Learning a Multiview Part-based Model in Virtual World for Pedestrian Detection

Jiaolong Xu, David Vázquez, Antonio M. López, Javier Marin and Daniel Ponsa
Computer Vision Center
Autonomous University of Barcelona
Edifici O, 08193 Bellaterra, Barcelona, Spain
{jiaolong, dvazquez, antonio, jmarin, daniel}@cvc.uab.es

***Abstract*— State-of-the-art deformable part-based models based on latent SVM have shown excellent results on human detection. In this paper, we propose to train a multiview deformable part-based model with automatically generated part examples from virtual-world data. The method is efficient as: (i) the part detectors are trained with precisely extracted virtual examples, thus no latent learning is needed, (ii) the multiview pedestrian detector enhances the performance of the pedestrian root model, (iii) a top-down approach is used for part detection which reduces the searching space. We evaluate our model on Daimler and Karlsruhe Pedestrian Benchmarks with publicly available Caltech pedestrian detection evaluation framework and the result outperforms the state-of-the-art latent SVM V4.0, on both average miss rate and speed (our detector is ten times faster).**

I. INTRODUCTION

Advanced Driver Assistance Systems (ADAS) aim at improving traffic safety by providing warnings and performing counteractive measures in dangerous situations. Reliable image-based pedestrian detection is the major challenge of a pedestrian protection system, a type of ADAS, because the pedestrians present high variability in clothes, pose, viewpoint and distance to camera, all under uncontrolled outdoor illumination [1]. Multiresolution pyramidal detection tries to cope with the variability in distance to the camera, proper features/descriptors address variability due to clothes, and illumination changes (*e.g.*, HOG [2]), while multiview and part-based models focus on robustness to imaging viewpoint and pose variability respectively. Multiview and part-based models for pedestrian detection are the focus of this paper.

For instance, multiview models have proven to be effective for face recognition as well as pedestrian detection. The rational behind this is that mixing views for model training turns out on more blurred models than by somehow training a model per view. Training multiview models requires clustering the object examples according to the considered views. This can be done manually or automatically, thought depending on the view granularity and object complexity, it is not a trivial procedure. Shape (or contour) has been effective for viewpoint and pose clustering, and widely used for pedestrian classification [3], [4], detection and tracking [5], direction estimation [6].

The changing pose and articulation of the limbs suggests a classifier based on the integration of local image representations as opposed to a holistic representation and a sub-region method has been demonstrated effective in [7]. Part-based models try to capture the idea that most objects are composed of parts and these can only be in a set of possible relative positions. Some part-based models [8], [9], [10] require manual labelling of object parts in order to perform supervised training of such models. Since obtaining manual labels is a tiresome process prone to errors, no large amounts of reliable examples are usually available for training. This can eventually limit the generality of the learnt models, and so their performance. However, the most promising method up to date for training part-based object models seems to be the structural latent SVM [11], which is giving excellent results for pedestrian detection [12] and does not require labelled body parts for performing the training of the pedestrian model. Moreover, latent SVM also includes view point as latent information, thus allowing learning more general object models. The price to pay, however, is that this method turns out to be complicated in training and sensitive to initialization.

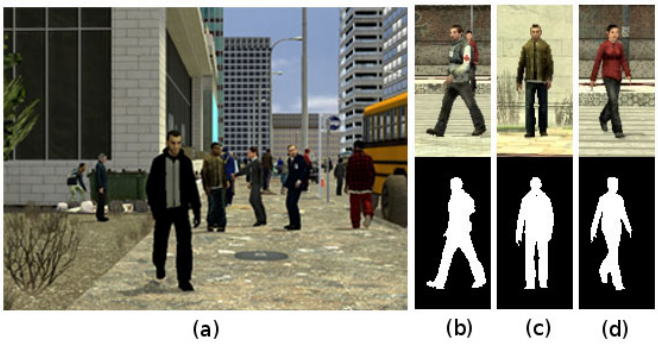

Fig. 1. Virtual-world data: (a) image; (b)-(d) pedestrian examples (top) with their corresponding groundtruth mask (bottom).

