On the use of calibration for pedestrian detection in on-board vehicular cameras

Gustavo Führ, Claudio R. Jung
Institute of Informatics
Federal University of Rio Grande do Sul

Mauricio B. de Paula
Mathematics and Statistics Department
Federal University of Pelotas

***Abstract*—This paper presents a new approach for pedestrian detection in the context of Driver Assistance Systems (DAS). Given a camera with known intrinsic parameters, a flexible online calibration scheme that explores the expected road geometry is used to obtain the extrinsic parameters. With the full camera parameters, the expected geometry and size of a standing person is used to customize a baseline pedestrian detector based on sliding windows and multiple scales. Our experimental results show that the proposed approach allows the use of detachable cameras in the context of DAS, improving the accuracy of the baseline pedestrian detector. Furthermore, the flexible calibration scheme allows to estimate the distance from detected pedestrians to the camera using detachable cameras, opposed to the fixed onboard cameras in commercial vehicles that support vision-based DAS.**

***Keywords*-pedestrian tracking; pedestrian detection; onboard vehicular cameras; camera calibration;**

I. Introduction

Road traffic accidents are a significant cause of death around the world. There are over 1.2 million of lives lost per year related to traffic accidents, which also cause 20 to 50 million non-fatal injuries [1]. The most fragile element in traffic scenes is the pedestrian. Even at relatively low speeds, crashes involving pedestrians are potentially lethal. For example, Davis [2] modeled the relationship between the risk of pedestrian fatality P and the impact speed s (in km/h) for different age groups, and for the elderly (60+) he obtained

$$P = 1 - \frac{e^{9.87-0.20s}}{1+e^{9.73-0.20s}}, \tag{1}$$

so that the fatality rate is 90% for an impact speed of 60km/h, which is the urban speed limit in some countries, as Brazil.

In the past years, a great amount of money has been invested by both governments and the automotive industry to increase road safety by providing automobiles roads with some kind of intelligence. Many commercial vehicles are now equipped with Driver Assistance Systems (DAS), which inform the user about possibly dangerous situations on the road or even taking action to prevent an accident. In particular, pedestrian detection modules are responsible for identifying pedestrians using onboard cameras (possibly with the support of other more expensive sensors, such as LIDAR), so that the driver could be alerted (or the vehicle could break) when collisions with pedestrians are about to happen.

Such technologies are already present in commercial vehicles, and some car manufacturers, such as Lexus and Toyota, intend to add pedestrian detection as a standard feature in almost all vehicles in the United States by the end of 2017[1]. However, vehicles in developing countries (where the number and fatality of traffic accidents is higher) are far less technological.

This paper presents a simple and cheap pedestrian detection scheme using detachable onboard cameras (such as smartphones). Given an offline calibration scheme for the intrinsic parameters (which must be done only once for fixed focal lens cameras), it allows a flexible installation setup in the interior of the vehicle by performing an online calibration scheme for the extrinsic parameters [3]. Given the full camera matrix, it explores the expected height and vertical stance of walking pedestrians to re-weight the output of baseline vision-based pedestrian detection schemes using geometrical priors, as in [4]. To reduce the computational burden of pedestrian detection, temporal constraints based on the known speed of the vehicle can be used to estimate pedestrian locations, so that the detection process itself is not performed at every frame. Finally, the flexible calibration scheme also allows the estimation of the distance from the detected pedestrians to the camera.

The remaining of this paper is organized as follows. Section II presents a revision of related works about pedestrian detection, focusing on DAS. Section III describes the proposed method in details, and the results are presented and discussed in Section IV. Finally, the conclusions and directions for future work are provided in the last section.

II. Related Work

There has been an increase in vision-based pedestrian detection in the past years, focusing on still images and video sequences [5]. In the context of DAS, there are also requirements of real-time execution and robustness [6], which makes the problem even more complex. There are several existing methods for pedestrian detection, as noted by recent survey papers [7], [5]. This section will briefly revise some classical pedestrian detection methods, and focus on approaches tailored for DAS.

[1] http://corporatenews.pressroom.toyota.com/releases/lexus+toyota+automated+braking+standard+2017.htm

A ground breaking work based on Haar-like features was introduced by Viola and Jones [8]. Such features consist of sums of pixels within rectangular regions, which can be done at constant time regardless of the window size if integral images are used, and were combined with motion cues in the context of pedestrian detection [9]. Despite the computational advantage, the use of such simple features limited the accuracy for complex problems such as pedestrian detection.

One very popular feature used in pedestrian detection is based on Histograms of Oriented Gradients (HOGs). In [10], Dalal and Triggs used HOGs to encode the characteristics of standing people, and used a linear Support Vector Machine (SVM) in the classification step. To cope with scale and translation, sliding windows and multiresolution features, computed at pyramids of images, were used. Schwartz and colleagues [11] combined HOG with several other descriptors, and used Partial Least Squares (PLS) to cope with high-dimensionality features. Felzenszwalb et al. [12] proposed a Deformable Parts Model (DPM) detector assuming that an object is constructed by its parts using the histogram of oriented gradient (HOG) to extract the characteristics of the object and a latent SVM classifier.

