Vision-Based Pedestrian Detection for Rear-View Cameras

Shai Silberstein, Dan Levi, Victoria Kogan and Ran Gazit

Abstract—We present a new vision-based pedestrian detection system for rear-view cameras which is robust to partial occlusions and non-upright poses. Detection is made using a single automotive rear-view fisheye lens camera. The system uses “Accelerated Feature Synthesis”, a multiple-part based detection method with state-of-the-art performance. In addition, we collected and annotated an extensive dataset of videos for this specific application which includes pedestrians in a wide range of environmental conditions. Using this dataset we demonstrate the benefits of using part-based detection for detecting people in various poses and under occlusions. We also show, using a measure developed specifically for video-based evaluation, the gain in detection accuracy compared with template-based detection.

I. INTRODUCTION

According to the National Highway Traffic Safety Administration (NHTSA) about 18,000 people a year are hurt in back-over accidents in the U.S. alone, with about 3,000 suffering “incapacitating” injuries. 44 percent of the incidents involve children under age 5. In a related study [19], NHTSA found that no efficient sensing-based solution exists for preventing many of these accidents, and that camera-based solutions are the most promising technology. In the recent years, significant progress has been made in the field of camera-based pedestrian detection (e.g. [24]). However, current solutions focus on front applications and are designed for upright, fully visible pedestrians. Such solutions do not work for pedestrians in non-upright poses, and specifically children. They have also difficulties in detecting partially occluded pedestrians, and close-by pedestrians which are also not fully visible to the camera. Solving these issues is critical for reaching systems capable of preventing back-over accidents.

Existing vision-based on-vehicle pedestrian detection systems mainly rely on so-called “template-based” appearance models. These models capture the pedestrian appearance using one rigid template matched to each image position. Template-based detection, such as the Histograms of Oriented Gradients (HoG) [5] shown in figure 1(a), can deal with small local deformations of the pedestrian shape, but cannot handle large ones caused for example by object articulations. In the recent years “part-based” appearance models such as the Deformable Part-based Model (DPM) [11] and Feature Synthesis (FS) [2] have shown capability to handle general object detection tasks and improve benchmark results [10], [7]. As illustrated in figure 1(b), part-based models capture appearance of single object parts and combine the parts using a geometric model of spatial part relations. This flexibility allows the model to inherently support large variations in object appearance caused for example by object articulation (i.e. different human poses) and partial occlusions [16]. Moreover, increasing the number of parts in the model can further improve robustness [2]. The main obstacle towards implementation of part-based detection algorithms remains their high computational cost. Most existing systems are therefore limited to only several parts [11], [21].

Pedestrian detection in the rear-view camera poses challenges stemming from two different factors: the camera characteristics and the application scenarios. Rear-view cameras will be mandated in all new vehicles in the U.S. in the next coming years. Their main purpose is driver display, and they are typically targeted to cover a wide angle, 180° Field-Of-View (FOV), at a low cost. A common solution uses a VGA resolution fisheye lens camera with substantial distortions especially in the periphery. The scenarios in which the application must operate are also different from front
applications. Backing over is done many times in parking lots, where pedestrians and in particular children may appear, in various activities and in many different postures. The proposed system addresses these challenges directly.

This paper builds on a multiple-part based detection method, the Feature Synthesis (FS) which was first put forth in [2]. The FS uses hundreds of parts in its appearance model. A real time version, the Accelerated Feature Synthesis (AFS), was presented in [17], making the use of this model practical in terms of running time. The main contribution in this paper is a robust on-vehicle system for rear-view pedestrian detection using the AFS along-side additional algorithmic blocks. The proposed system includes three additional algorithmic modules: image warping that corrects the camera distortions, geometric reasoning for cropping irrelevant image regions which improves both running time and accuracy performance, and frame-to-frame target association which enables the use of video-based evaluation. The system is a first real-time on-vehicle system making use of multiple-part based object detection. We also provide an experimental video-based evaluation of our system compared to a template-based method on a new pedestrian dataset captured using the rear-view camera. The dataset contains over 200K annotated pedestrians in various real world scenarios. We use a video-based evaluation methodology which we developed in [1] to show the performance gained by using our part-based detection method compared to a standard template based method (HoG). We also show the advantage our system has in detecting partially occluded people and people in various poses.

