is not feasible, and when the camera is calibrated from a near distance with letter paper-sized checkerboards, it re-focuses and changes its parameters.

Although all the real-world measurements and the corresponding annotations of the 2D points in the scene images were made as precise as possible, some inaccuracies must inevitably occur. In order to quantify these, we made a calibration based on the 2D ground-truth measurements $\hat{\mathcal{D}}$, by using the same methodology as described in Section 3 (with all the point's Z coordinate being 0). The ground truth calibration errors in the individual scenes are plotted in Figure 6 as the red bars. The same graph also shows the error (17) of our method for all the scenes (as measured by the more or less accurate ground truth).

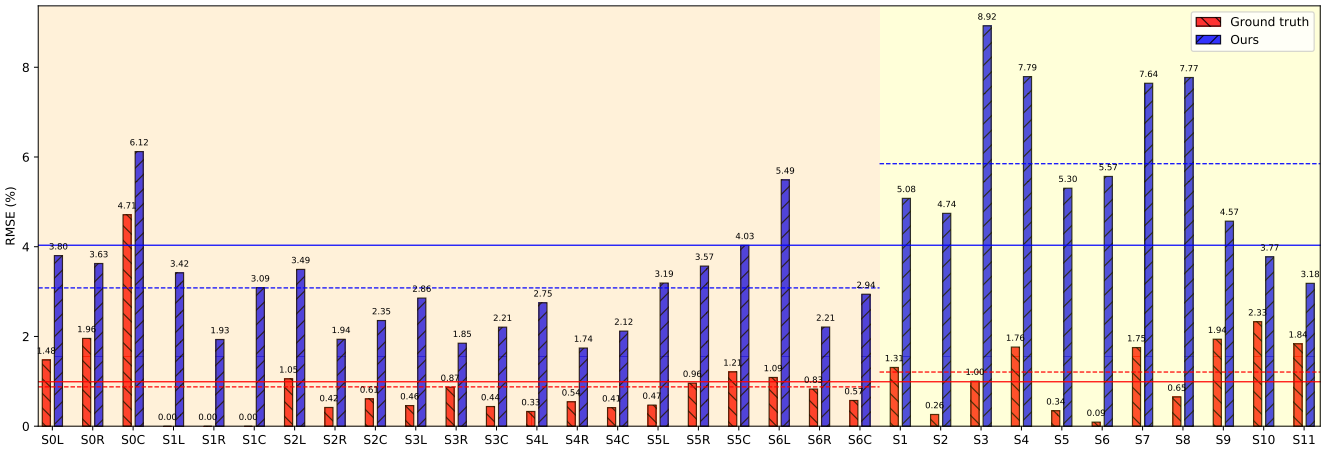

Fig. 6 Accuracy of the proposed calibration method vs ground truth calibration. The red bars describe ground truth calibration and the blue bars describe proposed algorithm accuracy; red and blue horizontal lines are averages per dataset (dashed) and over all measurements (solid). All is evaluated on *BrnoCompSpeed* (left part) dataset and new *BrnoCarPark* dataset (right part). The ground truth calibration has an average *RMSE* of 0.99 % and the proposed method has an average *RMSE* of 4.03 %.

We compared our proposed method to *AutoCalib* – the state-of-the-art alternative solution. Since the authors did not make *AutoCalib* code public, we reimplemented their algorithm according to their paper [3]. We used the same landmarks as in our method and also shared the 3D models. We assume that this should constitute an improvement to the original *AutoCalib* method, because then only rear views of the vehicles were used and the set of the landmarks was thus greatly limited. Also, the authors of *AutoCalib* used one united 3D model representing all sedan vehicles by estimating some average landmark positions and they do not distinguish between individual vehicle models. Focal length values are necessary for *AutoCalib* method – these are available within *BrnoCompSpeed* dataset, for the *BrnoCarPark* dataset, focal length values computed by ground truth estimation, as was mentioned earlier in this section, were used.

