Robust visual pedestrian detection by tight coupling to tracking

Alexander Gepperth, Egor Sattarov, Bernd Heisele, Sergio Alberto Rodriguez
Florez

HAL Id: hal-01098703
https://inria.hal.science/hal-01098703

Submitted on 28 Dec 2014

Robust visual pedestrian detection by tight coupling to tracking

Alexander Gepperth[1], Egor Sattarov[2], Bernd Heisele[3] and Sergio Alberto Rodriguez Flores[2]

Abstract— In this article, we propose a visual pedestrian detection system which couples pedestrian appearance and pedestrian motion in a Bayesian fashion, with the goal of making detection more invariant to appearance changes. More precisely, the system couples dense appearance-based pedestrian likelihoods derived from a sliding-window SVM detector to spatial prior distributions obtained from the prediction step of a particle filter based pedestrian tracker. This mechanism, which we term *dynamic attention priors* (DAP), is inspired by recent results on predictive visual attention in humans and can be implemented at negligible computational cost. We prove experimentally, using a set of public, annotated pedestrian sequences, that detection performance is improved significantly, especially in cases where pedestrians differ from the learned models, e.g., when they are too small, have an unusual pose or occur before strongly structured backgrounds. In particular, dynamic attention priors allow to use more restrictive detection thresholds without losing detections while minimizing false detections.

I. INTRODUCTION

There is an extensive body of biological insights on various aspects of visual attention, which is sometimes seen as guided by static local image properties[1], sometimes by static spatial context[2]. Even if non-static image features, such as local motion, are used[3], [4], such attention mechanisms are always reactive in the sense that they guide attention towards the detected features but do not anticipate future events.

Recent work[5] however reveals that humans learn highly precise dynamic models *predicting* the movement of objects, and that such predictions are used to guide eye movements to the predicted locations ahead of time. This predictive mechanism is shown to permit the visual pursuit of highly dynamic objects, such as squash balls, with the very limited amount of fixations per second that can be realized by the human visual system.

As even stronger restrictions usually apply in technical systems, we consider this mechanism of predictive visual attention to be a crucial ingredient in the analysis of dynamic scenes. In particular, learned high-level models of future object behavior may permit to keep track of complex object motion with a small number of measurements (fixations), and help to make detection more robust in case of simple or no motion.

The presented work is concerned with the detection of pedestrians in traffic scenes. It introduces a mechanism similar in spirit to [5] by applying attentional feedback (termed *dynamic attention priors*, as feedback is derived from dynamic quantities such as moving pedestrians) to image locations predicted by a state-of-the-art tracking algorithm (particle-based PhD filter). This approach biases the subsequent detection step by slightly enhancing detection likelihoods around predicted particles. As long as the used prediction models are correct, this will enhance performance because detection likelihoods are boosted on and around existing pedestrians, but not elsewhere. On the other hand, no effects usually occur when motion models are violated, because the applied boosts are small and do not lead to detections when there is little detection evidence.

A. Related work

Pedestrian detection has been researched for decades, however it seems that the state-of-the-art approaches [6], [7], [8], [9], [10], [11] are reaching a boundary that is not easy to break[12]. Most approaches include tracking, usually in various flavors of Kalman or particle filtering[13], [14], [15], [16], [17], [18], [19]. Both tracking methods perform a kind of "late fusion", combining detection results with a motion model, the latter having no infleucne at all on detection process. Such a "late fusion" approach faces complications when the motion model is violated, e.g., because an object changes direction. This manifests itself by "ghost" detections that may deviate considerably from the true object position until the motion model is successfully updated based on the prediction errors. In case of noisy detection scenarios, such an update can take several seconds, resulting in a considerable time interval in which potential safety applications (e.g., emergency braking) receive incorrect data.

There is extensive literature about the computational modeling of the various aspects of visual attention[20], [1], [3], [21], [22], [23], [24], as well as some limited work attempting to make use of comparable mechanisms in technical applications[25], [26], [27], [3], [2]. This body of work shows that visual attention is a potentially powerful tool to improve real-world object detection, but also that the effort to make it work in real problems is a considerable one. There is no work we know of, except potentially [28], that makes use of dynamic attention mechanisms (i.e. derived from the dynamics of detected objects) to stabilize its detections.