Focusing on the problem of multiview and part-based pedestrian detection, we think that one of the key issues is to have reliable labelled examples, *i.e.*, the bounding box of the pedestrian examples as well as labelled parts. Obtaining cheap annotated pedestrians has been addressed in our previous work [13], where we generated virtual-world data using a video game. No part annotations were used,

just the pedestrians bounding box. The developed pedestrian detector, based on a holistic HOG/Linear-SVM pedestrian classifier, showed a performance comparable to analogous detectors obtained from real-world manually labelled data.

Accordingly, in this paper we further explore the potential of video game based virtual-world data for training a multiview deformable part-based pedestrian model. As we will see, thanks to this type of data we can train and use such a flexible pedestrian model without the need of latent variables accounting for pose and view. The consequence is that we obtain equal or even better performance but speeding up training, and specially detection (testing). In particular, our pedestrian detector is ten times faster than the one based on latent SVM.

The rest of the paper is organized as follows. Section II illustrates the procedure of multiview clustering of examples and their automatic parts labelling with virtual-world dataset. Section III describes our multiview deformable part-based model and the corresponding training. Performance assessment and comparison to latent SVM 4.0 are given in section IV. Finally, section V draws the main conclusions.

II. Virtual-World Training Examples

The virtual-world dataset is created from the video game Half-Life 2 based on our previous work in [13]. The scenarios contain realistic virtual cities with roads, trees, buildings, vehicles, traffic signs, pedestrians as well as different illumination conditions. The pedestrians and vehicles follow physical laws and artificial intelligence. We obtained images containing pedestrians with different poses and backgrounds as well as pixel-wise groundtruth. The images of the dataset have resolution of 640×480 pixels. Fig. 1 shows the virtual-world data: image, pedestrian examples and corresponding groundtruth.

A. Multiview pedestrian clustering

Our previous dataset in [13] only contains holistic bounding box annotations without viewpoint label while in this paper we further explore its potential to create a multiview pedestrian dataset. The pedestrians can be naturally divided into three relatively separated classes according to their views: frontal/back, left and right. This is interesting because mixing all views eventually would give rise to a more blurred model than using per-view separate models [14]. Moreover, with the virtual-world data the type of view is known automatically. Unlike the manually separated examples in [14], we generate multiview pedestrian examples automatically using template convolution. The multiview templates can be computed by averaging a few groundtruth masks of each view (see Fig. 1 (b)-(c)-(d)). In particular we used only 50 groundtruth samples for the frontal view and the same number for the left view (right one is just obtained by mirroring). We apply template convolution as

$$F(T) = \arg\max_k \frac{T_k \otimes T}{\|T_k\|}, \; k = 1, ..., m \; , \qquad (1)$$

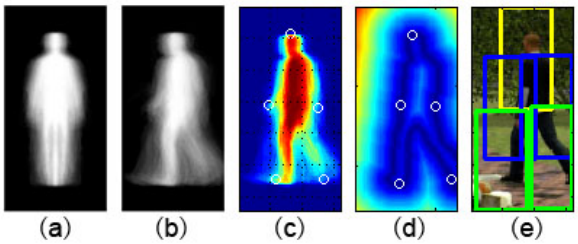

Fig. 2. Matching part locations by distance transform. Figure (a) and (b) are the templates. Figure (c) shows the canonical part locations and figure (d) is the matching result of the part locations. Figure (e) shows the locations of the part cropping window for the virtual pedestrian example.

where T is the mask of the pedestrian example to be clustered, m is the number of the root classes, T_k is the k-th template and $F(T)$ is the root class which gives maximum response to template convolution. In [11], mixture components are clustered by comparing bounding box aspect ratio and symmetry components are further clustered by comparing the similarity of the HOG features. However, this method may fail since the bounding boxes of pedestrians usually have canonical size for most pedestrian datasets. Thus, the shape based clustering method is more reliable for the multiview pedestrian clustering.

B. Localization of Pedestrian Parts

1) Initialize the parts of a pedestrian model: The part anchor points are inexplicit in latent SVM framework, thus they must be obtained by heuristic initialization. With virtual-world data, the anchor points can be obtained in an explicit way. For each pedestrian example we consider its groundtruth mask (Fig. 1 (b)-(d)), we compute the corresponding binary skeleton and obtain the end points of it. Such end points are considered as semantic anchor points (center of the head, end of the arms, etc.). Thus, the accumulation of all them is understood as the sampling of an underlying distribution of semantic anchor points locations. We model such a distribution using a Gaussian Mixture Model (GMM) that is fitted using the Expectation-Maximization (EM) algorithm. We set the number of clusters to five: head, arms and legs. The overall body is expected to be captured by the root model, but we set the parts canonical size to 32 x 64 pixels, which implies that, although centred in the head and the extremities, the five parts cover the whole body. The procedure is an offline process which runs in less than 80 seconds and is done just once during the training[1].