The comparison of our method with *AutoCalib* is shown in Figure 7. The mean RMSE across all the scenes was decreased from 6.56 % by *AutoCalib* to 4.03 % by our approach. It should be noted that Bhardwaj et al. [3] report the error of 8.98 % on their data in their paper; our implementation thus seems on par or even slightly better than the original solution (though it should be noted that the evaluation dataset is different).

In all our experiments, the *Differential Evolution* parameters are set as follows: population size (number of parents, NP) – 15 times the number of parameters (75); crossover probability (CR) – 0.9; dither technique for setting weighting factor F is used and values are randomly selected for each generation from the interval $[0.5, 1.0]$; the method for creating trial candidates is $DE/best/1/bin$ (the notation and meaning of all the parameters is explained by Storn and Price [57]).

5.1 Weighting Parameter α Used for Calibration

As explained in Section 3, it is beneficial to use weights $\mathbf{w}^{\mathbf{c}_i}$ of the observed vehicles during calibration process. These weights are meant to suppress the influence of vehicles whose landmarks were detected inaccurately. Figure 8 shows the result of an experiment designed to look for the proper parameter α used for the calibration (13). Different parameters α were tested and for each of them, the plot shows the distribution of the errors (17) across all the scenes shown in Figures 6 and 7. The blue boxplot shows the median value (black central line) and quartils; the red dotted line in each box shows the average error across all the scenes. Hollow circles show major outliers – scenes that notably failed.

As can be seen from eq. (13), parameter α emphasizes vehicles with a smaller re-projection error (12) and suppresses vehicles with a higher error. It appears that small values of α lead to instability caused by using the majority of vehicles which can also contain very noisy landmarks detections, but on the other hand, large α tends to use very few vehicles with the smallest re-projection error and not exploiting extra information from the other vehicles detected. Therefore in the experiments reported here, $\alpha = 4$ was used. It should be mentioned that the average error across all the scenes is smaller than *AutoCalib* result 6.56 % in all possible tested values of parameter α.