II. METHODS

The overall architecture we use is depicted in Fig. 1. In fact the architecture is entirely built from state-of-the-art

---

[1] Alexander Gepperth is with ENSTA ParisTech, 828 Blvd des Marchaux, 91762 Palaiseau, France and INRIA FLOWERS, 200 Avenue de la Vieille Tour, 33405 Talence, France `alexander.gepperth@ensta.fr`

[2] Egor Sattarov and Sergio A. Rodriguez F. are with IUT d'Orsay, Université de Paris-Sud, Plateau de Moulon, 91400 Orsay `egor.sattarov,sergio.rodriguez@u-psud.fr`

[3] Bernd Heisele is with Honda Research Institute USA, Inc., 425 National Avenue, Mountain View, CA-94043, USA. `BHeisele@hra.com`
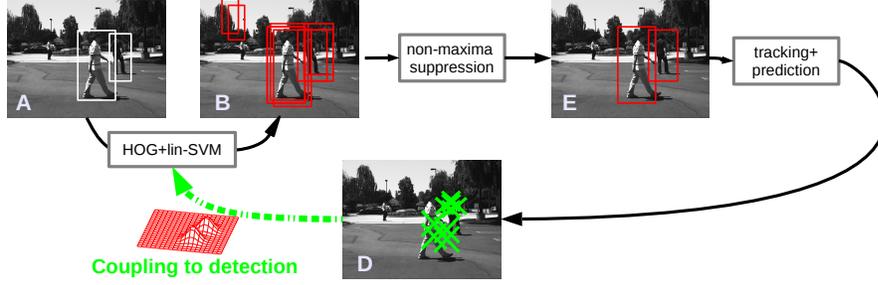
Fig. 1. Architecture schematics. A key point of the proposed architecture is the feedback path coupling tracking results (in the form of predicted particles indicated as green crosses) to detection.

algorithms such as HOG+SVM for detection and particle-based PhD filtering for tracking, giving a high credibility to the results we obtain. Additionally, the implementation of our architecture is open source which will make our results reproducible[1].

A. Detection

Pedestrian detection is performed using the method described in [10] using the free implementation available through the OpenCV computer vision library [29] in its version 2.3.1. In this way we avoid having to train a pedestrian detector by ourselves, instead relying on the "standard people detector" included in OpenCV, which is again based on the detector described in [10]. Parameter setting for HOG are the standard settings of the implementation which we repeat here for completeness. Adopting the terms presented in [10], they read:

- a cell size of 8x8 pixels
- a block size of 16x16 pixels
- a border of 0 pixels
- a window size of $w_0 \times h_0$=64x128 pixels
- a window stride of $\delta_x \times \delta_y$=8x8 pixels
- a total number of $S = 5$ spatial scales (pyramid decomposition)
- a factor of $f = \sqrt{2}$ between scales

Output of this algorithm is, for each visual input image, a set of $S = 5$ likelihood maps representing, for each scale $s \in [0, \ldots, S-1]$ all detection likelihoods in a topologically organized fashion, see Fig. 2. Each point $l(x, y, s)$ in a likelihood map at scale s is associated with a bounding box D centered on the image location $\vec{c}_D = (x\delta_x, y\delta_y)^T f^s$, and whose size $d_D = 0.5w_0 f^s$ is characterized by half its width. As the sliding detection window is fixed to $w_0 \times h_0 = 64x128$ pixels, a value of 0.5 is fixed for the aspect ratio of detections at any spatial scale, thus making it redundant to specify the two size variables.

Potentially, the likelihoods $l(x, y, s)$ can be modified at this point by a feedback from the tracking algorithm as described in Sec. II-D. By applying the detection threshold

---

[1]The code is available under www.gepperth.net/alexander/downloads/itsc2014-code.tar.gz

θ_{det} to (modified or unmodified) likelihoods, we identify a set of detections $\{D_i\}$ represented by bounding boxes with associated scores l_i, $D_i = (\vec{c}_i, d_i, l_i)$ which are passed on to non-maxima suppression (see next section) and then to tracking. The detection threshold is not fixed but is varied in the experiments from very low (-1.5) to very high (1.5) values to obtain ROC-like plots.