2) Locate the parts by distance transform: We define canonical part location C_j as the point located on the silhouette of the template and is used to identify part j (See Fig. 2 (c)). Firstly, we locate the full body cropping window to a precise position by finding the maximum response of the template convolution, a procedure which is similar to the one in section II-A. Secondly, we match the canonical part locations to the silhouette by distance transform, as

[1]All experiments in this paper run with Intel Xeon CPU E5420 @ 2.5GHz, 16GB RAM.

illustrated in Fig. 2 (d). In this way, we are able to extract part examples precisely. The pixel-wise groundtruth plays an important role in the procedure but it is usually not available in real-world training data.

III. Virtual Deformable Part-based Model

In the following, we briefly review the deformable part-based model (DPM) in [11], *i.e.*, based on latent SVM, and then move on to our multiview part-based model and its corresponding training. For comparability, we adopt the notation of [11]. We use the term VDPM to refer to our counterpart model as trained with virtual-world dataset.

A. DPM review

A DPM is a mixture model with m components. Each component is a star structural model consisting of a coarse root and n parts placed in the double resolution layer of the feature pyramid. The object hypotheses of component $c, (1 \leq c \leq m)$, is defined by $z = (c, p_0, \ldots, p_n)$, where $p_j = (u_j, v_j, \sigma_j)$ specifies the position (u_j, v_j) and scale level σ_j of part j, $(j = 1, ..., n)$. For the general training dataset, we can identify the bounding box of root part p_0 whereas the parts $p_1, \ldots, p_n$ are not available and they are the so-called latent variables.

Given training dataset $\{(I_i, y_i)\}_{i \in \{1,\ldots,N\}}$, where I_i defines an example and $y_i \in \{-1, 1, \ldots, m\}$ is the background (-1) or the class of the components $(1, \ldots, m)$. The score of a hypothesis z can be expressed in terms of a dot product, $score = \beta \cdot \Psi(I_i, y_i, z)$, where $\beta = (\beta_1, ..., \beta_m)$ is the vector that contains all parameters of all mixture components and $\Psi(I_i, y_i, z)$ is a sparse feature vector with non-zero entries $\psi(I_i, c, z)$ at component c. The detectors of different component are trained in a one-versus-rest way. Using the standard hinge loss, the optimization problem for component c is:

$$\begin{aligned} \min_{\beta, \xi \geq 0} \quad & \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \\ sb.t. \quad & \forall i : y_i = c : \max_h \langle \beta, \Psi(I_i, y_i, z) \rangle \geq +1 - \xi_i \\ & \forall i : y_i \neq c : \max_h \langle \beta, \Psi(I_i, y_i, z) \rangle \leq -1 - \xi_i \ . \end{aligned} \tag{2}$$

B. Our VDPM proposal

With virtual-world dataset, views and corresponding n part locations can be obtained using the methods in section II, thus we can remove latent variable $z = (c, p_0, \ldots, p_n)$ from (2), *i.e.*, the latent SVM is simplified to the classic linear SVM. Therefore, our model is explicit and straightforward. In particular, is composed of a joint multiview root classifier, part classifiers and a combination classifier (ensemble). The ensemble takes into account the scores from each classifier as well as the part deformation cost.

1) Multiview root models: Combining several dependently trained root classifiers together may perform worse than a single root classifier, because the output score of the root classifiers may not be comparable. Considering this, we define our multiview root model as a joint model which appends the features of the multiple roots together, similar to [15]. The joint features $\Psi_r(I_i, y_i)$ can be expressed as

$$\Psi_r(I_i, y_i) = \begin{cases} (\psi_r(I_i, y_i), 1, 0, 0), & \text{if } y_i = 1 \\ (0, 0, \psi_r(I_i, y_i), 1), & \text{if } y_i = 2 \ , \end{cases} \tag{3}$$

where y_i indicates the root class (we define *frontal* as 1 and *left* as 2), $\psi_r(I_i, y_i)$ denotes the features of root y_i, and "1" is the bias term. Equation (3) is equivalent to training each multiview root model independently, but in this way the bias term is balanced because they refer to the same hyper-plane in the same vector space. By doing so, we can obtain reliable root classifiers and the training can even be accelerated as the independent models can be trained in parallel. Note that by using the same strategy of processing symmetry models in [11], we can omit the training of the right view model. In this paper, we only use three views, namely, frontal/back, left, and right views. We do not separate the front and rear views is because the two views are similar for HOG-based models and more views require more expensive training.