This paper is organized as follows. We present an overview of related works in the next section. We then describe the proposed system in Section III. In Section IV we introduce our new dataset as well as a method for evaluating detection in video data, and the experimental evaluation results. Conclusions and future work are presented last in Section IV-D.

II. RELATED WORK

Vision-based pedestrian detection systems based for Advanced Driver Assistance Systems (ADAS) have been extensively studied in the last decade [16]. Monocular vision-based systems usually include several stages: Foreground segmentation, in which candidate areas are extracted from the image; Object classification, which uses an appearance model to classify candidates; Verification in which candidates are eliminated based on information which is orthogonal to the classifier; And tracking, which exploits the continuity of the detection location over time. Different systems propose different implementations to each of these stages. This paper focuses on the object classification stage. The majority of the systems use template-based classifiers, either by matching the silhouette [3], [15] or the full appearance [20], [5]. Modern methods (e.g. [20], [5], [15]) use machine learning techniques such as Support Vector Machines (SVMs) in order to train the classifier. Several systems decompose the template into several individual parts (e.g. [22], [21]), but since the part location is fixed with respect to the template, such models are not suitable for multi-pose detection. Most related to our work is the automotive pedestrian detection system presented in [4], which uses an optimized cascade-DPM based detector [12], but is limited to using several parts only to run in real time. For a in depth survey on ADAS pedestrian detection systems we refer the reader to [16].

Part-based methods [14], [11], [2] use object parts with a deformable configuration to model objects, increasing their ability to cope with partial occlusions and large appearance variations compared with template-based methods. Furthermore, using a large number of parts learned automatically with diverse appearances improves detection accuracy [2]. Evidently, part-based methods are highly ranked on large scale benchmarks [10], [7]. Such methods, however, are limited in the number of parts modeled [11] in order to run in reasonable time, or are impractical in terms of run time [2]. Accelerating part-based detection [12], [8], [6] efforts in the past mostly focused on methods relying on a small number of parts such as the DPM [11] since computation time increases linearly with the number of parts. In our detection system we use the Accelerated Feature Synthesis (AFS) algorithm, which we presented in [17]. The AFS uses hundreds of parts in its object model, and runs in real-time, maintaining performance comparable with the original FS algorithm. Acceleration is achieved by a series of approximations as following. A “coarse-to-fine” strategy for early window elimination and location refinement is used. In each image location, only the closest parts are compared, and for each part, only locally maximal-appearance positions are used for classification. In contrast, existing part-based methods (e.g. DPM, c-DPM) consider all parts in a dense grid of positions. Finally, to find the closest parts in each image positions a novel approximate nearest neighbor (ANN), termed “KD-Ferns”, has been developed.

Several pedestrian detection datasets were collected in the past. The INRIA [5] and MIT [20] were among the first pedestrian detection datasets but are relatively of small scale. Recently a number of databases were collected specifically in driving scenarios with O(100K) pedestrians marked: the Caltech [7] and DaimlerDB [9]. These databases, however, are not directly applicable for testing rear-view camera pedestrian detection, since the rear-view camera imagery is very different in FOV and quality, and the scenarios and driving speeds are different as well. In [23] a dataset was specifically collected for evaluating rear-view pedestrian, but since it is currently not public we could not use it for our evaluation. In addition, the reported evaluation uses a cropped-window measure, which gives a poor prediction on full-frame performance [7]. In our work we collected a new large dataset specifically for the rear-view pedestrian detection application, containing more than 200,000 marked pedestrians in full length video sequences. We measure performance on this dataset using a methodology we specifically developed for evaluating video-based detection [1].
III. PEDESTRIAN DETECTION SYSTEM

The system hardware configuration is as following: a rear-view camera (NTSC, 180° FOV) is connected to an i7 laptop via a frame grabber. The video is grabbed at 30 frames per second at VGA resolution (640 × 480). The detection system running on the laptop is composed of several algorithmic blocks as described in figure 2. In Stage A, image warping, we perform lens correction and virtual forward view rendering, which brings straight upright objects in the world to appear straight and vertical in the image. Stage B consists of geometry-based image scale-pyramid creation. Stage C consists of applying the accelerated feature synthesis detection method to the image pyramid, producing the detections in the current video frame. Finally using a simple association mechanism, targets are tracked across frames and are assigned the same ID. The output in each frame is a list of targets with the following properties: \([x, y, w, h, S, ID]\) - image position, width and height, classification score and ID. We next describe each step in more details.