B. Non-maxima suppression

As the tracking algorithm does not appreciate multiple detections per object, a simple form of non-maxima suppression is performed on the detection results, see previous section. Non-maxima suppression is a standard post-processing step in object detection which expects a set of bounding boxes with associated scores $\{D_i\}$, and produces a thinned out list of boxes/scores $\{\tilde{D}_i\}$ where only the locally most confident detections survive. In detail, the algorithm runs as follows, relying on the overlap measure

$$o(D_i, D_j) = \frac{\text{area}(D_i \cap D_j)}{\text{area}(D_i \cup D_j)} \qquad (1)$$

Algorithm *Simple NMS*($\{D_i\}$)
1.   Sort $\{D_i = (\vec{c}_i, d_i, l_i)\}$ in descending order of score l_i
2.   **for** $i \leftarrow 1$ **to** N
3.   **for** $j \leftarrow i+1$ **to** N
4.   **if** D_i not marked for deletion
5.   **then**
6.   **if** $o(D_i, D_j) \geq \theta_{\text{NMS}}$
7.   **then** mark D_j for deletion
8.   Erase marked detections and return list

In all evaluations, a value of $\theta_{\text{NMS}} = 0.1$ is used.

C. Particle-based PHD tracking

The PHD filter is represented by a time-variable number of tracks $\{x_k\}$. Each track x_k contains N particles. Each particle $\xi_{x_k,n}$ contains the quantities $\vec{c}_{x_k,n}$, $d_{x_k,n}$ and $\vec{v}_{x_k,n}$, $\vec{c}_{x_k,n}$ being the center coordinates, $d_{x_k,n}$ the detection scale and $\vec{v}_{x_k,n}$ the associated speed. Also, each particle has a weight $\omega_{x_k,n}$. A track's parameters are: death probability P_d, birth probability P_b, false negative probability $P_f n$, a vector of resampling and association parameters $\sigma_i, i \in \{0, 1\}$, one for each spatial dimension. In order not to complicate the
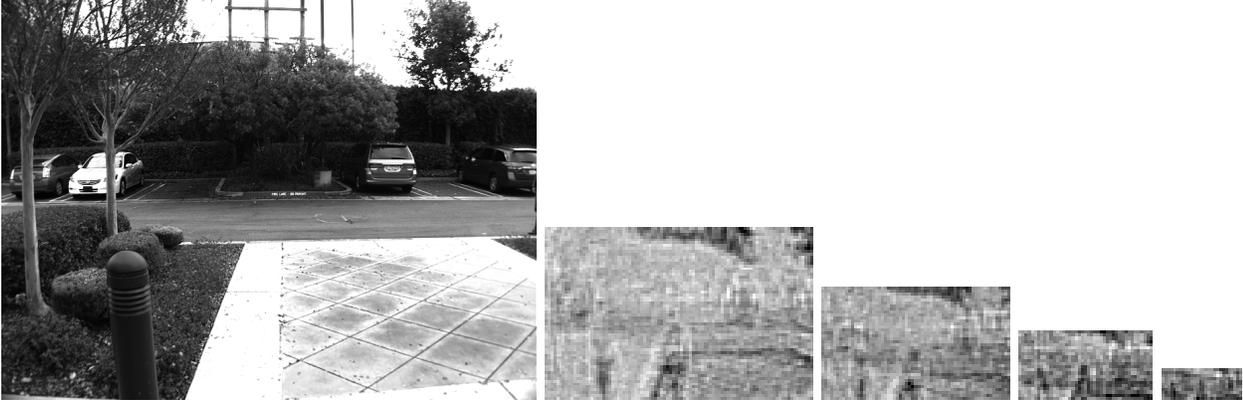
Fig. 2. Topologically arranged detection likelihoods as computed by the HOG+SVM method from the camera image (left). Likelihoods are computed on increasingly coarse spatial scales (second-to-left to right), which means that likelihood images effectively diminish in size.

tracking with multiple overlapping detections, we perform non-maxima suppression on detections as detailed in the previous section. Additionally, we impose an additional condition on detections $\tilde{D}_i$ in order to be passed to the tracking: their confidence l_i must satisfy $l_i \geq \theta_{\text{track}}$. This is to prevent spurious detections from forming tracks, which would reinforce the same spurious detections in the case of feedback, which could lead to permanent "ghost" detections.

