Camera Auto-Calibration Using Pedestrians and Zebra-Crossings

Michael Hödlmoser
CVL, Vienna University of Technology
hoedl@caa.tuwien.ac.at

Branislav Micusik
AIT Austrian Institute of Technology
branislav.micusik@ait.ac.at

Martin Kampel
CVL, Vienna University of Technology
kampel@caa.tuwien.ac.at

Abstract

In this paper we present a novel camera self-calibration technique to automatically recover intrinsic and extrinsic parameters of a static surveillance camera by observing a traffic scene. The scene must consist of one or more pedestrians and a zebra-crossing. We first extract a horizontal vanishing point and a vanishing line from a zebra-crossing. The observation of pedestrians allows calculating a so called vertical line of mass. All lines of mass are parallel in 3D space and therefore the vertical vanishing point can be estimated. The second horizontal vanishing point can be calculated by introducing the triangle spanned by three orthogonal vanishing points. All three vanishing points are then taken to gather the intrinsic parameters. The extrinsic parameters are calculated after the determination of the camera’s height from the distance between two zebra-crossing edges. By combining static and dynamic calibration objects, the method gets robust against outliers. This robustness in combination with the practicability is shown in our experiments, which are carried out by using synthetic and real data of different application scenarios.

1. Introduction

Camera calibration is known as the determination of the interrelationship between a reference plane and the camera coordinate system (extrinsic parameters) and between the camera and the image coordinate system (intrinsic parameters). Calculating these so called intrinsic and extrinsic parameters is an important task for 3D computer vision applications, which range from the area of surveillance networks (e.g., security scenarios or ambient assisted living) to autonomous robotics and ubiquitous network robotic devices. According to [8], the task of camera calibration can mainly be divided into two parts, namely calibration using a known calibration object and self-calibration. When the dimensions of an object are known, the 3D information of a scene can precisely be extracted by establishing correspondences between different views showing the same calibration object. In case of self-calibration, no calibration object is needed to calibrate a camera from uncalibrated images.

We present a novel calibration method, which is able to gather intrinsic and extrinsic camera parameters from 2D images using only the a-priori assumption that one or more pedestrians are walking on or near a zebra-crossing. There are no restrictions to the humans walking in a certain manner, direction, or certain velocity. The approach is also able to classify between pedestrians and cars and is therefore of great use for surveillance applications. Figure 1 shows a sequence of one pedestrian walking on a zebra-crossing.

Vanishing points are defined as the intersection of parallel lines, projected onto the image plane. When the parallel lines are also equally spaced, the vanishing line can be extracted. These two concepts are very useful for self-calibrating a camera. The edges of a zebra-crossing, which consists of alternating black and white patterns, are equally spaced and parallel in 3D space. Therefore, the first horizontal vanishing point \( v_x \) and the vanishing line \( v_l \) are
extracted by analyzing a zebra-crossing. Since the main axes of all instances of a pedestrian’s trunk, which are observed over a sequence of frames and perpendicular to the ground plane, are parallel to each other, a vertical vanishing point \( vz \) can be determined. The second horizontal vanishing point \( vy \) can then be estimated by using the triangle spanned by three vanishing points. The image of the absolute conic can be formed using all three vanishing points. By applying the Cholesky decomposition \([14]\) the intrinsic parameters can be extracted from the image of the absolute conic. Extrinsics are estimated by determining the camera’s height from the width of the zebra-crossing’s bright area.

The paper is organized as follows: Section 2 gives an overview on related work concerning self-calibration using vanishing points. Section 3 explains the calculation of the intrinsic and extrinsic parameters for a surveillance camera using static and moving objects within a traffic scene. To show the improved performance of using a combination of static and dynamic objects, experiments using synthetic and real world data are carried out in Section 4.

2. Related Work

Gathering 3D information from 2D images has been studied extensively in the last few years. Having the 3D information can e.g. be helpful for the reconstruction of a scene, as can be seen in \([3, 11]\), for video and image metrology, as in \([5, 7, 17]\), or for classification and pose recovery of objects, as in \([2, 18, 19]\). Beardsley described the extraction of intrinsic camera parameters from three vanishing points in \([1]\). By the determination of three vanishing points within one image, the principal point and the focal length can be recovered sequentially. In \([4]\), Cipolla et al. described the camera calibration using three vanishing points of an image. Their semi-automatic self-calibration method uses building facades to determine three vanishing points. The user needs to select a set of parallel image lines in order to search for a correct vanishing point initialization. After initialization, the intrinsic parameters are recovered. The relative rotation between a camera pair is estimated using the calculated points on the plane at infinity, the translation is calculated by using further points of interest in a scene.