A. Image warping

The first stage in the system consists of warping the image such that straight upright objects in the world appear straight and vertical in the image (Block A in figure 2). The purpose of this operation is to obtain a pedestrian appearance which is invariant to the camera lens distortion, the camera tilt and to the image position of the pedestrian. The warping transformation is automatically computed using the following steps. We compute the internal and external camera parameters using the Caltech Camera Calibration Toolbox. The tilt is obtained by including several checkerboards on the ground in the calibration process and averaging the normal direction of the ground in camera reference coordinates. From this calibration we obtain two warping transformations: the radial lens distortion correction, and the Homography which transforms the image into a virtual forward-looking view. We then compute a scaling transformation that ensures that we can detect a pedestrian at minimal height \(h\) and maximal distance \(D\), given that the system requires a target height of 90 pixels. The system was tuned to \(h = 170\) cm, \(D = 7\) m. Finally, we crop the image output size to cover 180° FOV in width and almost full image extent in height. We also developed a semi-automatic tool enabling the user to see the resulting warped image, with upright objects and distance markers enabling the user to fine-tune the warping parameters. Figure 3 shows a captured image (a) and the warped image (b). In the first image of figure 7 we show the extent of the warped image in the original image marked by a blue dotted line.

B. Image Pyramid

In the second stage (stage B in figure 2), the warped image is down-sampled at multiple resolutions creating an image scale pyramid. This is the standard technique for applying a fixed size detector to detect pedestrians at multiple sizes. The detector works in 4 full octaves (\(\times0.5, \times0.25, \ldots\)) and 4 sub-octaves, 16 scales in total.

Using the known camera extrinsic parameters we use a geometrical model to crop out irrelevant regions in the image assuming a “roughly planar ground” as explained next. In each scale \(s\) of the image pyramid we use a fixed detector of size \(128 \times 64\) pixels bounding a ∼ 90 pixels height pedestrian within it. The bounding box at scale \(s\) corresponds to a larger bounding box size with height \(h_s\) pixels in the original image. We compute for a bounding box of height \(h = h_s\), what are the positions in the image in which it can appear assuming it tightly bounds a pedestrian with reasonable height located on a flat ground. These positions can be shown to lie between two horizontal rows in the image, defined by \(y_{min}, y_{max}\), which are computed as following (Note we set \((0, 0)\) as bottom left image pixel). Given a pedestrian bounding box with bottom part at pixel \(y_b\) and top part at pixel \(y_t\), we can compute its height in the real world, \(H\), as explained in figure 4. The result is a mapping \(M\) from each \(y_b,y_t\) to the respective pedestrian height \(H\). We then query \(M\) to obtain all entries \(Q\) in which \(y_t - y_b = h\) and \(H_{min} < H < H_{max}\), where \(H_{min}, H_{max}\) are the parameters for minimal and maximal pedestrian height respectively. The
valid raw range is set accordingly to performance: GlobalMax, Sigmoid, Localized, LDA and HoG component features. For example, a localized feature is computed as: \( f(I_s) = \max_{l \in L} \mathcal{G}(\alpha'(l)) \cdot \mathcal{N}(l; \mu_r, \sigma_{l_2x2}) \) where \( \mathcal{N} \) is a 2D Gaussian function of the detection location \( l \) and \( \mathcal{G} \), a learned sigmoid function on the appearance score. Such features represent location sensitive part detection, attaining a high value when both the appearance score is high and the position is close to a preferred part location \( \mu_r \), similar to parts in a star-like model [13]. Figure 1(b) shows 3 selected localized features. The final classification score is a linear combination of the feature values: \( C(I_s) = \sum_{f \in F} (W_f \cdot f(I_s)) \). We next describe briefly the AFS method for detecting objects in full images in real-time.