Aiming to reduce the cost of building the multiscale features in HOG, Dóllar and colleagues presents a hybrid approach in which features are computed in sparse pyramid of images, and interpolated in-between. Their method presented a good trade-off between the speed of the scale invariant features in [8] and the flexibility of gradient-based information of [10], and inspired several more recent generic-purposed pedestrian detection schemes [13], [14], [15] and also in the context of DAS [16]. Also, a recent trend for generic purpose pedestrian detection is the use of deep learning methods, as in [17], [18].

Differently from other approaches for object recognition that use local features to classify an object, Torralba et al. [19] use a context-based vision system for place and object recognition. Global image features are computed to predict the scene, which are used as priors for the local object detection and recognition. The authors use GIST descriptors to develop a low dimensional representation of the scene, which feed a dynamic Bayesian Network/HMM [19], [20]. Premebida and Nunes [21] proposed another context-based system based on multiples sensors, composed by a LIDAR module acting as the first stage of object detection, a module that informs the system with contextual information from a semantic map of the roads, and an image-based detector (based on a HOG+SVM classifier) that uses sliding windows with the role of validating the pedestrian in ROIs generated by the LIDAR. Their method also uses a Bayesian approach on the mediation between the local (from LIDAR and image modules) and global (from maps of the roads) information. Kooij et al. [22] presented a context-based model based on HOG+SVM classifier combined with a dynamic Bayesian network for pedestrian path prediction in the intelligent vehicle domain.

In the context of DAS, additional modalities (such as infrared, laser scanners or stereo cameras) are usually explored to improve robustness [23]. However, since the scope of this paper is to deal with low-cost solutions using a detachable camera, only methods that explore monocular cameras will be analyzed.

In DAS, the onboard camera is typically installed in the central portion of the windshield, approximately aligned with the central axis of the vehicle. The upper portion of the images is usually related to the sky, which could be used to reduce the search space. Brehar and Nedevschi [24] explored a pinhole camera model to reduce the search space of generic pedestrian detection methods based on sliding methods, so that windows closer to the camera are larger. Prioletti and his group [25] also explored world-camera relationship for pedestrian detection, using the inverse perspective mapping to remove detections yielding implausible pedestrian heights. In fact, the use of camera knowledge was previously presented in [26], where the authors explored a simplified camera model (knowledge of the horizon line) and local object geometry to improve the performance of object detectors, in particular pedestrians.

Still in the context of DAS, traditional pedestrian detection schemes and adaptations have been used. Chen et al. [27] explored Haar-like features with motion information, which are fed to an SVM for the classification step. Ohn-Bar and Trivedi [28] studied the effect of including appearance patterns (orientation, occlusions and visual cues) in the context of pedestrian detection, concluding that these patterns can boost detection rates. Zhang and collaborators [16] used a Haar-like template pool, describing the body as parts (head, upper body and lower body) that appear in a given order for standing pedestrians.

For DAS, one class of approaches focuses on adapting traditional pedestrian detection schemes, while others [24], [25] somehow explore camera information to either reduce the ROI or remove implausible detections. However, the world-camera relationship is either obtained for specific datasets, as in [24], or does not tackle camera calibration schemes, which limits the application for fixed-camera setups [25]. In this work, we also explore the camera-world relationship for pedestrian detection. However, we use a camera calibration technique that allows the use of detachable cameras [3], since extrinsic parameters are obtained on-the-fly exploring geometric information from the road. Also, the calibrated camera is not used only to reduce the search space or to remove candidates: it serves as a geometric prior that can be plugged into any baseline pedestrian detector.

III. The Proposed Method

The main goal of this paper is to propose a flexible approach for pedestrian detection in the context of DAS. It allows the use of any conventional monocular camera loosely mounted in the interior of the windshield.

A. The camera setup

Let us consider a monocular camera installed in the interior of a vehicle, on the central portion of the windshield. We define the world coordinate system (WCS) such that the central point of the camera is located at point $(0, h, 0)$ in the WCS, as

illustrated in Fig. 1, where α (pitch), β (yaw) and γ (roll) are the Euler angles related to the x, y and z axis, respectively.

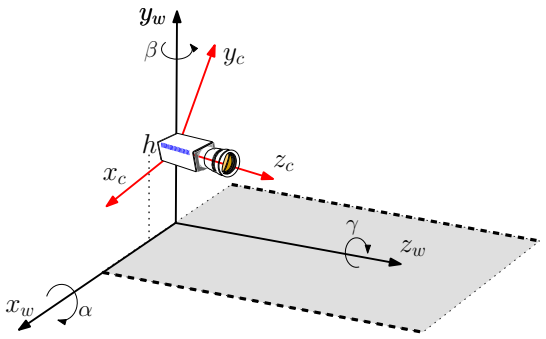


Fig. 1. 3D world and camera coordinate system.

