Towards an Automatic Blind Spot Camera: Robust Real-time Pedestrian Tracking from a Moving Camera

Kristof Van Beeck, Toon Goedemé
Lessius Mechelen, Association K.U. Leuven
De Nayerlaan 5, St.-Kat.-Waver, Belgium
kristof.van.beeck@mechelen.lessius.eu

Tinne Tuytelaars
K.U. Leuven, ESAT/PSI-VISICS
Kasteelpark Arenberg 10, Heverlee, Belgium
tinne.tuytelaars@esat.kuleuven.be

Abstract

In this paper, we present a real-time pedestrian tracker that combines a robust appearance-based pedestrian detector with application-specific constraints and motion information. The targeted application is the automatic detection of vulnerable road users in blind spot cameras on trucks. This application imposes several challenges that need to be tackled. Vulnerable road users are a very diverse class, and we need a high precision and recall rate with real-time performance. Here, we present a first step towards such an automatic detection system. The novelty of our approach is the extension of a robust pedestrian detector towards real-time performance. The information from the appearance-based detector is used in combination with motion-based estimations to efficiently reduce the search space for the appearance-based detector in consecutive frames. This results in a multi-pedestrian tracker from a moving camera which is both optimized in terms of accuracy and speed. We recorded several data sequences to evaluate our pedestrian tracker, and performed initial experiments with promising results.

1 Introduction

Approximately 1300 casualties are caused by blind spot accidents every year in the European Union [1]. Bicyclist and pedestrians are the most common victims. The most widely used solution is the use of blind spot mirrors, which is obliged in the EU by law since 2003. However, it is shown that these mirrors are often not, or incorrectly used. We believe that an active driver-independent system offers a better solution. Therefore, our goal is to develop an application that can automatically detect vulnerable road users in blind spot cameras. When a dangerous situation occurs, the system should warn the driver. This is an extremely challenging task. Firstly, vulnerable road users are a heterogeneous object class. Besides pedestrians, we also need to detect bicyclists, children, wheelchair users and mopeds. Secondly, because the field of view of the camera covers the blind spot area on the side of the truck, we have a highly dynamic background. Since the camera is moving techniques like adaptive background estimation or background substraction, which can be calculated fast, are not an option. The biggest challenge, is the hard real-time character of the application, combined with the need for a high precision and recall rate. We only have limited time available to detect the vulnerable road users. This paper presents work on a first part of the application: we developed a pedestrian tracker for a moving camera which is both robust and fast. To achieve this, we extend Felzenszwalb & Ramanan's appearance-based pedestrian detector [5, 6], which is known for its excellent recognition performance, but is not very fast. The main idea behind our approach is that we start from a reliable frame-by-frame detection, and maximally integrate the spatial information from this detector with the temporal information at hand. This combined information is used to narrow down the search space for the appearance-based detector, thereby resulting in a fast and reliable tracker. Using this approach there is no need for a multi-camera setup or camera calibration. To test our proposed algorithm, we recorded a number of data sequences and performed initial experiments on them. The outline of this paper is as follows: in section 2 related work on this topic is discussed. Section 3 describes our pedestrian tracker. In section 4 we discuss the results of this approach. We conclude in section 5 with final remarks and future work.

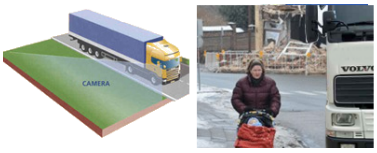

Figure 1. Blind spot area of a truck

2 Related Work

Several pedestrian tracking algorithms are already described in the literature. Most of them are based on a fixed camera, and rely on background substraction (e.g. [16, 14]), but this cannot be used in our application. Some of these static camera approaches use a thermal camera to eliminate the influence of shadows [12]. In the case of a moving camera two approaches are used. One is the use of an appearance-based detector, by using a sliding window technique: across the entire image one looks at all possible locations and all possible scales. Currently this approach does not achieve real-time performance. Methods have been proposed using a detector cascade with a fast rejection of false detections [15], while [11] proposes a powerful branch and bound scheme to tackle this problem. Our method eliminates the need for a full search over the entire image using the spatial information from the pedestrian detector, and an estimation of the next position based on temporal information.