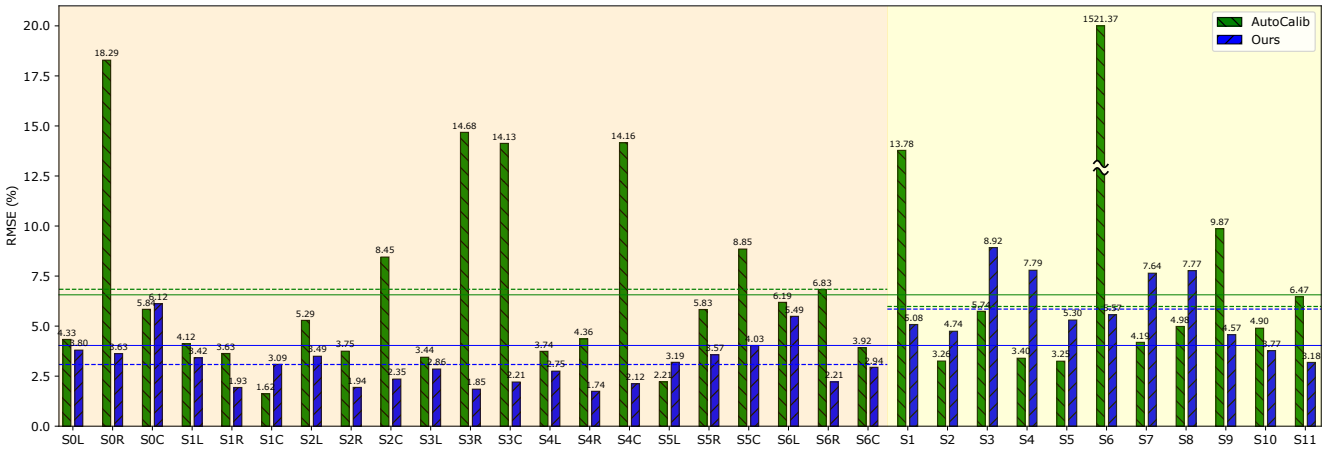

Fig. 7 Comparison of accuracy for the proposed and *AutoCalib* method. Average results are as follows: *BrnoCompSpeed* dataset – *AutoCalib* 6.84 %, ours 3.08 %; *BrnoCarPark* dataset – *AutoCalib* 5.98 %, ours 5.84 % (it should be mentioned that *AutoCalib* failed significantly on scene *S6* and thus this single scene is not evaluated for *AutoCalib* method); both datasets – *AutoCalib* 6.56 %, ours 4.03 %.

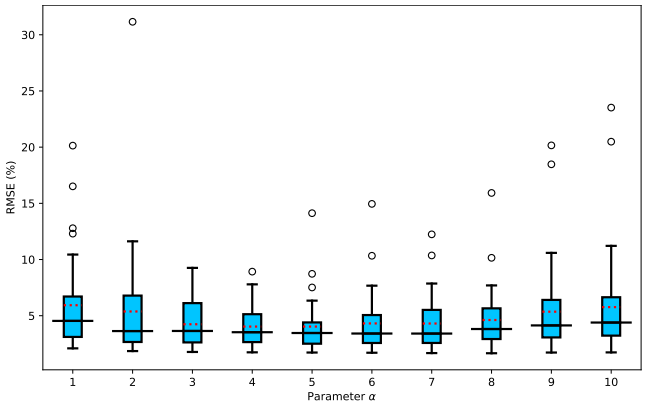

Fig. 8 Calibration error with different values of parameter α used for calibration.

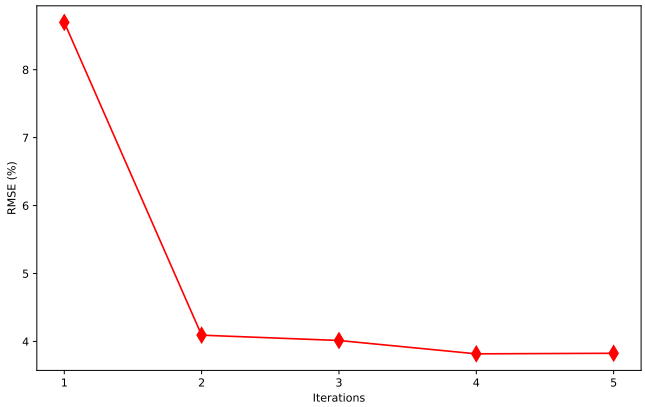

Fig. 9 Error with different count of calibration iterations.

5.2 Number of Calibration Iterations

At the end of Section 3, we described that the whole calibration process is twofold. In the first step, the calibration process is computed with weights $\mathbf{w}^{\mathbf{c}_i}$ set to the value 1.0 and in the second step the weights are computed with more precise focal length value obtained in the first step. We also made an experiment if more iterations with re-computation of the weights with new value of the focal length are beneficial.

Figure 9 shows the result of the experiment; it is apparent that one iteration of calibration with weights set to the value 1.0 produces a much higher error than the usage of multiple iterations. It seems that a higher number of iterations is not so beneficial, where the largest error reduction is between one and two iterations (about 53 % error reduction), thus two iterations are used within our experiments. The high reduction of error between one and two iterations also shows the advantage of the weights used, as in the case of one iteration, weights are neglected (set to 1.0).

5.3 Study of Calibration Parameters

Till now we considered almost perfect camera with principal point in the middle of image plane and with no distortion present. However, in real scenarios, cameras can suffer by distortion and thus we experimented how the proposed method can handle this situation. We extended the described method so that it also estimates the principal point and the distortion parameters. In this setting, intrinsic matrix $\mathbf{K}$ contains also the principal point and before the computation of 3D distances (10), localized 2D landmarks are undistorted by the distortion parameters. Since the experiment should only prove the ability to extend the proposed method by other calibration parameters, only two radial distortion parameters k_1, k_2 are used (different models of camera distortion work with different numbers of parameters).