As in [3], we also consider that there is no roll (since such rotation is typically prevented by the windshield), and assume a pinhole camera model. Then, the calibration procedure explores the expected lane geometry within a straight portion of the road, allowing to compute the rotation angles α and β, as well as the camera height h. In such model, given a point $\boldsymbol{x}_w = (x_w, y_w, z_w)^T$ in the world coordinate system, the corresponding point $\boldsymbol{x}_c = (x_c, y_c, z_c)^T$ in the 3D camera coordinate system is given by a rigid transformation:

$$\boldsymbol{x}_c = R(\boldsymbol{x}_w - \boldsymbol{x}_0), \tag{2}$$

where $\boldsymbol{x}_0$ is the position of the camera system, and

$$R = \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ -\sin\alpha\sin\beta & \cos\alpha & -\cos\beta\sin\alpha \\ \cos\alpha\sin\beta & \sin\alpha & \cos\alpha\cos\beta \end{bmatrix}. \tag{3}$$

is the rotation matrix (with no roll). The corresponding projection of $\boldsymbol{x}_c$ into an image pixel $\boldsymbol{u} = (u, v)^T$ is given by

$$u = \frac{f s_u x_c}{z_c}, \quad v = \frac{f s_v y_c}{z_c}, \tag{4}$$

where f is the focal length of the camera, s_u and s_v relate to the pixel dimensions, and $(u, v)^T$ are image coordinates relative to the optical axis $(u_o, v_o)^T$ of the camera. For camera with fixed focal length lenses, fs_u, fs_v and $(u_o, v_o)^T$ are constant and can be obtained by performing an offline calibration procedure a single time. There are many available techniques for obtaining the intrinsic parameters of a camera, and we chose the Matlab Camera Calibration Toolbox [29] due to its simplicity and popularity.

The extrinsic parameters α, β and h depend on the camera placement in the vehicle, and can be estimated on-the-fly based on the geometry of a linear portion of the road with dashed lane markings, as presented in [3] and revised next.

B. Camera calibration

Typical highways contain several straight portions of the road, and the driver is expected to drive approximately in the middle of the road, parallel to the lane boundaries. Moreover, markings that separate adjacent lanes are usually dashed (except when overtaking or lane changes are prohibited). The rectangular geometry of the road in straight portions, as well as the separation between lane markings, are explored for camera calibration. Additionally, an approach that explores planar motion is also presented to estimate the camera height.

1) Obtaining the pitch and yaw: Assuming that the vehicle is moving parallel to the lane boundaries, the equations for left and right lane boundaries in world coordinates are $x_w = x_0$ and $x_w = x_0 + W$, where x_0 is the orthogonal distance from the left lane to the camera, and W is the lane width, as shown in Fig. 2. These parallel lane boundaries in world coordinates are mapped into intersecting lines in camera coordinates, and the point of intersection $\boldsymbol{u} = (u_u, u_v)^T$ is a vanishing point of the scene. In fact, applying Equations (2) and (4) to $x_w = x_0$ and $x_w = x_0 + W$ leads to the parametric equations of the lane boundaries in image coordinates, as a function of z. Computing the limit as $z \to \infty$ for either of the projected lane boundaries leads to the coordinates of the desired vanishing point:

$$\boldsymbol{u} = \left(-f_u \tan\alpha, -f_v \frac{\tan\beta}{\cos\alpha} \right)^T, \tag{5}$$

where $f_u = fs_u$ and $f_v = fs_v$ are intrinsic parameters of the camera.

If $\boldsymbol{u} = (u_u, u_v)^T$ is known, both α and β can be easily obtained through

$$\alpha = -\tan^{-1}\left(\frac{v_u}{f_u}\right), \tag{6}$$

$$\beta = -\tan^{-1}\left(\frac{v_v}{f_v}\cos\alpha\right). \tag{7}$$

Here, a key issue is how to obtain the intersection point $\boldsymbol{u}$ of the linear lane boundaries automatically and in a robust manner. There are several existing approaches for lane boundary detection, and an adequate choice for our purpose is the linear-parabolic model used in [30]. In such model, the lane boundaries are modeled as a linear function in the near field, so that the vanishing point $\boldsymbol{u}$ can be computed directly from the intersection of the linear portion of the right and left lane boundaries as presented by the mathematical model in Equation (8).

$$f^k(v) = \begin{cases} a^k + b^k(v - v_m), & \text{if } v > v_m \\ a^k + b^k(v - v_m) + c^k(v - v_m)^2, & \text{if } v \leq v_m \end{cases}, \tag{8}$$

where $k \in \{r, l\}$ denotes which lane boundary we are referring to (right or left), v_m defines the boundary in the image between the near and far fields, and v is the vertical pixel component.

2) Obtaining the camera height: To obtain the camera height we explored the motion of planar points with known motion pattern. Let us consider a 3D point $\mathbf{W_1} = (x_1, 0, z_1)^T$ on the ground plane. Based on the known rotation matrix R, $\mathbf{W_1}$ is projected onto image point $\mathbf{I_1} = (u_1, v_1)^T$, and solving

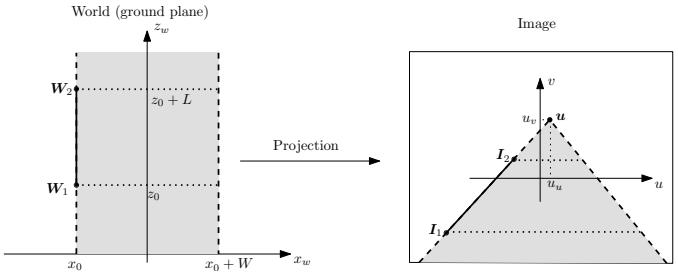


Fig. 2. Left: road in world coordinates. Right: road in image coordinates.