Calibrating a camera using a pedestrian was first introduced by Lv et al. in \([12]\). Top and bottom points are determined in the images and three vanishing points are extracted. A closed-form solution is used to obtain the intrinsic parameters afterwards. The extrinsic parameters with respect to one camera are calculated to compute the complete pose of a camera within a defined world coordinate system. Based on this knowledge, Junejo proposed a quite similar calibration approach for pedestrians walking on uneven terrains only in \([9]\). The vanishing points do not need to be orthogonal to each other and the intrinsic parameters are estimated by obtaining the infinite homography from all the extracted vanishing points. Micusik and Pajdla presented a surveillance camera calibration method based on foot and head points of pedestrians in \([13]\). By introducing the Quadratic Eigenvalue Problem, extrinsic and intrinsic parameters are extracted as well as a foot-head homology is estimated. In \([20]\), Zhang et al. presented a self-calibration method using the orientation of pedestrians and vehicles. The method extracts a vertical vanishing point from the main axis direction of their trunk, perpendicular to the ground plane. Additionally, two horizontal vanishing points get extracted by investigating all the moving cars. The described approach recovers both the extrinsic and intrinsic parameters of a surveillance camera. The accuracy of the algorithm is shown by experiments using real traffic data.

Schaffalitzky and Zisserman first introduced a direct method for automatic detection of vanishing lines from equally spaced parallel lines in \([15]\). It is shown that by using vanishing lines and points, the affine geometry of a scene can be fully recovered. Se presented a method in \([16]\), where the edges of a zebra-crossing are extracted and used for pose estimation of the line segments. This approach has been developed for helping the partially sighted.

In this paper, we propose a self-calibration procedure based on vanishing points, which eliminates two main problems in previously presented self-calibration approaches, namely

1. a restriction in terms of constraints which cannot be fulfilled precisely in surveillance scenarios (e.g. cars must be driven in a straight manner, as presented in \([20]\)) and

2. the need of a-priori information, e.g. camera’s or pedestrian’s height like in \([12]\) or \([20]\).

We therefore exploit the approach of \([16]\) and determine both a horizontal vanishing point and a vanishing line of the edges of a zebra-crossing. By having a person walking through the working volume and near or on a zebra-crossing but without any restriction in terms of a certain manner, velocity, or direction, all three vanishing points get extracted from a scene and the camera can be calibrated.


Most of the surveillance scenarios (garages, parking lots, urban traffic scenes) basically provide two types of moving objects, namely pedestrians and vehicles. When pedestrians are observed it is often the case that they are crossing the street on or near a zebra-crossing. Two constraints need to be fulfilled in order to extract vanishing points from surveillance scenarios using our algorithm.
• Pedestrians are walking on an even ground plane with their body perpendicular to the plane.
• Zebra-crossings having equally spaced and parallel black (or darker) and white (or brighter) patterns are present somewhere in the observed scene.

As the mentioned properties arise in many traffic surveillance scenarios, they can easily be fulfilled and used for gathering three vanishing points, perpendicular to each other and denoted by \( vz = (vx, vy) \), \( vy = (vyx, vyy) \), \( vz = (vzx, vzy) \).

### 3.1. Moving Object Detection and Classification

As we use static surveillance cameras, simple background subtraction is taken for extracting the moving blobs within a scene. The blob’s shadow is removed by using the normalized RGB colour model. Since blobs can be separated (e.g. between top and bottom of pedestrians), a morphological closing is performed for reunion. The center of mass is calculated for each line and for each row of the moving blob. Afterwards, two lines called lines of mass are fitted through all centers in horizontal and vertical direction. Figure 2 shows the vertical (\( mv \)) and horizontal (\( mh \)) lines of mass of a walking person and a car. When observing a pedestrian, the difference between horizontal and vertical line of mass is smaller than the difference between the two lines of an observed car. This difference occurs due to a more cubic and regular shape of a car compared to the pedestrian. We therefore introduce a threshold to classify between a car and a pedestrian. In practice, this threshold is set to 45°. Figure 2 shows the vertical and horizontal line of mass and the classification between pedestrians and cars. As can be seen, the segmentation must not be precise (e.g. holes in the body of the pedestrian, clipped head) to get a correct classification result and the two lines of mass.