Calibration parameters	Data	Error
f, r_x, r_y, r_z, t_z	Original	4.03 %
f, r_x, r_y, r_z, t_z	Distorted	5.15 %
$f, r_x, r_y, r_z, t_z, p_x, p_y, k_1, k_2$	Original	4.01 %
$f, r_x, r_y, r_z, t_z, p_x, p_y, k_1, k_2$	Distorted	4.05 %

Table 1 Error of variant with extended parameters and distorted data.

The first two distortion parameters are the most influencing ones [60] and thus these are crucial and we are not interested in the others. The computation of undistorted points with some distortion parameters can be used as follows:

$$\begin{aligned} x_u &= x_d(1 + k_1 r^2 + k_2 r^4) \\ y_u &= y_d(1 + k_1 r^2 + k_2 r^4), \end{aligned} \tag{18}$$

where $r^2 = x_u^2 + y_u^2$, (x_u, y_u) is undistorted point, and (x_d, y_d) is distorted point. These equations can be approximated by iterative algorithm and further details can be found in [4, 60].

We tested two settings — in the first one, the precision of calibration was tested on unchanged original dataset only with extension of the estimated calibration parameters. The second experiment should test how the method can deal with distortion and thus localized 2D landmarks in the image plane were distorted by randomly set distortion parameters k_1 and k_2 (in our experiment, random values with uniform distribution from ranges $(-1.0; 1.0)$ and $(0.0; 3.0)$ were used for k_1 and k_2 respectively). As can be seen in Table 1, our method seems to be able to handle both cases – both the extended number of parameters, and the case when the input data are really distorted. Although the results for the case of extended parameters estimation are slightly better, the main disadvantage of this approach is computational time — 4 additional parameters (p_x, p_y, k_1, k_2) must be estimated, and therefore the computation takes more than twice as much time.

6 Future Work

The results presented in Section 5 are a considerable improvement over the previous state-of-the-art method [3], but they are still not totally satisfying: both the accuracy of the ground truth (around 1 % error) and the results of the calibration themselves (around 4 % error evaluated by the mentioned ground truth). We have carried out preliminary experiments which should lead to considerable improvement of the accuracy. The calibration methodology presented in this article remains the same; the improvements are centered in providing better inputs to the algorithms: providing both better ground truth measurements and even more importantly, better landmarks on the vehicle.

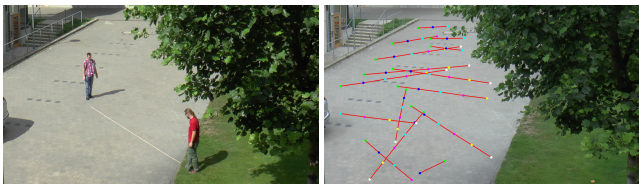

Fig. 10 Experiments with more precise ground truth calibration measurements. *left:* Two people place a rope with regular marks (five 1 m sections in this case) into multiple locations in front of the camera. *right:* In a short and uncomplicated process, a high number of precise measurements can be obtained – for each rope placement, one frame of the video gives several very accurate distance measurements in the ground plane by manual or automatic recognition of the distance marks.

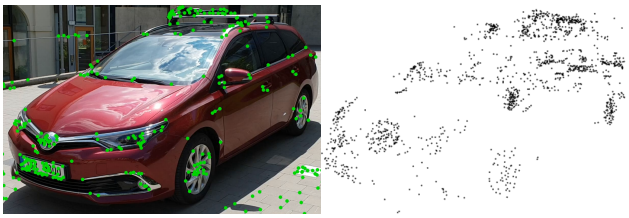

Fig. 11 SfM reconstruction of a particular vehicle model, later used for accurate landmark localization. *left:* one frame of the source video, with the keypoints marked by green, *right:* reconstructed point cloud; points are candidate landmarks for the calibration.