Concrete parameter values are: $P_d = 0.1$, $P_b = 0.3$, $P_f n = 0.1$, $\sigma_i = 15$pix, $\theta_{\text{track}} = $ variable, $N = 150$.

1) Prediction : Tracks and theirs particles propagate themselves according to the movement model: $\xi_{x_k,n,t|t-1} = f_{\xi_{x_k,n,t}|\xi_{x_k,n,t-1}}(\xi_{x_k,n,t-1})$. For the particular case, $d_{x_k,n,t|t-1} = d_{x_k,n,t-1}$, $\vec{v}_{x_k,n,t|t-1} = \vec{v}_{x_k,t-1}$, $\vec{c}_{x_k,n,t|t-1} = \vec{c}_{x_k,n,t-1} + \vec{v}_{x_k,n|t-1}$.

2) Association : Input observations after non-maxima suppression, $\{\tilde{D}_j\}$, are assigned to existing tracks x_k, see below for more details. Those tracks which are assigned observations increase their associated probability by P_d: $P_{k,t} = \max(P_{k,t-1} + P_d, 1)$. Tracks that are not associated update their probability by: $P_{k,t} = \max(P_{k,t-1} - P_d, 0)$. Observations which are not associated to any tracks create new tracks having an initial probability of $P_{k,t} = P_b$. In case a new track is created, the probabilities of its particles are $\omega_{x_k,n} = P_{k,t}/N, n = 1, N$. In detail, we proceed as follows:

1) For all pairs $(\tilde{D}_j, x_k)$, we calculate the distance $G(\xi_{x_k}, \tilde{D}_j) = \mathcal{N}(\vec{c}_{D_j} - \vec{c}_{x_k,n,t|t-1}, \sigma_x)\mathcal{N}(d_{D_j} - d_{x_k,n,t|t-1}, \sigma_d)\mathcal{N}(\vec{v}_{D_j|\xi_{x_k}} - \vec{v}_{x_k,n,t|t-1}, \sigma_v)$ as a product of three Gaussians with manually tuned variances $\sigma_{c,d,v}$, where the "speed" of a detection relative to a particle is defined as $\vec{v}_{D_j|\xi_{x_k}} = \vec{c}_{x_k,n,t|t-1} - \vec{c}_{D_j}$.
2) Calculate an association threshold, based on σ_i.
3) Find the nearest pair. If the distance is smaller than the threshold, associate chosen detection to chosen track. Remove this track from list of pairs to associate. Repeat 2). If the distance is smaller than the threshold, go to 3).
4) Finally, we have a list of associated pairs, a list of non-associated detections and a list of non-associated tracks.

3) Observation : For each new observation $\tilde{D}_j$ and for each particle $\xi_{x_k,n}$ of the track the observation was assigned to, a distance is calculated: $G(\xi_{x_k,n}, \tilde{D}_j)$. The distances are normalized relative to observations. Therefore, the sum of all distances from one observation is one. The weights of particles are calculated as a sum of distances: $\omega_{x_k,n} = \sum_j G(\xi_{x_k}, \tilde{D}_j) + \omega_{x_k,n} \times P_f n$. The last term represents the "old" particle weights in order to stabilize fluctuations.

4) Resampling : If its probability falls below a threshold θ_d, a track is deleted. Otherwise, its particles resample themselves using random fluctuations, according to the $\sigma_i, i = 1 \ldots D$ and a multiplier which increases σ_i if a track's probability is lower than one (in order to disperse particles when a track is "lost", making it easier to "pick up" the track later). For new particles: $\Xi_{x_k,t} = R(\Xi_{x_k,t|t-1})$, where Ξ represents a full set of particles for one track, and $R()$ is a random fluctuation function.

5) Merging: If some of tracks are very close and move with the same trajectory, they are supposed to be one and the newest track is deleted.