### 3.2. Extraction of the Vertical Vanishing Point

When using a sequence of two or more instances of one or more walking pedestrians, indicated by \( n = 1, 2, \ldots, N \), we have multiple vertical lines of mass, denoted by \( mv_n \). Next to calculating these lines for each blob classified as a pedestrian, we can determine the vertical vanishing point \( vz \). When using synthetic data, all intersections of possible pairs of \( mv_n \) must converge to one point \( vz \). Since the initialization of \( mv_n \) may be noisy when using real data, an approximation needs to be performed. Let \( P1(px, py) \) and \( Q1(qx, qy) \) be the foot and head point of the first instance of the pedestrian. By exploiting cross-ratio, the coordinates of the vertical vanishing point \( vz \) may be written as

\[
\frac{vy - py}{vyx - vx} = \frac{qy - py}{qx - px}
\]

or

\[
vzx(qy - py) + vy(qx - px) = (qx \cdot py - px \cdot qy)
\]

### 3.3. Extraction of the First Horizontal Vanishing Point and the Vanishing Line

After the determination of the vertical vanishing point \( vz \), the first horizontal vanishing point \( vh \) and the vanishing line \( vT \) need to be estimated. We therefore analyze a zebra-crossing, which consists of black (or darker) and white (or brighter) equally spaced patterns. For the calculation of the vanishing line, at least three equally spaced parallel lines need to be extracted. In a first step, a Canny Edge Detector is used to determine the edges in the zebra-crossing image. Only positive gradients are taken for further calculations. When positive and negative gradients would be used, this would mean that the width of black and white patterns must be the same to have all parallel edges equally spaced. When only using positive gradients, all black or all white patterns must have the same width but not both colors.

In a second step, we eliminate redundant edge segments (e.g. one edge segment of a zebra-crossing can be separated in two line segments when a pedestrian is occluding parts of the edge). We remove the shorter segment and only leave the longer one. We then search for all edge segments,
which support a common vanishing point $vx$. By exploiting RANSAC [6], the intersection of a random pair of edge segments is taken as an initial guess for $vx$. Only those segments where $vx$ is located within a certain distance to them are taken as inliers. In practice, this threshold distance is three pixels. The result having the most inliers is taken as the best one. The final vanishing point is then estimated by only using inlier edges. For each inlier edge segment two points on the line are taken to get a least square estimation for $vx$ (as described for the estimation of $vz$ in Section 3.2).

It may occur that also segments which are not part of the zebra-crossing support a vanishing point but those lines get eliminated in the next step. The vanishing line $vl$ can only be calculated by a closed-form solution when using exactly three equally spaced and parallel edge segments. Having three different line segments $l_i$, $l_j$ and $l_k$, which may not be consecutive, this closed form solution is given by [16]

$$vl \propto [(l_i \times l_k)\cdot(l_j \times l_k)]l_j + (k-i)[(l_i \times l_j)\cdot(l_j \times l_k)]l_k.$$  

We use RANSAC [6] to estimate the vanishing line. Therefore a triplet of random edge segments are taken to calculate an initial guess for $vl$ using Equation 4. We can check if another edge segment supports the current estimation of $vl$ by calculating

$$E = l_i \times l_j + (l - i)\mu v_l \times l_i,$$

where

$$\mu = -\frac{l_i \times l_j}{v_l \times l_j} = -\frac{l_i \times l_k}{2 \cdot (v_l \times l_k)}.$$  

In practice, $vl$ supports $v_l$ when $|E|$ is smaller than a threshold, chosen to be 0.0008. Figure 3 from left to right shows all detected edge segments, the segments supporting one common vanishing point, the three edge segments used for calculating the vanishing line, the estimated $vl$ and $vx$.