for x_1, z_1, leads to

$$\begin{pmatrix} x_1 \\ z_1 \end{pmatrix} = \begin{pmatrix} -\frac{f_u h v_1 \cos^2\beta + f_u h v_1 \sin^2\beta}{f_u f_v \sin\beta + f_v u_1 \cos\beta} \\ -\frac{h u_1 \sin\beta - f_u h \cos\beta}{u_1 \cos\beta + f_u \sin\beta} \end{pmatrix} \tag{9}$$

Assuming that the vehicle is moving along the central axis of the road with known speed v, the position of $\mathbf{W_1}$ in the subsequent frame will be $\mathbf{W_2} = (x_1, 0, z_1 - d_z)^T$, where $d_z = v/\text{FPS}$ is the displacement along the z axis and FPS is the frame rate (frames per second) of the video sequence. Subtracting d_z from z_1 in Eq. (9) and computing the direct mapping leads to the corresponding image coordinates $\mathbf{I_2} = (u_2, v_2)^T$ expressed by

$$\begin{pmatrix} u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} \frac{f_u(h\cos\beta - \Phi\sin\beta)}{h\sin\beta + \Phi\cos\beta} \\ \frac{f_v\left(f_u h v_1 \cos^2\beta + f_u h v_1 \sin^2\beta\right)}{(f_u f_v \sin\beta + f_v u_1 \cos\beta)(h\sin\beta + \Phi\cos\beta)} \end{pmatrix} \tag{10}$$

where $\Phi = d_z - \frac{h u_1 \sin\beta - f_u h \cos\beta}{u_1 \cos\beta + f_u \sin\beta}$.

To find a pair of correspondence points $\mathbf{I_1}$ and $\mathbf{I_2}$ in adjacent frames, we select region of interest around the two detected lane boundaries, apply the classic Harris detector [31] to find points with high curvature (such as the corners of dashed lane markers), and then track them using the pyramidal Lucas-Kanade (LK) algorithm [32], [33]. Given such correspondence pair, each coordinate of Eq. (10) can be used to obtain the camera height. More precisely, they are given by

$$h_u = \frac{d_z f_u^2 \sin^2\beta + d_z u_1 u_2 \cos^2\beta +}{f_u u_1 \cos^2\beta - f_u u_2 \cos^2\beta +} + \cdots$$
$$\cdots + \frac{d_z f_u u_1 \cos\beta \sin\beta + d_z f_u u_2 \cos\beta \sin\beta}{f_u u_1 \sin^2\beta - f_u u_2 \sin^2\beta}, \tag{11}$$

$$h_v = \frac{d_z u_1 v_2 \cos^2\beta +}{f_u v_1 \cos^2\beta - f_u v_2 \cos^2\beta +} + \cdots$$
$$\cdots + \frac{d_z f_u v_2 \sin\beta \cos\beta}{f_u v_1 \sin^2\beta - f_u v_2 \sin^2\beta}. \tag{12}$$

In a given calibration frame, we can obtain a set of n correspondence points between frames t and $t+1$, leading to n estimates $h_u^{(j)}(t)$ and $h_v^{(j)}(t)$, for $j = 1, ..., n$. Although they should be redundant, errors in R, in finding correspondence points or by chosen points that are not on the ground plane (e.g. on other vehicles) generate discrepancies. We then find a single estimate for $h_u(t)$ and $h_v(t)$ at frame t given by

$$h_u(t) = \underset{j}{\mu_\kappa}\left(h_u^{(j)}(t)\right), \tag{13}$$

$$h_v(t) = \underset{j}{\mu_\kappa}\left(h_v^{(j)}(t)\right), \tag{14}$$

where $\underset{j}{\mu_\kappa}(\cdot)$ is a $\kappa-$trimmed mean [34] in variable j of a sequence of values, which provides a robust estimate of the mean.

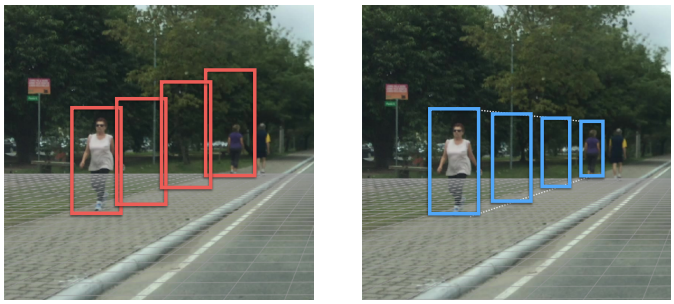
To obtain the final estimate of the height, we consider the values $h_u(t)$ and $h_v(t)$ at a given frame t, for $t = 1, ..., N_f$, where N_f is the number of frames used in the analysis. Therefore, the final estimated height $\bar{h}$ is given by

$$\bar{h} = \frac{\hat{h}_u + \hat{h}_v}{2}, \tag{15}$$

where $\hat{h}_u = \underset{t}{\mu_\kappa}(h_u(t))$ and $\hat{h}_v = \underset{t}{\mu_\kappa}(h_v(t))$ are the robust mean estimates in time of $h_u(t)$ and $h_v(t)$, respectively.