6) Correction: New track states are found by averaging over the internal states of all associated particles: $x_k, t = \frac{1}{N} \sum_{n=1}^{N} \xi_{x_k,n,t} \omega_{x_k,n,t}$

D. Integrating dynamic attention priors into detection

The principal quantities used for generating dynamic attention priors from tracking results are the predicted particles $\{\xi_{x_k,n}\}$, which need to be computed *before* the detection step is applied to a new image. Each particle will slightly boost detection scores in a Gaussian profile around its image center position $\vec{c}_{x_k,n}$ at its scale $d_{x_k,n}$, where the radius of influence σ_p is a free parameter of the method which we set to half a window width computed at the particle's scale. The second components are the detection likelihoods $l(\vec{c}^L, d^L)$ provided by detection, see Sec. II-A, which depend on their spatial positions $\vec{c}^L$ and their scale index $d^L \in \{0, 1, 2, 3, 4\}$. As in Sec. II-A, the associated center coordinates $\vec{c}^I$ and scale d^I in image coordinates need to be computed first in order to determine whether they are in a particle's radius of influence

defined by σ_p:

$$\vec{c}^{I} = \left(c_x^L \delta_x, c_y^L \delta_y\right)^T f^{d^L}$$
$$l(\vec{c}^L, d^L) \rightarrow l(\vec{c}^L, d^L) +$$
$$+ k \sum_{x_k, n} \omega_{x_k, n} g(\vec{c}^{I} - \vec{c}_{x_k, n}, \sigma_{\text{sp}}) \times$$
$$\times g(\log_{\sqrt{2}}(d_{x_k, n} w_0^{-1}) - d^L, \sigma_{\text{sc}})$$
$$\text{with } g(x, \sigma) = \exp(-\frac{x^2}{2\sigma^2}), \quad (2)$$

where the sum runs over all particles, the coefficients $\omega_{x_k, n}$ being their weights, and k being a coefficient that governs the global influence of feedback, for which we chose a value of $k = 2$. For the scale parameter σ_{sc}, we chose a value of 1.0 that will affect neighbouring scales only at a factor of e^{-1}, whereas we set a rather broad spatial influence $\sigma_{\text{sp}} = w_0 f^{d^L}$.

After the modification of scores, detection goes on as described in Sec. II-A, the only difference being that the modified detection scores now implicitly contain information about previous detections and the trajectory model that is assumed.

It should be noted that eqn.(2) has a time complexity that is linear in the number of particles and pixels in the likelihood maps. However, strong speedups can be obtained by carrying out the calculations only for pixels close to the centers of gravity of particle groups.

III. EXPERIMENTS AND RESULTS

To evaluate the performance of the presented system, we test it on a set of benchmark videos containing a single pedestrian each appearing before a complex background[2]. In order to show the benefit of dynamic attention priors, we choose those videos where the used pedestrian detector, see Sec. II-A, does not achieve satisfactory performance on its own. This is due to a number of reasons: first of all, pedestrians having a size slightly smaller than the nominal 64x128 detection window are common. Secondly, the background is complex and thus violates the learned models. And lastly, pedestrians exhibit unusual poses, in most cases the extreme striding pose which causes problems to many detectors. Examples and details of these videos can be found in Fig. 3 and in Tab. I.

A. Evaluation conditions and measures

We measure performance in the *baseline condition* by deactivating dynamic attention priors (setting $k = 0$), which amounts to just measuring the performance of detection followed by non-maxima suppression. In all cases, the evaluation measure is a ROC-like diagram obtained by varying the detection threshold θ_{det}. This diagram plots average false detections per image against the percentage of correctly found objects. In the *feedback condition*, we activate dynamic attention priors by setting $k = 2$, whereupon detection likelihoods are modified as explained in Sec. II-D. We run the

[2]Video streams and annotations are available for download under www.gepperth.net/alexander/downloads/itsc2014.tar.gz

system several times in the feedback condition using different values of the tracking threshold θ_{track}, in order to determine sensible values for this parameter while keeping in mind that higher values are probably more suited as they prevent the appearance of self-stabilized "ghost detections", see Sec. II-C.