3.4. Estimation of the Second Horizontal Vanishing Point

The third vanishing point, $vy$, can be calculated after the estimation of $vx$, $vz$ and $vl$. Figure 4(a) shows different instances of a pedestrian walking on a zebra crossing and the interrelationship of $vx$, $vy$, $vz$ and $vl$. If $vl$ has a gradient equal to zero, it is called the horizontal line. This gradient angle is known as the yaw angle $\gamma$ of the camera. Before calculating $vy$, the vanishing line needs to be aligned with the horizontal line. This is established by rotating $vl$ by the negative gradient of the vanishing line. By aligning $vl$, the whole scene, including all points needed for further calculations, must also be rotated $(vx = vx', vz = vz')$. The principal point $pp$ can safely be assumed to be the image center and $pp$ is rotated to obtain $pp'$. As $pp'$ is the ortho-center of the triangle spanned by the three vanishing points, $vy'$ can be calculated geometrically as shown in Figure 4(b) and described in [12]. To obtain $vy$, the scene is rotated back around the image origin by the gradient of $vl$.

3.5. Camera Calibration from Vanishing Points

Intrinsic parameters of a pinhole camera determine the relationship between image plane and camera reference frame. The camera calibration matrix $K$, which describes the interior orientation of the camera is defined by

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}.$$  

The matrix is composed of the intrinsic camera parameters focal length ($f$) and the coordinates of the principal point $pp = (u_0, v_0)$. The interrelationship between camera frame
and world coordinate system is described by the extrinsic parameters which consist of a rotation matrix $R$ and a translation vector $t$. The rotation can be set up by combining three metrics, namely the pan angle $\alpha$, the tilt angle $\beta$ and the yaw angle $\gamma$. The translation is described by a three dimensional direction vector. Together with the camera intrinsic parameters they set up the camera matrix dimension.

After the calculation of all three vanishing points, the image is invariant to any applied rotation or translation. Under the assumption of squared pixels and zero skew, $\omega$ has the form of

$$\omega = \begin{bmatrix} \omega_1 & 0 & \omega_2 \\ 0 & \omega_1 & \omega_3 \\ \omega_2 & \omega_3 & \omega_4 \end{bmatrix}$$

The intrinsic parameters are directly related to $\omega$ by

$$(KK^T)^{-1} = \omega$$

By applying the Cholesky decomposition [14], the intrinsic parameters are extracted from the image of the absolute conic.

Next to calculating the intrinsic parameters, we need to determine the extrinsics. Under the assumption of zero skew and an aspect ratio of 1, $KK^T = \omega^{-1}$ can be rewritten as

$$\begin{bmatrix} \lambda_1 vx_x & \lambda_2 vy_x & \lambda_3 vz_x \\ \lambda_1 vx_y & \lambda_2 vy_y & \lambda_3 vz_y \\ \lambda_1 & \lambda_2 & \lambda_3 \end{bmatrix} = KR$$

where $\lambda_i$ are scaling factors. By rearranging Equation 10, these scaling factors are determined by solving the equation system

$$vx_x^2 \lambda_1^2 + vy_x^2 \lambda_2^2 + vz_x^2 \lambda_3^2 = f^2 v_0^2$$

$$vx_y^2 \lambda_1^2 + vy_y^2 \lambda_2^2 + vz_y^2 \lambda_3^2 = f^2 v_0^2$$

$$\lambda_1^2 + \lambda_2^2 + \lambda_3^2 = 1$$

Three angles $\alpha$, $\beta$ and $\gamma$ are then extracted from $R$.

The position and scaling within a common world coordinate system is described by a translation $t = R(0 \ 0 \ H_c)^T$, where $H_c$ is the camera’s height. To get the height, a scaling needs to be determined. Brighter areas of a zebra-crossing do not have to be standardized but in practice, most of them are at least 2 meters long and exactly 50 centimeters wide. We exploit this knowledge and extract the brighter areas, where at least two border pixels of the area are located within a distance of two pixels to the extracted zebra-crossing edges, which are used to gather $v_i$. We also extract the centroid of the areas. Next, for each extracted pattern we find two intersections of the area’s border lines and the line, which has the vanishing point’s orientation and goes through the centroid. These intersection pairs are taken as reference points where the real world distance of each pair must be 50 centimeters. Figure 5 shows the determination of both reference points for one brighter pattern of the zebra-crossing. Since both points are located on the same plane, the 3D coordinates can be recovered from a single camera. As we know $K$ and $R$, we increase the camera’s height and project the pair of points to 3D space until the estimated distance between the two points meets the reference distance of 50 centimeters. To gain higher accuracy, the mean distance of intersection pairs of all detected bright patterns is taken.