Another approach for moving cameras is to exploit disparity characteristics [8]. When using a monocular approach, most pedestrian trackers on moving vehicles use a forward-looking camera [7, 8], or still need to be extended to multi-person tracking [13]. We

differ from these trackers: our goal is a monocular multi-pedestrian tracking system with field of view aimed sidewards, towards the blind spot of the vehicle (see fig. 1), at real-time performance. This field of view results in motion blur and large distortion. To achieve a moving camera multi-person tracker, we use a tracking-by-detection approach and start from an existing appearance-based object detector based on the discriminatively trained part models, introduced by Felzenszwalb et al [5, 6]. These authors extend the idea of the Dalal-Triggs model [2], which is based on the gradients (HOG), with a part-based model. Recently they proposed a cascade object detection using their part models [4], where they first look at hypotheses that score high using a weak model. If that is the case these hypotheses are evaluated further using a more complex model, otherwise they can immediately be discarded. This way a significant detection speed-up is achieved. We use the pedestrian part-based model and the cascade object detection as detector for our pedestrian tracker.
In [3], Enzweiler et al. give an overview and perform experiments using different pedestrian detection approaches. They compare the Dalal and Triggs model (HOG together with lineair SVM as classifier) with a wavelet-based AdaBoost cascade [16]. Besides these approaches, they also examine two other methods, one based on neural networks and one based on a combined shape-texture model. Their work clearly shows an advantage of the HOG-based approach at the cost of lower processing speeds. These results are the motivation why we use a HOG-based detector. Yet instead of using a single global HOG-descriptor, we select the parts-based model proposed by Felzenszwalb et al., which has also shown excellent results in the Pascal VOC challenge [5, 6].

3 Proposed tracker method

To achieve a high precision and recall rate, our pedestrian tracker extends the cascade object detection part-based model as proposed by Felzenszwalb et al. [4]. Their method works as follows. The object that has to be detected is described using a HOG model.

The model consists of a root filter and a number of smaller part filters. The position of each of the parts are latent variables, which are optimized during the detection (fig. 2). A first step is the construction of a scale-space pyramid from the original image. This is done by repeated smoothing and subsampling. For each entry of this pyramid, a feature map is computed, which is built using a variation of the HOG features presented by Dalal and Triggs [2]. For a specific scale one computes the response of the root filter and the feature map, combined with the response of the part filters and the feature map at twice the resolution at that scale. The transformed responses of the part filters are then combined with the response of the root filter to calculate a final score. Using weak hypothesis first, a fast rejection is possible. Our algorithm works as follows. At the first video frame, a detection runs over the entire image frame. We only search on the scales that are needed in our application. We use a linear Kalman filter [10] to estimate the next position of the pedestrian, based on a constant velocity model. Our experiments showed that this assumption holds and suffices for a robust detection. We use the position and velocity as our state estimates: $x_k = \begin{bmatrix} x & y & v_x & v_y \end{bmatrix}^T$.

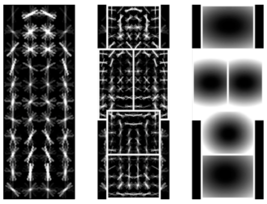

Figure 2. The used HOG model. Root filter (L), Part Filters (C), Prior estimate of postion of the part filters (R)

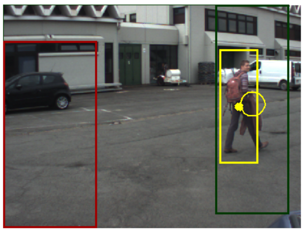

Figure 3. Example detection with estimated circular region (yellow), estimated next search space (green) and the standard search space (red)

The Kalman filter is implemented with the following time update equation: $\hat{x}_k^- = A\hat{x}_{k-1}$.

Note that $\hat{x}_k^-$ refers to the *a priori* state estimate at timestep k, while $\hat{x}_k$ refers to the *a posterior* state estimate at timestep k. We used a constant velocity motion model, and can only observe the position. The process matrix A then becomes:

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

As initial Kalman motion vector we use the opposite of the motion vector of the vehicle, because the walking speed of pedestrians is negligible as compared to the truck speed. Around the estimated new pedestrian centroid a circular region is constructed with a radius based on the position in the image. Detections which are closer to the horizon are given a smaller radius, since they are further away from the camera. This circular region is using to look for a new matching centroid in the next frame. A new search region is calculated around the estimated new centroid, based on the extension of the previous bounding box area. For the consecutive frames we only look for pedestrians in the estimated search location, thereby reducing processing time and increasing the processing speed. Overlapping bounding boxes are combined into a single search space. The use of this search space also eliminates false detections, which could otherwise be found in the image where no pedestrians are possible. Because we only search in parts of the frame based on estimates using previous detections, we need to include

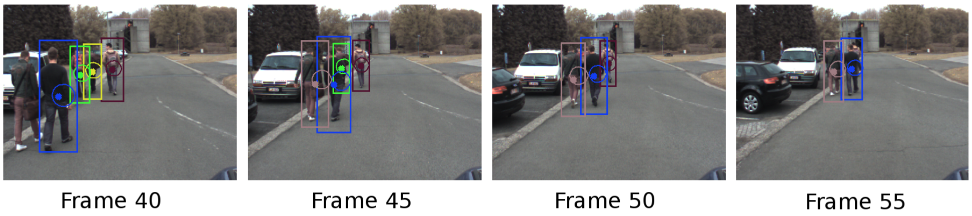

Figure 4. Example of our tracking algorithm

a way to detect pedestrians that enter the frame and pedestrians which are lost during tracking because of e.g. occlusion. One solution to the first problem is constructing a standard search region at the boundary of the image at the specific locations where one can expect that pedestrians enter the frame. The occlusion problem is handled using our Kalman motion model. Fig. 3 shows the standard and estimated next search space, together with the circular region in which a new centroid is expected.