AFS consists of 3 stages: a coarse detection stage which outputs candidate targets and a fine detection stage which scores all windows around the candidate targets. The input is the image pyramid and the output is the object detections represented by a set of bounding boxes at multiple locations and scales in the image and their classification scores. The AFS algorithm flow (see Figure 5) is composed of a two-level coarse-to-fine cascade. The coarse level uses the sliding window methodology. It uses a trained coarse classifier \( C_1 = (F_1, R_1, W_1) \) to compute the classification score for a dense set of sub-windows sampled in scale and position space. For a specific scale, sub-windows are sampled on a regular grid with a spatial stride \( s = s_1 \) pixels. Image locations which received a large enough score are then passed to the second level. Around each such location a local region is defined and sub-windows are sampled in that region on a finer grid with stride \( s = s_2 \) and processed by the second level with classifier \( C_2 = (F_2, R_2, W_2) \) to produce the final classification score. Computing the classification score for each sampled sub-window is similar for both cascade levels. We refer to this procedure as a one-level detection (blue rectangles in Figure 5). The input to the first-level detection is the entire image pyramid and to the second level detection only the candidate image regions. Each image area is processed (either by coarse or fine detector) is in three stages, parameterized by the classifier model \( C(F, R, W) \). The first stage computes local gradient orientation histograms, then fragment similarity maps are computed and finally the classification scores. Each of the computations has been optimized for speed with minimal accuracy loss. In the fragment similarity map computation we use a novel ANN algorithm, "KD-Ferns" developed specifically for this application. For further details on the computation of each stage in the AFS please refer to [17]. The output of this stage is a list of detections or bounding boxes, where each bounding box \( A \) is defined by its position and classification score \( (S) : A = [x, y, w, h, S] \).

D. Non-Maximal Suppression

The next stage is the Non-maximal suppression (NMS) (Block D in figure 2), which removes redundant and inaccurate detections. We experimented with different variations of NMS, and found the following to maximize detection performance.
We compute a graph in which each detection is represented by a node. Let \( A, B \) be two detections, and assume w.l.o.g. that \( s(A) < s(B) \). Then an edge connects two detections \( A \) and \( B \) if one of two following conditions is met:

- \( \frac{|A \cap B|}{\sqrt{|A||B|}} > t_{NMS}^1 \)
- \( \frac{|A \cup B|}{|A|} > t_{NMS}^2 \) \& \( |A| < |B| \), \( t_{NMS}^2 > 1 - \delta \)

Where \(|\cdot|\) denotes area. In other words, either the bounding boxes sufficiently overlap, or \( A \), the lower scoring bounding box, is almost fully contained by \( B \). Once graph is created, any detection \( A \) with a graph neighbor \( B \) s.t. \( s(A) < s(B) \) is removed. In all the reported results the NMS parameters were set to: \( t_{NMS}^1 = 0.75, t_{NMS}^2 = 0.97 \).

### E. Target association

The last stage of the detection is assigning a target ID to each bounding box (Block E in figure 2). This is done for the sole purpose of enabling video-based evaluation. In the video-based evaluation, explained in detail in section IV-B, real targets and false alarms on the same object are grouped to single detections across several frames for evaluation. Consequently, to obtain a good evaluation result, the detection system is encouraged to keep the target ID consistent across video frames. Our association mechanism follows a “tracking-by-detection” approach. In the first frame we assign arbitrary unique IDs to each target. We then associate targets in the current frame to close-by targets in the previous frame as following. Given a bounding box \( B_t \) in frame \( t \), we look for the closest bounding box \( B_{t-1} \) in frame \( t-1 \), according to the mean squared distance between the upper left and lower right corners of the two bounding boxes. We then assign \( B_t \) the same ID as that of \( B_{t-1} \). In case multiple bounding boxes are assigned with the same ID, only the highest scoring one gets this ID, while the other are assigned with new IDs. Finally, we translate the bounding box to the original image by transforming the bounding box center to its coordinates in the original image and estimating the transformation of the bounding box width and height.