The basic measure for determining whether a detection D_i has hit an annotated object A_i is the overlap measure introduced in Sec. II-B. Using an association threshold $\theta_{\text{ass}} = 0.4$, we thus demand $o(D_i, A_j) \geq \theta_{\text{ass}}$ for a correct detection. This low threshold value was chosen due to the less-than-perfect quality of annotations. Any detection that does not satisfy this condition for at least one annotation is considered a false detection. If an annotation is smaller than 50×100 pixels, it is removed from the set of annotations. If it is detected nevertheless, it will not count as a false detection.

B. Results

As can be seen in Fig. 4, the detection performance in the feedback condition strongly outperforms the baseline condition in all streams except in stream 07 where baseline and feedback condition show approximately the same performance. Generally, a threshold setting of $\theta_{\text{track}} = 0.4$ seems to be the most favorable, which shows that a conservative threshold is needed here in order to avoid self-stabilized false detections. These findings are discussed in Sec. IV.

C. Computational performance

As mentioned in Sec. II-D, the computational cost incurred by dynamic attention priors can be strongly reduced by a few simple heuristics. Here, we choose to modify only the 11x11=121 likelihoods around a given particle. As there are no complex functions, apart from the Gaussian, to be evaluated in this process, the computational overhead is very small, decreasing the frame rate on a current off-the-shelf Linux PC from 10.2 to 9.2 fps. By using lookup tables, this number might be reduced even further which we did not attempt for the time being.

IV. DISCUSSION

The presented system demonstrates that considerable performance increases can be obtained by a rather simple and computationally inexpensive approach: extrapolating future pedestrian positions by analyzing their past trajectories and subsequent modification of detection likelihoods. After analyzing the reasons for this overall performance increase, we can state that it is due to the following points:

- "Filling in" of intermittent missed detections. Often, pedestrians are missed because they exhibit an unusual pose that will last for only a single frame. Most prominent here is the extreme striding pose where the legs are maximally apart, which is apparently under-represented during detector training. Due to attentional feedback, these detections, which seem to be just slightly sub-threshold, are pushed past the detection threshold θ_{det}.
- Detection of small pedestrians. When pedestrians get smaller than the size of the smallest detector window,

Fig. 3. Example images from evaluation streams. Background and pedestrian identity and clothing vary strongly between video streams, and lighting conditions are less than optimal.

stream	1	2	3	4	5	6	7	8	9	10
images	1023	751	742	698	1479	720	558	982	1011	988

TABLE I

VIDEOS USED FOR TESTING. THERE ARE 10 SEQUENCES CONTAINING A SINGLE PEDESTRIAN IN SOME BUT NOT ALL IMAGES. THE TOTAL NUMBER OF IMAGES IS 9766, WHICH GIVES A LENGTH OF 16 MINUTES AT A FRAME RATE OF 10Hz.
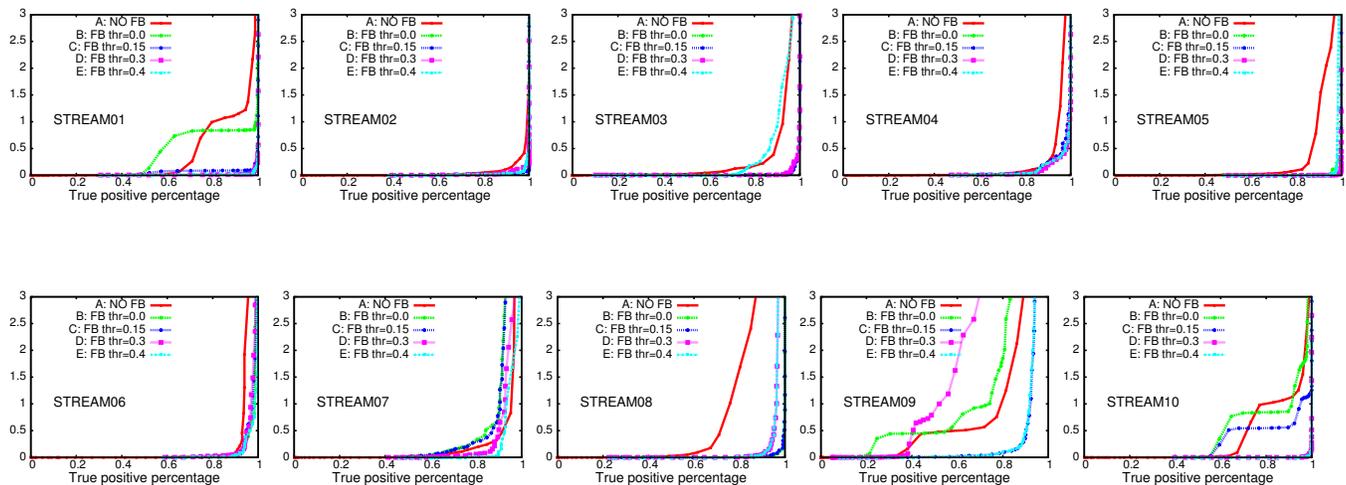