Firstly, Figure 10 illustrates the new approach to ground truth measurements. The motivation for the new design is twofold. The measurements done with a precisely constructed straight rope are more precise, because they do not rely on distances between ambiguous natural elements in the scene. Besides, this approach allows to obtain a multitude of measurements with small effort and in a short time. Each rope placement provides several (five in the shown case) measurements along one straight line, and the rope can be easily placed into multiple locations and orientations. The experiment captured in Figure 10 resulted in ground-truth calibration of around 0.45 % (cross-validated on the rope measurements).

Secondly, the keypoints/landmarks on the vehicles can be extracted specifically for the given type of the observed vehicle, not by the generic (and fairly imprecise) extractor used now [61]. The true 3D distances $\hat{\delta}(\mathbf{c}_i, a, b)$ are already make & model specific (Sec. 3), and therefore assuming fine-grained recognition of the vehicle ([52, 54]) does not constitute a new requirement. We made a detailed reconstruction of one vehicle (Toyota Auris SW 2017) from a walk-around video by using an existing Structure-from-Motion solution *Open-*

Fig. 12 Observations of a vehicle of known type, whose landmarks can be obtained accurately and their mutual distances are reconstructed precisely (Fig. 11).

MVG [40], see Figure 11. This vehicle moved randomly in the scene and 25 observations varying in the vehicle's location and orientation were selected, see Fig. 12. The landmarks/keypoints were manually extracted and their real-world 3D distances $\hat{\delta}(\mathbf{c}_i, a, b)$ were obtained from the point cloud reconstructed by the SfM. The calibration obtained by our algorithm from this input achieved error of 0.79 % (evaluated by the new ground truth, Figure 10). We are working on making this process fully automatic, but this preliminary experiment shows that the algorithm presented in this article can promise very usable accuracy, when the input data is sufficiently precise. The purpose of this section is to show that such precise input data indeed could be obtained.

7 Conclusion

This article presents a methodology for automatic calibration of a surveillance camera by observing vehicles. Contrary to previous solutions, the new method does not assume a particular view direction to the vehicles, it does not require straight motion of vehicles, and it does not require any extra information (such as the camera focal length). The solution was evaluated on a previously published dataset *BrnoCompSpeed*, where vehicles are moving in straight and mutually parallel trajectories. It was also measured on a set of videos collected at two parking lots called *BrnoCarPark*, where vehicles are occluded, varying in their observed size, and they move on arbitrary and very variable trajectories. When collecting this dataset, we measured multiple distances in the ground plane for obtaining a ground truth calibration. We make this set of videos, together with source codes, public for future research and for comparison.

The evaluation shows that the proposed approach is considerably more accurate than its predecessor *AutoCalib* [3] (error decreased from 6.56 % to 4.03 %); at the same time, our solution is more flexible and robust, and it does not require the camera's focal length as the input. Still the accuracy is not sufficient for many applications; we seem to be facing the limits imposed by the small amount of vehicle landmarks and especially by their inaccuracy. Experiments show that the proposed method can deal with camera distortion as well and estimate the distortion parameters.

In Section 6, we show preliminary experiments indicating that the methodology presented in this article could lead to a much more precise calibration, if the method is provided with more accurate and model-specific landmarks on the vehicles. We are currently investigating this possibility.

Acknowledgements 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 IT4Innovations excellence in science — LQ1602.