C. Context-aware pedestrian detection

Traditional pedestrian detectors (and object detectors in a broader sense) employ the detection filter at different scales to generate suitable candidates for detection. Such multi-resolution approaches can be implemented using a pyramid of images (in which the original image is re-scaled), a pyramid of classifiers (in which only the original image is used, and the scale of the detection window changes), or the combination of both (in which some re-scaled versions of the original image are used, and a pyramid of classifiers is used in-between) [35]. However, there is only a small range of scales that make sense to the dimensions of a real pedestrian for a given camera setup, as illustrated in Fig. 3. More precisely, Fig. 3(a) shows a fixed-size scanning window, which is plausible for the woman, but too large for the other locations. Fig. 3(b) shows geometrically-aware windows, for which the size depends on the ground-plane location.



(a) (b)

Fig. 3. Example of detection windows with (a) fixed size, which lead to implausible pedestrian heights at some points, and (b) adaptive size, depending on the ground plane location.

It is important to mention that Hoiem and colleagues [26] explored a simplified camera model (knowledge of the horizon line) and local object geometry to improve the performance of object detectors, and used generic pedestrian detection as an example. Also, Brehar and Nedevschi [24] used the fact that pedestrians closer to the camera are bigger in the ICS, while Prioletti et al. [25] used the inverse perspective mapping to remove detections yielding implausible pedestrian heights.

In this work, we rely on a better camera model (full calibration, as described in the previous section) than [26], [24], and explore geometric cues about a standing pedestrian in the detection itself, not just for validation purposes as [25].

For a given bounding box B, let E_B denote some kind of pedestrian image-based evidence computed on B (e.g. HOG or Haar-like features), and let y_B denote its height in the WCS computed using the known camera parameters (assuming that the base of the bounding box is on the ground plane). Following a Bayesian classifier, a pedestrian is detected when

$$P(ped)p(E_B, y_B|ped) > P(\neg ped)p(E_B, y_B|\neg ped), \quad (16)$$

where $p(E_B, y_B|ped)$ and $p(E_B, y_B|\neg ped)$ are the joint PDFs of E_B, y_B for the pedestrian and non-pedestrian classes, and $P(ped)$ and $P(\neg ped)$ are the corresponding *a priori* probabilities. Assuming that y_B and E_B are independent and that $p(y_B|\neg ped)$ follows a uniform distribution, inequation (16) reduces to

$$\frac{p(E_B|ped)}{p(E_B|\neg ped)} p(y_B|ped) > T, \quad (17)$$

where T is an acceptance threshold.

Considering that the score $R(E_B)$ of any "baseline" pedestrian detector can be used to approximate the likelihood ratio $p(E_B|ped)/p(E_B|\neg ped)$ (disregarding normalization issues) and that $p(y_B|ped)$ follows a normal distribution with mean y_{avg} and variance σ^2, the proposed detector is given by

$$S(B) = R(E_B) \exp\left[-\frac{(y_B - y_{avg})^2}{2\sigma^2}\right] > T_S, \quad (18)$$

where the acceptance threshold T_S is inherited from the baseline detector $R(E_B)$. Due to the fast decay of the normal distribution, just a few bounding boxes B with WCS heights in the range $[y_{avg} - k\sigma, y_{avg} + k\sigma]$ are needed in practice for each location. In our experiments, we used 5 uniformly spaced heights, with $k = 2$.

For detection methods that rely on image pyramids, a classifier is trained with a pre-defined pedestrian model size, typically a rectangular region with height v_{model}. In the classification stage, the model is kept constant, and the image is re-scaled to capture pedestrians at different scales: upsampling is required to detect pedestrians smaller than the model, and downsampling for pedestrians larger than the model. In general, just downsampling is applied, so that the smallest detectable pedestrian in the scene is roughly the height of v_{model}. Given a maximum pedestrian height $y_{max} = y_{avg} + k\sigma$ (in the WCS), our method only creates candidates in which the height of the corresponding bounding box height is larger than a fraction of the height of the model bounding box v_{model}. This fraction depends on the height range of pedestrians in WCS and its value can be employed in order to limit the number of levels of the pyramid.

The largest pedestrian in the image should dictate the smallest resolution of the image pyramid. Since the pyramid is pre-computed in some methods to speed-up the process (as in [35]), the use of a calibrated camera can also define the smallest scale of the pyramid. Given a pedestrian with size v_{ped} (in the ICS), the ideal scale s in the pyramid should satisfy $2^{-s} v_{model} = v_{ped}$ (assuming that the scale factor is 2^{-s}). Hence, we scan all image pixels related to the ground plane and compute the projection of a pedestrian with the maximum allowed size y_{max}, retrieving the height of the largest bounding box in the ICS, called v_{max}. Hence, the smallest scale in the pyramid is defined as $s = \log_2(v_{model}/v_{max})$.

Finally, our candidates are evaluated in this reduced pyramid using the test given by inequation (18). Furthermore, as usual in sliding-window techniques, the final detection is achieved after performing non-maxima suppression to the outputs of $S(B)$, and not on the scores $R(E_B)$ of the baseline detector. In general, this helps to better fit the scale of the detection window to the actual pedestrian size.

To show the potential of our method, we used as baseline method the pedestrian detector presented in [7], and our experimental results show that detection accuracy were increased using camera information. We also show results using HOG+SVM [10].