4. Experiments

To show the practicability and the accuracy of our approach, this section provides experiments using synthetic and real data in various scenarios.

4.1. Synthetic Data

As in practice the calibration parameters are often affected by noisy input data (e.g. poor segmentation, distortions, changing lightning conditions), we try to simulate this effect by using synthetic but noisy input data. We position a synthetic camera, having a focal length of 600 pixels, providing three angles, $\alpha = -20^\circ$, $\beta = -30^\circ$ and $\gamma = 0^\circ$ at a height of $H_c = 20$ units from the ground plane, as can be seen in Figure 7. The image provides the dimensions $640 \times 480$ so the principal point is located at $u_0 = 320$, $v_0 = 240$. In this test set, five pedestrians and three parallel and equally spaced lines, representing three equally spaced edges of a zebra-crossing, are used. All instances of the pedestrian and the edges of the zebra-crossing are positioned randomly within the scene. The distance between two edges is 15 units. Nevertheless, all objects are placed on the ground plane. By introducing Gaussian noise, described by its parameter $\sigma$, the pixel locations are randomly moved.
in both vertical and horizontal direction. As we want to answer the question whether the edges of the zebra-crossing or the foot/head locations of the pedestrians are less robust against outliers, we distort those two classes of points first separately and then combine the two. Figure 6 shows the results for $f$, $u_0$, $v_0$, $\alpha$, $\beta$, and $\gamma$ from left to right and top to bottom. The metrics are represented by the vertical axis. The horizontal axis represents the standard deviation $\sigma$, which goes from 0-4 pixels by a stepsize of 0.2. The calibration is performed 1000 times at each noise level and the mean results are taken for comparison. The distorted foot/head points, distorted zebra-crossing edges and both locations distorted are represented by $\times$, $\circ$ and $\Diamond$, respectively.

As can be seen, the output is not very sensitive to pedestrian locations. At a maximum noise level of $\sigma = 4$, the relative errors of $f$, $u_0$, $v_0$, $\alpha$, $\beta$, and $\gamma$ compared to the ground truth are 1.37\%, 0.04\%, 0.01\%, 0.25\°, 0.51\° and 0.00\°, respectively. In practice, the sensitivity regarding the zebra-crossing depends on the distance between the black and white patterns in the image space. The wider the distance is, the more robust the calibration setup is. Therefore, the calibration procedure is more robust when the camera’s height is greater than the distance to the first pattern of the zebra-crossing. When both top/bottom locations of the pedestrians and the edges of a zebra-crossing are distorted, we get a maximum relative deviation to the ground truth data of 1.10\%, 14.20\%, 2.82\%, 1.00\°, 0.23\° and 1.68\° for $f$, $u_0$, $v_0$, $\alpha$, $\beta$, and $\gamma$, respectively. As expected, the curves of all metrics show that the accuracy decreases by increasing the noise level. As the camera’s height has a maximum relative deviation of $0.02\%$ when both zebra edges and pedestrian locations are distorted, $H_c$ and $t$ are not taken into account for evaluation.

4.2. Real Traffic Scene

As we also want to show the practicability of our approach, we test the algorithm on real world data. For this purpose, we use a Canon Digital IXUS 60 camera for capturing two image sequences with a resolution of 640x480 pixels and a focal length of 665 pixels, taken from the EXIF tags. A pedestrian is walking near or on a zebra-crossing. Four sequences, labeled PED01-PED04 and shown in Figure 8, are taken for evaluation purposes. The three thick lines represent the pencil of zebra-crossing edges used for calculating the vanishing line. PED01 and PED02 show the same zebra-crossing, PED03 and PED04 offer a different one and 10, 11, 8, 11 instances of a pedestrian, respectively. Due to privacy issues, pedestrians are shown as white silhouettes. The camera is located at a measured height of $H_c = 164/166/152/153$ centimeters (cm) above the ground plane for PED01-PED04, respectively.
Table 1. Comparison of calculated intrinsic/extrinsic parameters using PED01-PED04 to the results obtained by using the CCTfM and the method of Lv et al. (using sequence PED01). The ground truth focal length is 665 pixels, the measured camera heights are 164/166/152/153 centimeters for PED01-PED04, respectively.