### IV. EXPERIMENTAL RESULTS

We test our system on a new rear-view pedestrian dataset we collected. The dataset was collected in various conditions and situations relevant to backing over scenarios and contains a significant amount of annotated pedestrians. We use a video-based evaluation [1], developed for measuring performance as perceived in a continuous visual display. We next describe the dataset, the evaluation criteria and the evaluation result.

#### A. Rear-view pedestrian detection dataset

To collect the dataset we mounted the fisheye camera previously described on a vehicle. We mounted the camera in front for safety reasons, but in a typical rear-view installation pose: 107cm height and 25° downward tilt angle. The dataset contains 15 filming sessions, each taken in a different day with different scenarios. Each session contains multiple clips with duration ranging from several seconds to several minutes. In total, the dataset contains 250 clips with a total duration of 76 minutes and over 200K annotated pedestrian bounding boxes. Each clip was recorded as a raw 30fps with simultaneous recording of the vehicle CAN information. There are two types of sessions, containing either staged or “in-the-wild” pedestrians. The staged scenarios include mainly pedestrians walking in front of the camera at different positions and directions in a controlled manner, spanning the different use cases of a rear alert or breaking automotive feature. In the remaining sessions the vehicle drove either in public roads or in parking lots and captured incidental pedestrians. The different locations include: indoor parking lots, outdoor paved/sand parking lots, city roads and private driveways. We filmed both day and night scenarios, with different weather and lighting conditions. We also filmed one session in our lab with staged people in various poses. Figure 7 illustrates the variety of scenarios captured. The entire dataset was annotated semi-manually by marking a bounding box around each pedestrian in each video frame, using the annotation tool provided by [7]. This tool allows skipping several frames, completing the intervals automatically and allowing fast corrections. We chose 70 clips as the test portion of the dataset for evaluation purposes, while the training part was kept for future system improvements. Night and lab sessions were excluded from the test set.

#### B. Evaluation criteria

For quantifying the detection accuracy of the system we developed a video-based evaluation criterion previously published in [1]. Evaluation methods in previous computer vision and automotive pedestrian detection work [10], [7] were devised to test detection performance in single images or independent movie frames. The ground truth annotations in this methodology are provided as bounding boxes in the images. For two detection bounding boxes \( A, B \), their Area of Overlap Ratio (AOR) is defined by:

\[
AOR(A,B) = \frac{|A \cap B|}{|A \cup B|}
\]

A detection bounding box is intersected with all the ground truth boxes in the image and if their \( AOR(A,B) > t \) then the detection is considered to be a true positive. Otherwise, the
Fig. 6.  

(a) Frame-based evaluation on the rear-view pedestrians dataset. In parenthesis: log-average miss rate between $10^{-2}$ and $10^0$ fppi (where such data exists).  

(b) Video-based evaluation on the rear-view pedestrians dataset. In parenthesis miss rate at 1 false alarm per second.  

(c) Frame-based evaluation on the Caltech dataset medium scale pedestrians. The “AccFeatSynth+Geometry” corresponds to the image pyramid block + AFS algorithm block from the presented system, and the “AccFeatSynth” corresponds to the AFS block only. See [7] for more details on the dataset and the evaluated methods.

Fig. 7.  

Examples of system outputs. Top-left image shows the full view of the rear-view camera, with blue-dotted line marking the operation area of the detection system. The rest of the images were cropped to working zone. Marked by green rectangles are the true positive detections above the 1 FAPS threshold and in red the false positives above the same threshold.
detection is considered a false positive. Ground truth objects not matched to any detection are counted as misses.

The miss detection statistic is the ratio of number of misses to the number of ground truth objects, and the false positives per image (Fppi) rate is the number of false positives divided by the total number of frames. We first use this criteria, which we denote by frame based evaluation, to obtain statistics on the frame independent detection quality.