The computational time of the detector can be further reduced by using the calibration in yet another manner. Instead of running the detector at each frame, we can run the whole process only at each t_w frames. In between those frames, a static pedestrian should present a displacement along the z axis in the WCS due to the motion of the vehicle d_z within two adjacent frames, as described in Section III-B2. Considering that the vehicle velocity is normally significantly larger than the pedestrian, we assume that their relative speed is approximately d_z – i.e. we always assume static pedestrians. Hence, the proposed pedestrian detection scheme is applied at each t_w frames, and in-between the trajectory of each pedestrian (bottom-central coordinate of the bounding box) is linearly interpolating assuming a constant displacement d_z in the WCS, which is projected to the ICS using the estimated camera model. The height of the bounding box is kept constant in the WCS, so that its projection to the ICS maintains the correct perspective of the pedestrian obtained in the previous detection frame.

IV. Experimental Results

In order to evaluate the performance of the proposed detection method, we created a small dataset consisting of a video taken from an on-board camera (iPhone 5S smartphone) mounted on the dashboard of a vehicle passing by an urban area. This dataset contains 2103 high-resolution frames (1920 $\times$ 1080), and a total of 1498 pedestrians were manually annotated (bounding boxes). The video, together with its ground truth, is available publicly for future reference and benchmarks[2]. It is important to note that there are several publicly available datasets for generic-purpose pedestrian detection and even in the context of automotive applications, such as [36], [7][3]. However, information about camera parameters is missing, so that our method cannot be applied.

[2]Link not yet included due to blind review

[3]http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

The set of experiments aimed to evaluate the performance of the detection procedure that includes geometric information and a baseline pedestrian detection method. In this paper, we modify the widely cited method by Dollar and colleagues [7] to include our geometric priors based on the calibration procedure. However, it is important to emphasize that the proposed approach is compatible with any sliding-window technique.

As it is common in the literature, we use the intersection over union approach (Jaccard coefficient) to determine if the detection is valid or not – value above 0.5 as in [37]. By varying the threshold of acceptance for the detection bounding boxes, as proposed in [35], [38], we generate precision-recall curves depicted in Fig. 4. It can be observed that, for any recall value (particularly those in the range $[0.3; 0.4]$), the precision value produced by our technique is higher than the baseline.

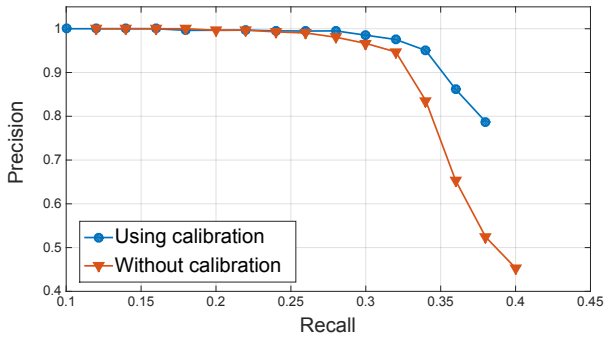


Fig. 4. Precision-recall curves for pedestrian detection proposed (using camera calibration) and the baseline method (without calibration) [36].

Additionally, we modified the traditional pedestrian detector based on HOG+SVM [10]. Our implementation uses the OpenCV standard models and the source code was made available[4]. The results of the standard sliding-window technique against our approach using calibration to generate candidates can be observed in Fig. 5. For this experiment, we made a pyramid of 10 levels using a reduction of 5% at each level. The baseline method created around 70K candidates, and ours 19K. The average running time for the unaltered version was $3.28s$ per frame, while ours was $0.3s$ per frame.

The results indicate that our approach of applying geometric information in the detector's pipeline significantly increases the overall accuracy of the system. The main reason behind this improvement is that the generation of candidates is much more coherent with the pedestrians appearing in the scene, and detection results with implausible pedestrian heights are avoided. An example of comparison between the baseline detector and the proposed approach is shown in Fig. 6, which shows the results from three different frames of our video sequence. As it can be observed, the baseline method produces detections (marked with arrows) that are clearly incompatible with real pedestrians. On the other hand, such candidates

[4] https://github.com/gustavofuhr/pedestrian_detector_calibrated.

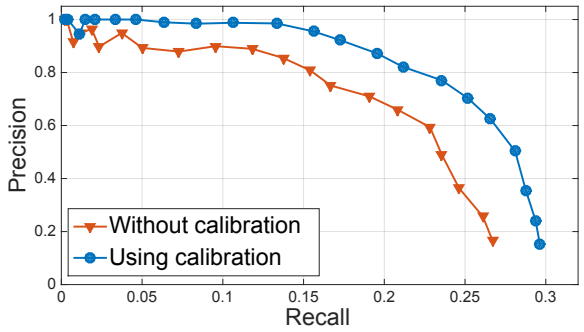


Fig. 5. Precision-recall performance on variations of the traditional HOG+SVM detector [10] with and without calibration.

would either have a very small score (due to the Gaussian weight based on pedestrian height estimates) or would not even be tested, since their heights are clearly much larger than y_{max}.