<table>
<thead>
<tr>
<th></th>
<th>f (pixel)</th>
<th>u₀ (pixel)</th>
<th>v₀ (pixel)</th>
<th>(α, β, γ)</th>
<th>t</th>
<th>Hx (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CCTfM</td>
<td>662.1</td>
<td>318.6</td>
<td>239.3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lv et al.</td>
<td>678.8</td>
<td>193.8</td>
<td>220.0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PED01</td>
<td>665.3</td>
<td>317.6</td>
<td>240.5</td>
<td>-45.4</td>
<td>-0.979</td>
<td>159.3</td>
</tr>
<tr>
<td>PED02</td>
<td>625.4</td>
<td>327.7</td>
<td>238.5</td>
<td>-43.3</td>
<td>-0.79</td>
<td>158.6</td>
</tr>
<tr>
<td>PED03</td>
<td>656.7</td>
<td>314.8</td>
<td>239.7</td>
<td>33.8</td>
<td>17.5</td>
<td>156.4</td>
</tr>
<tr>
<td>PED04</td>
<td>675.7</td>
<td>356.6</td>
<td>245.9</td>
<td>43.8</td>
<td>0.038</td>
<td>161.9</td>
</tr>
</tbody>
</table>

To evaluate the accuracy of the intrinsic parameters, the results are compared to the results obtained by using the Camera Calibration Toolbox for Matlab (CCTfM1) and an implementation based on [12], where PED01 was used as input. Table 1 presents the intrinsic parameters for the CCTfM, Lv et al. as well as estimations of intrinsics, extrinsics and the camera height for PED01-PED04 using our approach. As can be seen, all obtained results are very close to the ground truth data but the results for the principal point using our approach are closer than the results obtained by Lv et al. This occurs due to segmentation uncertainties which have more effect on the method of Lv et al. than on our proposed one. We learned from our experimental results that the focal length is more sensitive to outliers than the principal point, which occurs due to poor pedestrian segmentation (e.g. clipped head or legs) and the resulting calculation errors for each pixel. As can be seen in Figure 8, PED03 provides the highest yaw angle of γ = 6° and a smaller tilt angle than PED04.

In a second step, real world measurements are compared to the results obtained by our approach to evaluate the estimated extrinsic parameters. The calibration parameters of PED01 are used for this purpose and we determine two types of measurements. First, vertical ones (labeled h1-h7), where bottom points are located on the ground plane, having the 3D coordinates (x₁, y₁, 0) and top points located at (x₁, y₁, z₂) are measured. Second, we calculate horizontal distances (labeled d1-d13) where both start and end points are positioned on the ground plane and therefore share the same z-coordinate. With this restrictions it is possible to recover the 3D point from a single projection matrix, as described in [17]. The uncertainty analysis does take into account image coordinates, distorted by a Gaussian noise of σ = 5 in both vertical and horizontal direction. All metrics are presented in centimeters. As can be seen, the mean error for all distances between measured and calculated values is 5.4%, the standard deviation is 3.7%. Although Zhang et al. only used distances shorter than 2.3 meters and located near the camera, we reach slightly better results compared to the measurements introduced in [20] (mean error 6.3%, standard deviation 4.6%). Using longer distances results in higher relative error rates because due to measurement uncertainties far distances have a higher error weighting.
than near ones. This phenomenon is also visible in Table 2, where longer distances or distances far away from the camera have a higher uncertainty than shorter or near measurements.

5. Conclusion

Traffic scenarios often include zebra-crossings and pedestrians. We propose a novel self-calibration method for fixed surveillance cameras based on these two objects. An important advantage of our algorithm is that it needs no user input to completely recover both intrinsic and extrinsic parameters. This is necessary for camera calibration and re-calibration where physical access is impossible. One vertical vanishing point is extracted from the pedestrians’ lines of mass and in combination with the horizontal vanishing point and the vanishing line, obtained from three equally spaced zebra-crossing edges, the camera can be calibrated. The suggested system requires zebra-crossings to work, which limits the number of scenarios. Nevertheless, it is very useful for traffic surveillance applications, where 3D information is needed to improve e.g. pedestrian detection or tracking, pose estimation or object classification. The proposed system is both robust and accurate, which is demonstrated by experimenting on synthetic and real world data and by comparing it to state of the art calibration methods.

References