2) Deformable part models: We adopt the star structure for our deformable part models. The score of an object hypothesis is given by the score of appearance minus the deformation cost of each part. For the star model, the general energy minimum function can be expressed as

$$B_j(l_i) = \min_{l_j} (m_j(l_i) + d_{ij}(l_i, l_j)) \ , \tag{4}$$

where l_i is the location of the root node, l_j is the location of the part node, $m_j(l_i)$ is the appearance match cost, and $d_{ij}(l_i, l_j)$ is the deformation cost. The energy function can be understood as generalized distance transforms and can be solved by dynamic programming [10]. The distance transform over part grid space Ω can be expressed as

$$D_f(T_j(l_i)) = \min_{l_y \in \Omega} (\rho(T_j(l_i), l_y) + f(l_y)) \ , \tag{5}$$

where ρ and f are the distance measure and generalized indicator function, respectively. Remember that the $T_j(l_i)$ is the transformed root location which corresponds to the anchor point of part j in our star model and $f(l_y)$ is the appearance matching cost at position l_y. Note that, for the general DPM, the transformed root locations are obtained by heuristic initialization while for our VDPM, the canonical part locations are explicitly given. Moreover, our part geometric models are naturally defined as GMM, which means we can directly adopt the Mahalanobis distance to measure the deformation cost in the energy function. The Mahalanobis distance on the grid space of part j is

$$d_{ij}(l_i, l_j) = (T_j(l_i) - l_j)^T M_{ij}^{-1} (T_j(l_i) - l_j) \ , \tag{6}$$

where $T_j(l_i)$ is the anchor point of part j, and M_{ij} is the covariance matrix. However, the model can be even simplified by top-down strategy which does not require dynamic programming. We will explain it in the next section.

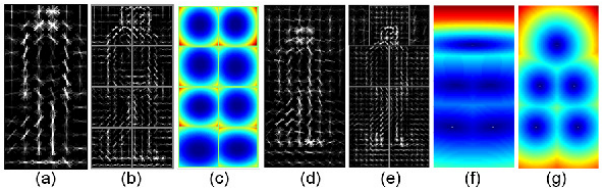

Fig. 3. Visulization of the DPM model (best viewed in color). The models from latent SVM: (a) to (c) are the coarse root model, part models, and deformation cost respectively. The models from VDPM: (d) to (g) are the coarse root model, part models, deformation cost with Mahalanobis distance, and deformation cost with Euclidean distance respectively.

3) Top-down inference: Dynamic programming and distance transform have been proven as general and efficient algorithms for tree (or star) structure models in bottom-up methods. However, scores have to be computed everywhere in the feature pyramid which is expensive and not necessary. We use a top-down method to search the best locations of the parts in the space covered by the high potential roots. The method is similar to [16] and [17], while both of them are based on latent SVM framework thus rely on the general distance transform and bottom up method for computing the score. Moreover, [16] requires additional training for learning the pruning threshold. Note that our method is independent to the distance function since root l_i is fixed. We use both Mahalanobis and Euclidean distance in our experiments and obtained equivalent results. The deformation cost of these two distance functions can be found in Fig. 3 (f) (g).

4) Combination model design: After obtaining multiview part-based detectors, we consider the problem of taking all of them into account for obtaining a final decision. Instead of using a straightforward way to combine all the scores of the individual detectors and to form an ensemble as in [8], we take into account the deformation penalty of each part in the final score. The combination model (ensemble) for root y_i is

$$\phi(I_i, y_i) = (s_0, s_1, ..., s_n, d_1, ..., d_n, 1) \ , \tag{7}$$

where s_0 is the score of the root, s_j, d_j are the score and deformation cost of part j, respectively. The last term is the bias for SVM. Therefore, the final model can be expressed as

$$\begin{array}{ll} \Phi(I_i) & = (\phi(I_i, 1), ..., \phi(I_i, m)) \ , \\ score(I_i) & = \beta \cdot \Phi(I_i) \ , \end{array} \tag{8}$$

where $\Phi(I_i)$ combines each $\phi(I_i, y_i)$ to form a multiple root vector, β is the parameter trained by SVM and $score(I_i)$ is the final score of example I_i.