Fig. 4. Effect of dynamic attention priors on each of the 10 test streams. In each plot, there is either no feedback (**A**) or feedback where the tracker is fed only with detections that exceed confidence thresholds of 0 (**B**), 0.15 (**C**), 0.3 (**D**) and 0.4 (**E**). A strong overall improvement of detection rates is clearly observable.

i.e., 64×128 pixels, their likelihoods become sub-threshold. A small boost by attentional feedback seems to do the trick here as well.

- More restrictive detection thresholds. As detections are boosted by attentional feedback from tracking, a higher threshold is less likely to affect them provided a track could be created in the first place. Therefore, higher detection thresholds can be used, leading to less false detections, while not increasing the number of missed detections.

In stream 07, feedback condition performance is roughly comparable to performance in the baseline condition. When viewing this stream visually, the reason is very obviously a self-stabilized false detection which negates the performance advantages otherwise obtained by attentional feedback. This just shows that a parametrization of the system has to be found which avoid this kind of effect. We could have eliminated the problem by specific parameter settings for stream 09, but chose not to do so as the parameters should be the same over all test streams. In particular, a slightly higher tracking threshold would have been sufficient, but only at the

expense of performance on other streams.

While results are encouraging, a few words of caution regarding the limits of dynamic attention priors are in order: first of all, extrapolating future pedestrian locations happens in image coordinates: any movement of the ego-vehicle will have to be either compensated for, or else integrated into the prediction algorithm. And secondly, great care must be taken to keep the feedback strength k (see Sec. II-D) sufficiently small in order to avoid positive feedback loops that lead to detections that can no longer disappear, thus being tracked, and finally reinforcing themselves.

Apart from this, we believe that dynamic attention priors have great potential in the context of pedestrian detection, as they reduce the dependency on appearance-based models: a pedestrian can thus still be detected if he moves consistently, even if his visual pattern is slightly different from the learned one due to rotation, deformation or structured background. Normally one tries to solve this problem by brute force, simply adding as many examples as possible to the training process and hoping to represent all relevant appearances in the right proportion. What we have shown here can potentially reduce the need for such costly efforts, allowing to focus on simple detectors and rather spend time on the creation of proper motion/behavior prediction models for pedestrians, which will be useful for security products in any event.

V. Conclusions

We see the presented work rather as a proof-of-concept than a hard performance benchmark: many parameters need to be investigated more closely in order to make the performance of the proposed system as generic as possible. A test on existing benchmark databases was not yet done as it would have had to include 3D information and vehicle ego-motion (which are available, for example, in the KITTI[30] database) into the tracking process, which we felt was too complex for the time being.

In the future, we will work to make the proposed method suitable for safety products by integrating 3D information and vehicle ego-motion into the tracking process, thus allowing to predict pedestrian positions from a moving car. An evaluation on a public benchmark database will then be the logical next step.

References

[1] L. Itti and C. Koch. Computational modeling of visual attention. *Nature Reviews Neuroscience*, 2(3):194–203, Mar 2001.

[2] Antonio Torralba. Contextual priming for object detection. *IJCV*, 53:2003, 2003.

[3] S Frintrop, G Backer, and E Rome. Goal-directed search with a top-down modulated computational attention system. In *Pattern Recognition*, Lecture Notes in Computer Science. Springer, 2005.