References

1. A Fischler M, C Bolles R (1981) Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24:381–395
2. Bell S, Lawrence Zitnick C, Bala K, Girshick R (2016) Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2874–2883
3. Bhardwaj R, Tummala GK, Ramalingam G, Ramjee R, Sinha P (2017) AutoCalib: Automatic traffic camera calibration at scale. In: The 4th ACM International Conference on Systems for Energy-Efficient Built Environments (BuildSys 2017)
4. Bukhari F, Dailey M (2013) Automatic radial distortion estimation from a single image. Journal of Mathematical Imaging and Vision 45, DOI 10.1007/s10851-012-0342-2
5. C Duchi J, Hazan E, Singer Y (2011) Adaptive subgradient methods for online learning and stochastic optimization. Journal of Machine Learning Research 12:2121–2159
6. Cathey F, Dailey D (2005) A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras. In: Intelligent Vehicles Symposium, pp 777–782, DOI 10.1109/IVS.2005.1505199
7. Dai J, He K, Sun J (2016) Instance-aware semantic segmentation via multi-task network cascades. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3150–3158
8. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional net-

works. In: Advances in Neural Information Processing Systems (NIPS), pp 379–387
9. Dailey D, Cathey F, Pumrin S (2000) An algorithm to estimate mean traffic speed using uncalibrated cameras. IEEE Transactions on Intelligent Transportation Systems 1(2):98–107, DOI 10.1109/6979.880967
10. Do VH, Nghiem LH, Thi NP, Ngoc NP (2015) 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, pp 1–5
11. Dubská M, Herout A (2013) Real projective plane mapping for detection of orthogonal vanishing points. In: British Machine Vision Conference (BMVC), The British Machine Vision Association and Society for Pattern Recognition, pp 1–10
12. Dubská M, Sochor J, Herout A (2014) Automatic camera calibration for traffic understanding. In: British Machine Vision Conference (BMVC)
13. Filipiak P, Golenko B, Dolega C (2016) NSGA-II based auto-calibration of automatic number plate recognition camera for vehicle speed measurement. In: EvoApplications 2016, Springer International Publishing, pp 803–818, DOI 10.1007/978-3-319-31204-0_51
14. Gao Y, Beijbom O, Zhang N, Darrell T (2016) Compact bilinear pooling. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
15. Girshick R (2015) Fast r-cnn. In: The IEEE International Conference on Computer Vision (ICCV), pp 1440–1448
16. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
17. Grammatikopoulos L, Karras G, Petsa E (2005) 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, pp 332–338
18. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
19. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: The IEEE International Conference on Computer Vision (ICCV), pp 2980–2988, DOI 10.1109/ICCV.2017.322
20. He XC, Yung NHC (2007) A novel algorithm for estimating vehicle speed from two consecutive images. In: IEEE Workshop on Applications of Computer Vision, WACV, DOI 10.1109/WACV.2007.7
21. Hesch JA, Roumeliotis SI (2011) A direct least-squares (dls) method for pnp. In: The IEEE International Conference on Computer Vision (ICCV), pp 383–390, DOI 10.1109/ICCV.2011.6126266
22. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861
23. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:150203167
24. Juránek R, Herout A, Dubská M, Zemčík P (2015) Real-time pose estimation piggybacked on object detection. In: The IEEE International Conference on Computer Vision (ICCV)
25. Kingma D, Ba J (2014) Adam: A method for stochastic optimization. International Conference on Learning Representations
26. Kirillov A, Girshick RB, He K, Dollár P (2019) Panoptic feature pyramid networks. CoRR abs/1901.02446
27. Kneip L, Li H, Seo Y (2014) Upnp: An optimal o(n) solution to the absolute pose problem with universal applicability. In: Fleet D, Pajdla T, Schiele B, Tuytelaars T (eds) Computer Vision – ECCV 2014, Springer International Publishing, Cham, pp 127–142
28. Lan J, Li J, Hu G, Ran B, Wang L (2014) Vehicle speed measurement based on gray constraint optical flow algorithm. Optik – International Journal for Light and Electron Optics 125(1):289 – 295, DOI 10.1016/j.ijleo.2013.06.036
29. Lepetit V, Moreno-Noguer F, Fua P (2008) Epnp: An accurate o(n) solution to the pnp problem. International Journal of Computer Vision (IJCV) 81:155–166
30. Lin D, Shen X, Lu C, Jia J (2015) Deep LAC: Deep localization, alignment and classification for fine-grained recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
31. Lin TY, RoyChowdhury A, Maji S (2015) Bilinear CNN models for fine-grained visual recognition. In: The IEEE International Conference on Computer Vision (ICCV)
32. Lin YL, Morariu VI, Hsu W, Davis LS (2014) Jointly optimizing 3D model fitting and fine-grained classification. In: European Conference on Computer Vision (ECCV)