C. VDPM Training

Now we consider the problem of training VDPM. As described above, we separate the training procedure into individual model and ensemble model training.

1) Training root and part detectors with SVM: We train the joint root model with multiview examples and random negatives obtained from the virtual-world scenarios without pedestrians. After the initial training, the classifiers are re-trained with a set of hard negatives collected by the initial classifier. The hard negatives are added to train a new root classifier which is commonly referred to as bootstrapping. During the training, the hard negatives will be saved and added to the training set for the parts, thus the root will share the negative features with the parts. Fig. 3 compares the models learnt by latent SVM V4.0 and by our method.

2) Adaptive Combination of Classifiers (ACC): The adaptive combination of classifiers has been seen in hierarchical classification structures and [8] used it for part-based human detection. The authors showed that applying ACC with linear SVM is an effective procedure for combining classifiers. Based on this conclusion, we adopt ACC to combine the output score of the part detectors as well as the deformation cost, *i.e.*, to build our ensemble.

IV. Experimental Results

Since we are interested in the ADAS field, in this section, we evaluate our method on Daimler [18] and Karlsruhe pedestrian benchmarks [19]. Recently, the results of the state-of-the-art pedestrian detection algorithms have been reported in [12], where a general evaluation method for pedestrian detection has been proposed. In that paper, latent SVM has shown the state-of-the-art results except for the method incorporating motion features. Here we do not include such motion features, thus, in order to compare our method with the state-of-the-art, we use such an evaluation framework for our experiment, as well as the latent SVM 4.0.

A. Experiment setup

For the training, we generated virtual-world dataset with frontal/back and left viewed pedestrians, containing 1000 examples per view, with canonical size of 64×128 pixels. We extract part examples from these dataset, thus we got 1000 positive examples with canonical size of 32×64 pixels for each of the five parts. For collecting negative examples we use 2000 virtual-world pedestrian-free images of 640×480 pixel resolution, *i.e.*, the same size as Daimler testing images. We use the same training data for latent SVM and our model. For the testing, we upscale the testing images to a factor to detect small pedestrians as in [12].

Both our method and latent SVM are using HOG-based features. The only difference is that our HOG applies anti-aliasing in scaling operations and latent SVM uses a 31 dimensional HOG [11] (claimed to perform better than standard HOG). In the experiments, we also compare with the holistic models as in [13]. The holistic models based on our HOG features have been trained on Daimler training dataset as well as on virtual-world dataset.

For Daimler pedestrian dataset, we focus on the images containing pedestrians which must be mandatorily detected. We use 974 images containing 1193 pedestrians and the 72 pixels as the minimal height of pedestrian in order to compare to the holistic model of [13].

Karlsruhe datasets contains bounding boxes for cars and pedestrians. We use the pedestrian testing set with 775 images. Furthermore, the dataset contains the orientations of each object, discretized into front, back, left, and right

classes for pedestrians. We consider front and back as one orientation in this paper. In order to evaluate under the same framework, we converted the groundtruth to the Caltech format. In particular, all groundtruth with label 'hard' are set to be ignored in Caltech groundtruth format since we only evaluate pedestrians over 50 pixels. The final groundtruth contains 1345 pedestrians.

B. Results

We assess detectors performance using per-image evaluation, which is highly recommended in [12] and also employed in [18]. In particular, we plot curves depicting the tradeoff between miss rate and the number of false positives per image (FPPI) in logarithmic scale. Fig. 4 top depicts different performance plots on Daimler dataset. For Karlsruhe pedestrian dataset, we further evaluated the performances at different scales (Fig. 4 middle). We evaluated detections on pedestrians with minimum height of 50 pixels, 80 pixels and 100 pixels. In Fig. 5, we show some qualitative results (detections and viewpoint estimation).