After the search space is constructed we use our appearance-based pedestrian detector only on these parts of the image. For each pedestrian we are tracking, we evaluate if a centroid of a new detection is found in the estimated circular region. If this is the case we have a match. If multiple detections are found, the nearest one is chosen as a match. The Kalman filter is updated with this new information, a new bounding box is calculated based on a weighted average between the previous bounding box size and the current bounding box size, and a new estimate position and circular region is calculated. If for a previous tracking no match is found, we update the Kalman filter based on our prediction. If this happens for multiple frames in a row, the tracker discards this person. When a detection is found where no previous tracker was available, we start to track it from there on. Only pedestrians which can be tracked over multiple consecutive frames are shown as detected. We impose a number of application-specific constraints to improve the performance of our tracker: firstly, we reject detections of which the estimate is too far away from the centroid. Secondly, detections above the horizon and detections of which the size of the bounding box is inconsistent with the scale in the image are discarded. The latter is determined using the ratio of the bounding box size and the radius of the circular region.

4 Experiments

To test our tracking approach we recorded several video sequences with a standard camera, which was mounted on a regular car. The camera was pointed towards the blind spot region: sidewards and slightly backwards. A resolution of 640x480 and frame rate of 15 fps was used. Videos were recorded with both pedestrians and bicyclists. After editing we maintained 1279 walking pedestrian video frames. Figure 4 shows our tracking algorithm on one of the video sequences that we recorded. We have implemented our algorithm in Matlab and partially in C. The results below are computed on an Intel Xeon Quad Core with a clock speed of 3 GHz, using an unparallelized, single-threaded manner. When using the standard software as provided in [6] on an entire image frame with our resolution detection time equals on average 1.17 seconds, i.e. 0.86 fps. Using our tracking approach we dramatically reduce the processing time. The size of the search region depends on the number of pedestrians in the image. On our test set video sequences, we measured an average maximum frame rate of 12.6 fps, an average frame rate of 8.6 fps, and the average worst-case frame rate was 2.8 fps.

Table 1. Performance results.

	# frames	avg. fps	precision	recall
seq. 1	129	9.57	0.76	0.92
seq. 2	118	8.97	0.95	0.93
seq. 3	60	9.47	0.95	0.8

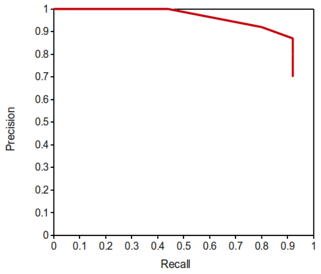

Figure 5. Precision-recall graph of our tracking algorithm as evaluated on our test set

Table 1 gives the results for three of our video sequences. Both a high precision and recall rate are achieved. Figure 5 shows the precision-recall graph as calculated over our video data set. Figure 6 displays the results of a tracking sequence of our algorithm on one of the video sequences. The top row indicates the position of the tracked pedestrians, per frame. The second row shows the size of the search area. The bottom row displays the frame rate. The average frame rate was 8.4 fps for this detection sequence. Using these figures we can evaluate the tracker based on the number of pedestrians that are detected. At the beginning of the sequence we see the initialization step, where the entire frame is evaluated. This results in the worst-case frame rate of 2.8 fps. In the next 10 frames no pedestrians enter the frame, and only the standard search region is evaluated, resulting in the best-case frame rate of 14 fps. In for example frame 30 we see that there were 4 pedestrians tracked, and the search area was approximately half of the entire image frame.