[4] Thomas Michalke, Alexander Gepperth, Martin Schneider, Jannik Fritsch, and Christian Goerick. Towards a human-like vision system for resource-constrained intelligent cars. In *The 5th Int. Conf. on Computer Vision Systems Conference*. Universit.tsbibliothek Bielefeld, 2007.

[5] Mary M Hayhoe, Travis McKinney, Kelly Chajka, and Jeff B Pelz. Predictive eye movements in natural vision. *Experimental brain research*, pages 1–12, 2012.

[6] M. Enzweiler, A. Eigenstetter, B. Schiele, and D.M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. pages 990–997. IEEE, 2010.

[7] M. Enzweiler and D.M. Gavrila. Integrated pedestrian classification and orientation estimation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 982–989. IEEE, 2010.

[8] Tarak Gandhi and Mohan M. Trivedi. Pedestrian protection systems: Issues, survey, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 8(3):413–430, 2007.

[9] B. Schiele P. Dollár, C. Wojek and P. Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition (CVPR)*, June 2009.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[11] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *British Machine Vision Conference (BMVC)*, 2010.

[12] B. Schiele P. Dollar, C. Wojek and P. Perona. Pedestrian detection: An evaluation of the state of the art. 2011.

[13] Darko Musicki and Barbara La Scala. Multi-target tracking in clutter without measurement assignment. *Aerospace and Electronic Systems, IEEE Transactions on*, 44(3):877–896, 2008.

[14] C Hoffman, Thao Dang, and Christoph Stiller. Vehicle detection fusing 2d visual features. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 280–285. IEEE, 2004.

[15] Uwe Franke and Clemens Rabe. Kalman filter based depth from motion with fast convergence. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 181–186. IEEE, 2005.

[16] Xavier Clady, François Collange, Frederic Jurie, and Philippe Martinet. Cars detection and tracking with a vision sensor. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 593–598. IEEE, 2003.

[17] Jens Schmuedderich, Nils Einecke, Stephan Hasler, Alexander Gepperth, Bram Bolder, Robert Kastner, Mathias Franzius, Sven Rebhan, Benjamin Dittes, Heiko Wersing, Julian Eggert, Jannik Fritsch, and Christian Goerick. System approach for multi-purpose representations of traffic scene elements. In *International IEEE Annual Conference on Intelligent Transportation Systems*, 2010.

[18] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. Wiley-Interscience, 2006.

[19] Hedvig Sidenbladh. Multi-target particle filtering for the probability hypothesis density. In *6th International Conference on Inforation Fusion*, 2003.

[20] V Navalpakkam and L Itti. An integrated model of top-down and bottom-up attention for optimal object detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2049–2056, New York, NY, Jun 2006.

[21] JM Wolfe. Guided search 2.0: a revised model of visual search. *Psychonom. Bull. Rev.*, 1:202–238, 1994.

[22] M.W. and Spratling. Predictive coding as a model of biased competition in visual attention. *Vision Research*, 48(12):1391 – 1408, 2008.

[23] Fred H Hamker. Modeling feature-based attention as an active top-down inference process. *Biosystems*, 86(1-3):91–99, 2006.

[24] Gustavo Deco and Edmund T Rolls. A neurodynamical cortical model of visual attention and invariant object recognition. *Vision Res*, 44(6):621–642, Mar 2004.

[25] Thomas Michalke, Alexander Gepperth, Martin Schneider, Jannik Fritsch, and Christian Goerick. Towards a human-like vision system for resource-constrained intelligent cars. In *The 5th International Conference on Computer Vision Systems*, 2007.

[26] Alexander Gepperth, Sven Rebhan, Stephan Hasler, and Jannik Fritsch. Biased competition in visual processing hierarchies: A learning approach using multiple cues. *Cognitive Computation*, 3(1):146–166, 2011.

[27] B. Dittes, M. Heracles, T. Michalke, R. Kastner, A. Gepperth, J. Fritsch, and C. Goerick. A hierarchical system integration approach with application to visual scene exploration for driver assistance. In *ICVS*, 2009.

[28] B Leibe, N Cornelis, K Cornelis, and L Van Gool. Dynamic 3d scene analysis from a moving vehicle. 2007.

[29] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Incorporated, 2008.

[30] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.