Additionally, we evaluate the number of candidates our approach creates against the baseline method. Since this values depends on the size of the input image and also the range of scales scanned in the baseline approach (which is usually set by the user), we downsampled our 1080p images using multiple scaling factors, and the results are shown in Fig. 7. Since our method samples the ground plane to generate multiple candidates and the horizon line of our video is within the image plane, such sampling could, theoretically, go on forever. However, we make a threshold based on pixels to limit the minimum height for the creation of candidates. In this experiment, we set the threshold to 10% of the height from the re-scaled image. Clearly, the number of candidates generated by our method is much inferior then a common sliding-window technique – at the original resolution, the number of candidates is reduced by a factor of 2.5.

We also tested our propagation discussed in the final part of Section III. Fig. 8 show the precision-recall curves of computing the detection by setting the temporal detection stride t_w to 5 and 10 frames, and propagating the bounding boxes in-between detections. As expected, the precision is reduced since no image feature is extracted in these frames, which can multiply the false positives of the detector (and add false negatives when new pedestrians enter the scene). However, if the window, t_w is kept small, such as $t_w = 5$, the accuracy may still be sufficient for real life applications. Moreover, since the computational time for the propagation is negligible compared from the detector, the speedup shows a factor of $t_w - 1$ in the performance. Also, for the experiments described here, the vehicle speed is constant at 40km/h for the whole sequence, and a system with automatic speedometer can be included to account for variable speeds.

Yet an additional advantage of using a flexible calibration scheme that allows the use of detachable cameras is that the distance from the detected vehicle to the pedestrian can also be estimated, as illustrated in Fig. 9.

Fig. 6. Comparison between the baseline detector (red) and the proposed improvement (blue).

V. Conclusion

This paper presented a flexible approach for pedestrian detection in the context of driver assistance systems that allows the use of detachable cameras instead of fixed vehicular onboard cameras. Initially, the intrinsic camera parameters are obtained using an offline procedure, and the extrinsic parameters that extract the rotational and translational parameters of a given camera setup are obtained online. With the full camera parameters, a baseline pedestrian detector is adapted to include geometric information about a typical standing pedestrian. An simple temporal prediction scheme can also be included, reducing the computational cost with small accuracy loss.

The experimental results showed that the proposed model is able to successfully discard detections with implausible human heights (which decrease the false positive rate), without compromising the true positive rate. Also, the use of a calibrated camera allows to precisely define a small range of detection scales for each pixel in the image, whereas typical pedestrian detectors based on sliding windows require an ad hoc definition of the global scale range. Furthermore, it is

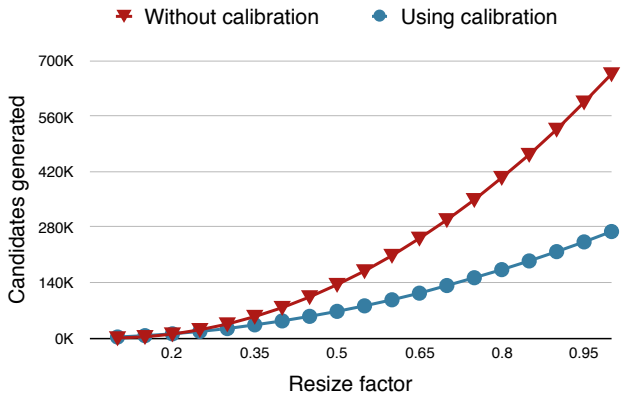


Fig. 7. Number of candidates generated by the methods as a function of image downsampling factor.

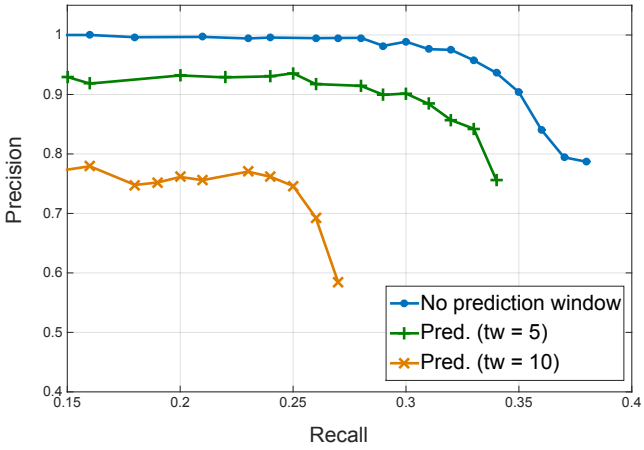


Fig. 8. Precision-recall curve using the temporal prediction scheme, in which the detection is only performed at every t_w frames.

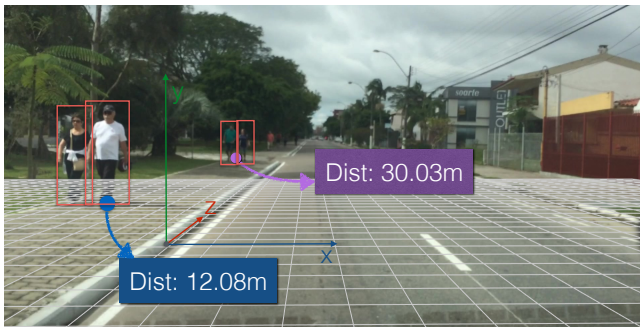


Fig. 9. Grid based on camera calibration overlaid to a frame of the video sequence, and estimated distances from pedestrians to the camera.

possible to estimate the distance from detected pedestrians to the camera, although the accuracy of these estimates must be further evaluated.