In frame based evaluation, the number of misses is calculated by summing them over all frames. This definition is simple, but it has the disadvantage that both of the following scenarios would have the same missed detection rate:

Scenario A: Targets 1 and 2 are both repeatedly detected and missed with a cycle period of half a second. Scenario B: Target 1 is detected in all frames and Target 2 is missed in all frames.

Clearly Scenario A is preferable to B from an application point of view, but both would result in the same score. Therefore, we define a grace period, Time of Miss Gap (TMG). Misses lasting for TMG seconds or less, which occur after a successful detection, are not counted.

For false positives, the per-frame false alarm rate is meaningless from a user point of view. Even when injected the same movie, systems are different from each other with respect to the number of frames they process. We therefore shift to a false alarm per-second measure. In addition, we define the minimum duration Time of False Gap (TFG) as the time duration counted as a single false alarm. Once a false alarm occurs, and additional false alarm is counted for every TFG seconds that the false target track persists. There is no grace period for the initial false target report: a false detection is counted when it first appears. The evaluation below uses the following values: TFG= 0.5 seconds, TMG= 0.5 seconds, and t = 0.2. These values were chosen subjectively and in the future we recommend choosing them according to user experience studies. The result is reported as the miss ratio versus the number of false alarms per second (FAPS), and called the video based evaluation.

The evaluation also considers the goal range and FOV of the detection system. In our case we set it to 7 meters and 130°. We translate it roughly to a minimal height of the bounding box around the pedestrian (64 pixels), and to left/right bounds on the area of detection in the image (corresponding to the dotted-blue marked area in figure 7). In order to avoid false alarms which are correct or incorrect detections outside these bounds, we mark the ground-truth objects and detections outside the bounds as “don’t care” and disregard them in the evaluation. If a bounding box within bounds is associated with a “don’t care” ground-truth it will be not counted as a negative or positive detection. The same goes for distant grouped people, which were difficult to annotate as separate individuals, and were marked as “people”. These objects are added to a “don’t care” list of objects.

C. Evaluation results

We evaluate our system using both frame and video based evaluation on the test part of the rear-view pedestrian dataset described previously. As a baseline method we use the exact same system except for the AFS (Block C in figure 2) replaced by the HoG [5] template-based detection. The HoG is trained on the INRIA pedestrians dataset [5], as is the AFS. We refer to the two rear view detection systems in short as rv-AFS and rv-HoG. In figure 7 we show several examples of the detection results of the our system (rv-AFS) including typical false alarms. In all these examples the detection threshold is set to obtain 1 FAPS in the video-based evaluation, which is the working point we use in real time operation of the system. As can be observed the system can perform in poor lighting conditions, heavy occlusions and large image distortions.

Figure 6(a) summarizes the results using the frame-based evaluation. We plot the miss rate as a function of the fppi on a log-log scale. The number in parentheses beside each method name is the log-average of the miss rates between $10^{-2}$ and $10^0$ fppi. Figure 6(b) shows the video based evaluation result. We plot the FAPS versus the miss detection rate. Here, the number in parentheses is the miss rate at 1 FAPS. At this working point our system (rv-AFS) miss rate is 5% compared to the rv-HoG with 7.4%. Results from both evaluation methods show that using part-based detection in this experimental setting enhances the overall system performance. Figure 6(c) shows the frame-based evaluation of the AFS with the geometry constrains blocks only on the Caltech pedestrians dataset [7], for medium size pedestrians, using the standard $t = 0.5$ overlap criterion. Our method is ranked second best among the compared methods. This result was previously published in [17] where additional Caltech dataset results can be found. The on-vehicle rv-AFS system runs in real-time on an i7 laptop processing between 4 and 12 frames per second depending on the scene complexity.

D. Conclusions

We presented a system for detecting pedestrians in the rear-view camera tested on a dataset we collected for training and evaluating such systems. The system is unique in using a part-based detection method, allowing it higher robustness for detecting close-by, partially occluded and non-upright pedestrians. In the future we aim at further increasing robustness and eliminating false alarms. We believe that incorporating a more sophisticated tracking algorithm and using motion-based features to complement appearance ones can improve the overall performance. In addition we aim at improving our appearance based detector, to specifically address the difficult challenge of children detection.

REFERENCES