33. Liu DC, Nocedal J (1989) On the limited memory bfgs method for large scale optimization. Mathematical Programming 45(1):503–528, DOI 10.1007/BF01589116
34. Liu H, Tian Y, Yang Y, Pang L, Huang T (2016) Deep relative distance learning: Tell the difference between similar vehicles. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
35. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: European Conference on Computer Vision (ECCV), Springer, pp 21–37
36. Llorca DF, Salinas C, Jimenez M, Parra I, Morcillo AG, Izquierdo R, Lorenzo J, Sotelo MA (2016) Two-camera based accurate vehicle speed measurement using average speed at a fixed point. In: 19th International IEEE Conference on Intelligent Transportation Systems (ITSC), DOI 10.1109/ITSC.2014.6958187
37. Luvizon D, Nassu B, Minetto R (2014) Vehicle speed estimation by license plate detection and tracking. In: Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on, pp 6563–6567, DOI 10.1109/ICASSP.2014.6854869
38. Maduro C, Batista K, Peixoto P, Batista J (2008) Estimation of vehicle velocity and traffic intensity using rectified images. In: 15th IEEE International Conference on Image Processing (ICIP), pp 777–780, DOI 10.1109/ICIP.2008.4711870
39. Meng X, Hu Z (2003) A new easy camera calibration technique based on circular points. Pattern Recognition 36:1155–1164, DOI 10.1016/S0031-3203(02)00225-X
40. Moulon P, Monasse P, Marlet R (2013) Global fusion of relative motions for robust, accurate and scalable structure from motion. In: The IEEE International Conference on Computer Vision (ICCV), pp 3248–3255, DOI 10.1109/ICCV.2013.403
41. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: European Conference on Computer Vision (ECCV), Springer, pp 483–499
42. Nurhadiyatna A, Hardjono B, Wibisono A, Sina I, Jatmiko W, Ma'sum M, Mursanto P (2013) Improved vehicle speed estimation using gaussian mixture model and hole filling algorithm. In: Advanced Computer Science and Information Systems (ICACSIS), 2013 International Conference on, pp 451–456, DOI 10.1109/ICACSIS.2013.6761617
43. Pearce G, Pears N (2011) Automatic make and model recognition from frontal images of cars. In: IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp 373–378, DOI 10.1109/AVSS.2011.6027353
44. Penate-Sanchez A, Andrade-Cetto J, Moreno-Noguer F (2013) Exhaustive linearization for robust camera pose and focal length estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 35(10):2387–2400, DOI 10.1109/TPAMI.2013.36
45. Pirsiavash H, Ramanan D, Fowlkes CC (2009) Bilinear classifiers for visual recognition. In: Bengio Y, Schuurmans D, Lafferty J, Williams C, Culotta A (eds) Advances in Neural Information Processing Systems (NIPS), Curran Associates, Inc., pp 1482–1490
46. Redmon J, Divvala SK, Girshick RB, Farhadi A (2015) You only look once: Unified, real-time object detection. Computing Research Repository (CoRR) abs/1506.02640
47. Ren S, He K, Girshick R, Sun J (2015) Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems (NIPS)
48. Schoepflin T, Dailey D (2003) Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation. IEEE Transactions on Intelligent Transportation Systems 4(2):90–98, DOI 10.1109/TITS.2003.821213
49. Shrivastava A, Gupta A, Girshick R (2016) Training region-based object detectors with online hard example mining. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 761–769
50. Simon M, Rodner E (2015) Neural activation constellations: Unsupervised part model discovery with convolutional networks. In: The IEEE International Conference on Computer Vision (ICCV)
51. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. Computing Research Repository (CoRR) abs/1409.1556
52. Sochor J, Herout A, Havel J (2016) BoxCars: 3D boxes as CNN input for improved fine-grained vehicle recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)
53. Sochor J, Juranek R, Herout A (2017) Traffic surveillance camera calibration by 3D model bounding box alignment for accurate vehicle speed measurement. Computer Vision and Image Understanding (CVIU) 161:87 – 98, DOI 10.1016/j.cviu.2017.05.015
54. Sochor J, Špaňhel J, Herout A (2017) BoxCars: Improving fine-grained recognition of vehicles us-