As future work, we plan to close the calibration-pedestrian detection loop, using the output of the pedestrian detector to refine the camera calibration procedure. We also plan to add more videos to the database acquired with different devices and camera placements, and to strongly explore temporal information for pedestrian detection.

ACKNOWLEDGMENT

The authors would like to thank would like to thank Brazilian agencies CNPq and Capes.

REFERENCES

[1] W. H. Organization, *Global status report on road safety 2013: supporting a decade of action*. World Health Organization, 2013. [Online]. Available: http://www.who.int/

[2] G. Davis, "Relating severity of pedestrian injury to impact speed in vehicle-pedestrian crashes: Simple threshold model," *Transportation Research Record: Journal of the Transportation Research Board*, no. 1773, pp. 108–113, 2001.

[3] M. B. De Paula, C. R. Jung, and L. G. Da Silveira, Jr., "Automatic on-the-fly extrinsic camera calibration of onboard vehicular cameras," *Expert Syst. Appl.*, vol. 41, no. 4, pp. 1997–2007, Mar. 2014. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2013.08.096

[4] G. Fuhr and C. Jung, "Camera self-calibration based on non-linear optimization and applications in surveillance systems," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2015.

[5] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognition*, vol. 51, pp. 148–175, 2016.

[6] A. Prioletti, A. Mogelmose, P. Grisleri, M. M. Trivedi, A. Broggi, and T. B. Moeslund, "Part-based pedestrian detection and feature-based tracking for driver assistance: real-time, robust algorithms, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1346–1359, 2013.

[7] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[8] P. A. Viola and M. J. Jones, "Rapid object detection using a boosted cascade of simple features," in *IEEE Conference on Computer Vision and Pattern Recognition*, Kauai, HI, December 2001, pp. 511–518.

[9] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[10] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 886–893.

[11] W. R. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis, "Human detection using partial least squares analysis," in *IEEE Conference on Computer Vision*. IEEE, 2009, pp. 24–31.

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[13] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2903–2910.

[14] C. Zhu and Y. Peng, "A boosted multi-task model for pedestrian detection with occlusion handling," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5619–5629, 2015.

[15] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li, "Robust multi-resolution pedestrian detection in traffic scenes," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3033–3040.

[16] S. Zhang, C. Bauckhage, and A. B. Cremers, "Efficient pedestrian detection via rectangular features based on a statistical shape model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 763–775, 2015.

[17] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 899–906.

[18] W. Ouyang, X. Zeng, and X. Wang, "Learning mutual visibility relationship for pedestrian detection with a deep model," *International Journal of Computer Vision*, pp. 1–14, 2016.

[19] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, Oct 2003, pp. 273–280 vol.1.

[20] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, pp. 145–175, 2001.

[21] C. Premebida and U. Nunes, "Fusing lidar, camera and semantic information: A context-based approach for pedestrian detection," *International Journal of Robotics Research*, vol. 32, no. 3, pp. 371–384, 2013.

[22] J. F. P. Kooij, N. Schneider, F. Flohr, and D. M. Gavrila, *Context-Based Pedestrian Path Prediction*. Cham: Springer International Publishing, 2014, pp. 618–633. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10599-4_40

[23] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 7, pp. 1239–1258, 2009.

[24] R. Brehar and S. Nedevschi, "Scan window based pedestrian recognition methods improvement by search space and scale reduction," in *IEEE Intelligent Vehicles Symposium Proceedings*, 2014, pp. 529–534.

[25] A. Prioletti, P. Grisleri, M. M. Trivedi, and A. Broggi, "Design and implementation of a high performance pedestrian detection," in *IEEE Intelligent Vehicles Symposium (IV)*, 2013, pp. 1398–1403.

[26] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 3–15, 2008.

[27] D. Chen, X. B. Cao, Y. W. Xu, H. Qiao, and F. Y. Wang, "A svm-based classifier with shape and motion features for a pedestrian detection system," in *IEEE Intelligent Vehicles Symposium*, 2006, pp. 331–335.

[28] E. Ohn-Bar and M. M. Trivedi, "Can appearance patterns improve pedestrian detection?" in *IEEE Intelligent Vehicles Symposium*, 2015, pp. 808–813.

[29] J. Y. Bouguet, "Camera calibration toolbox for Matlab," 2008. [Online]. Available: http://www.vision.caltech.edu/bouguetj/calib_doc/

[30] C. Jung and C. Kelber, "An improved linear-parabolic model for lane following and curve detection," in *Computer Graphics and Image Processing, 2005. SIBGRAPI 2005. 18th Brazilian Symposium on*, oct. 2005, pp. 131 – 138.

[31] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[32] J.-y. Bouguet, "Pyramidal implementation of the Lucas Kanade feature tracker," in *Intel Corporation, Microprocessor Research Labs*, 2000.

[33] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," 1981, pp. 674–679.

[34] J. Bednar and T. Watt, "Alpha-trimmed means and their relationship to median filters," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, no. 1, pp. 145–153, 1984.

[35] P. Dollár, S. Belongie, and P. Perona, "The fastest pedestrian detector in the west," in *British Machine Vision Conference*, 2010, pp. 68.1–68.11.

[36] P. Dollár, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 304–311.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[38] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *British Machine Vision Conference*, 2009, pp. 91.1–91.11.