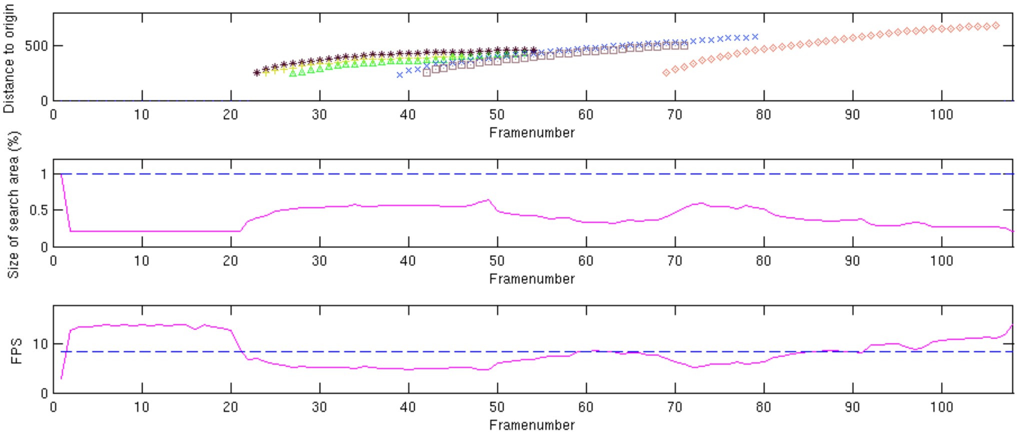

Figure 6. Results of our tracking algorithm on the video sequence shown in fig. 4. Top row: distance from centroid point to origin for each detected pedestrian. Middle row: size of the search area in % of total image size (dotted line indicates the entire image size). Bottom row: frame rate indicated per frame (dotted line shows average frame rate).

5 Conclusion and Future Work

We presented a real-time pedestrian tracker for moving cameras, by integrating an appearance-based detector and temporal information. By using application-specific constraints, and by limiting the search space we achieve an average frame rate of 8.6 fps, with a high precision and recall rate. Future work includes a further speed-up of our pedestrian tracker, which can be achieved by e.g. integration of scale information to further reduce the search space, or by a reimplementation without Matlab or on specialized hardware. In our application, the horizon was manually extracted from the images. We could also use automated methods to deduce the horizon directly from the images [9]. A next step towards the detection of vulnerable road users in blind spot cameras require the extension of our tracker to a more heterogeneous object class, and the inclusion of other image cues, such as optical flow motion.

References

[1] "Commision of the European Communities, European Road Safety Action Programme: mid-term review," Brussels, 22 february 2006

[2] N. Dalal, B. Triggs: "Histograms of Oriented Gradients for Human Detection," *International Conference on Computer Vision & Pattern Recognition*, Vol.2, no.12, pp.886-893, 2005.

[3] M. Enzweiler and D. M. Gavrila: "Monocular Pedestrian Detection: Survey and Experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.31, no.12, pp.2179-2195, 2009.

[4] P. Felzenszwalb, R. Girschick, D. McAllester: "Cascade Object Detection with Deformable Part Models," *In Proc. of the IEEE 2010 Conference on Computer Vision and Pattern Recognition*, 2010.

[5] P. Felzenszwalb, D. McAllester, D. Ramanan: "A Discriminatively Trained, Multiscale, Deformable Part Model," *In Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[6] P. Felzenszwalb, R. Girschick, D. McAllester: "Discriminatively Trained Deformable Part Models, Release 4," http://people.cs.uchicago.edu/~pff/latent-release4/

[7] A. Ess, B. Leibe, K. Schindler, L. Van Gool: "A Mobile Vision System for Robust Multi-Person Tracking," *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[8] D. Gavrila, S. Munder: "Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle," *International Journal of Computer Vision*, Vol. 73, No. 1, p.41-59, 2007.

[9] D. Hoiem, A. Efros. M. Hebert: "Recovering Surface Layout from an Image," *International Journal of Computer Vision*, Vol. 75, No. 1, 2007.

[10] R. Kalman: "New Approach to Linear Filtering and Prediction Problems," *Transaction of the ASME Journal of Basic Engineering*, Vol. 82, p.35-45, 1960.

[11] C. Lampert, M. Blaschko and T. Hoffmann: "Efficient Subwindow Search: A Branch and Bound Framework for Object Localization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2129-2142, 2009.

[12] A. Leykin, R. Hammoud: "Robust Multi-Pedestrian Tracking in Thermal-Visible Surveillance Videos," *In Proc. of the 2006 Conference on Computer Vision and Pattern Recognition Workshop* , 2006.

[13] V. Philomin, R. Duraiswami, D. Davis: "Pedestrian tracking from a moving vehicle," *In Proc. of the IEEE Intelligent Vehicles Symposium*, pp.350-355, 2000.

[14] F. Seitner, A. Hanbury, "Fast pedestrian tracking based on spatial features and colour," *In Proc. of the 11th Computer Vision Winter Workshop*, pp.105-110, 2006.

[15] P. Viola, M. Jones: "Rapid Object Detection using a Boosted Cascade of Simple Features," *In Proc. of the 2001 Conference on Computer Vision and Pattern Recognition*, 2001.

[16] P. Viola, M. Jones, D. Snow: "Detecting Pedestrians Using Patterns of Motion and Appearance," *International Journal of Computer Vision*, vol. 63, no. 2, pp. 153-161, 2005.