ing 3D bounding boxes in traffic surveillance. arXiv:1703.00686
55. Sochor J, Juránek R, Špaňhel J, Maršík L, Široký A, Herout A, Zemčík P (2018) Comprehensive data set for automatic single camera visual speed measurement. IEEE Transactions on Intelligent Transportation Systems pp 1–11, DOI 10.1109/TITS.2018.2825609
56. Song KT, Tai JC (2006) Dynamic calibration of Pan–Tilt–Zoom cameras for traffic monitoring. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 36(5):1091–1103, DOI 10.1109/TSMCB.2006.872271
57. Storn R, Price K (1997) Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11(4):341–359, DOI 10.1023/A:1008202821328
58. Szegedy C, Reed S, Erhan D, Anguelov D, Ioffe S (2014) Scalable, high-quality object detection. arXiv preprint arXiv:14121441
59. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2818–2826
60. de Villiers JP, Leuschner FW, Geldenhuys R (2008) Centi-pixel accurate real-time inverse distortion correction. In: Wen JT, Hodko D, Otani Y, Kofman J, Kaynak O (eds) Optomechatronic Technologies 2008, International Society for Optics and Photonics, SPIE, vol 7266, pp 320 – 327, DOI 10.1117/12.804771
61. Wang Z, Tang L, Liu X, Yao Z, Yi S, Shao J, Yan J, Wang S, Li H, Wang X (2017) Orientation invariant feature embedding and spatial temporal regularization for vehicle re-identification. In: The IEEE International Conference on Computer Vision (ICCV)
62. You X, Zheng Y (2016) An accurate and practical calibration method for roadside camera using two vanishing points. Neurocomputing DOI 10.1016/j.neucom.2015.09.132
63. Zagoruyko S, Lerer A, Lin TY, Pinheiro PO, Gross S, Chintala S, Dollár P (2016) A multipath network for object detection. arXiv preprint arXiv:160402135
64. Zhang B (2014) Classification and identification of vehicle type and make by cortex-like image descriptor HMAX. International Journal of Computational Vision and Robotics (IJCVR) 4:195–211
65. Zhang Z (2000) A flexible new technique for camera calibration. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 22
66. Zhang Z (2004) Camera calibration with one-dimensional objects. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(7):892–899
67. Zheng Y, Kneip L (2016) A direct least-squares solution to the PnP problem with unknown focal length. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1790–1798, DOI 10.1109/CVPR.2016.198
68. Zheng Y, Kuang Y, Sugimoto S, Åström K, Okutomi M (2013) Revisiting the pnp problem: A fast, general and optimal solution. In: The IEEE International Conference on Computer Vision (ICCV), pp 2344–2351, DOI 10.1109/ICCV.2013.291
69. Zheng Y, Sugimoto S, Sato I, Okutomi M (2014) A general and simple method for camera pose and focal length determination. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 430–437, DOI 10.1109/CVPR.2014.62