As expected, part-based methods show better performance than holistic ones. The "Holistic (Virtual)" achieves 48.6% of average miss rate, which is equivalent to the 49.4% obtained with the model trained by Daimler dataset (the "Holistic (Daimler)" in Fig. 4 top), and the result agrees with the conclusion in [13]. Our multiview root achieves 30.5% showing that the mixture models perform better than the single root. Training with the same virtual world-data, our VDPM achieves 24.9%, approximately 3 points better than the latent SVM 4.0 which reaches 27.9%.

The results on Karlsruhe dataset show that our model performs constantly better at all scales. Since we are using a mulitview model, we are also interested in the viewpoint estimation though viewpoint estimation is not the main focus of this paper. The viewpoint output can be obtained by tracing back the maximum score of the multiview classifiers. Fig. 5 row 3 shows the viewpoint estimation output by our model. We made quantized evaluation for both our multiview root model and multiview part-based model, following the proposal in [20]. We merged the front and back view groundtruth as one view since our model only contains 3 views. We tested on all 1345 samples and interestingly, our root model and part-based model got very similar performance. Results of our multiview part-based model are given in Fig. 4 bottom. Our model reaches up to 71%, 74% and 67% accuracy for front/back, left, and right views respectively. The overall correct decision rate is 70.4% per test sample.

Additionally, we assessed the time consumption on a desktop computer. Using the same virtual-world training data as we mentioned above, our VDPM is trained within 5 hours while the latent SVM consumed more than 6.5 hours. The entire testing time of latent SVM on 973 images of size 640×480 pixels is 3 hours and 44 minutes, while our VDPM only took 20 minutes, *i.e.*, 0.79 fps, thus our detector is ten times faster than latent SVM 4.0, which is close to the cascade model of [16]. Our method turns out to be simpler but effective in practice. Note that we are using 36

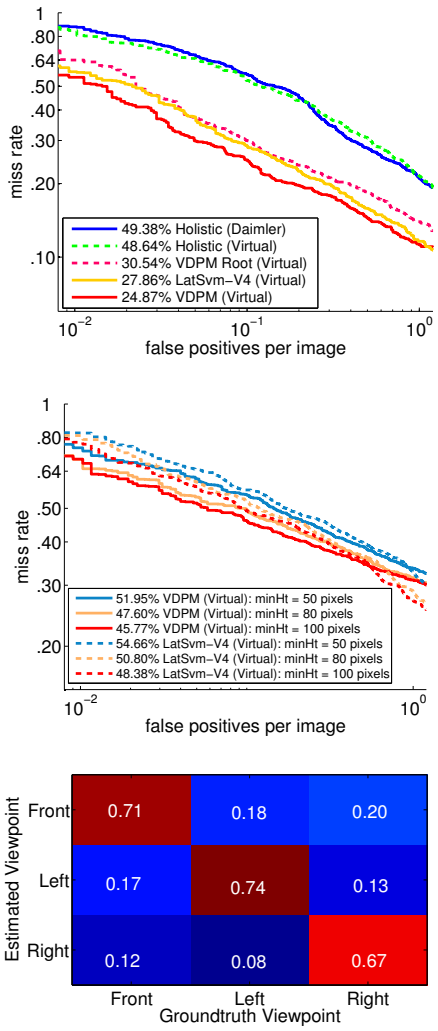

Fig. 4. Top: Average miss rate percentage on Daimler dataset. The training dataset is denoted in the parenthesis. Middle: Evaluation at different scales on Karlsruhe dataset. Bottom: Confusion matrix of viewpoint estimation on Karlsruhe.

dimensional HOG instead of a faster 31 dimensional HOG in latent SVM.

V. Conclusion

In this paper we have presented a method to train a multiview deformable part-based model with virtual-world data for pedestrian detection. The key point of our approach is the ability of automatically obtaining parts labels and view clusterization from the virtual-world pedestrians. This removes latent variables from our approach and allows us to design a very efficient pedestrian detector. In particular, we use a top-down approach to detect the parts which can largely reduce the searching space comparing to the bottom-up approach. So far, our detector is ten times faster than latent SVM 4.0.

A possible extension of our method is to take more advantages of the 3D virtual world, as for example using 3D pedestrian skeleton points for even more precise part sampling. Furthermore, these informative points can be used

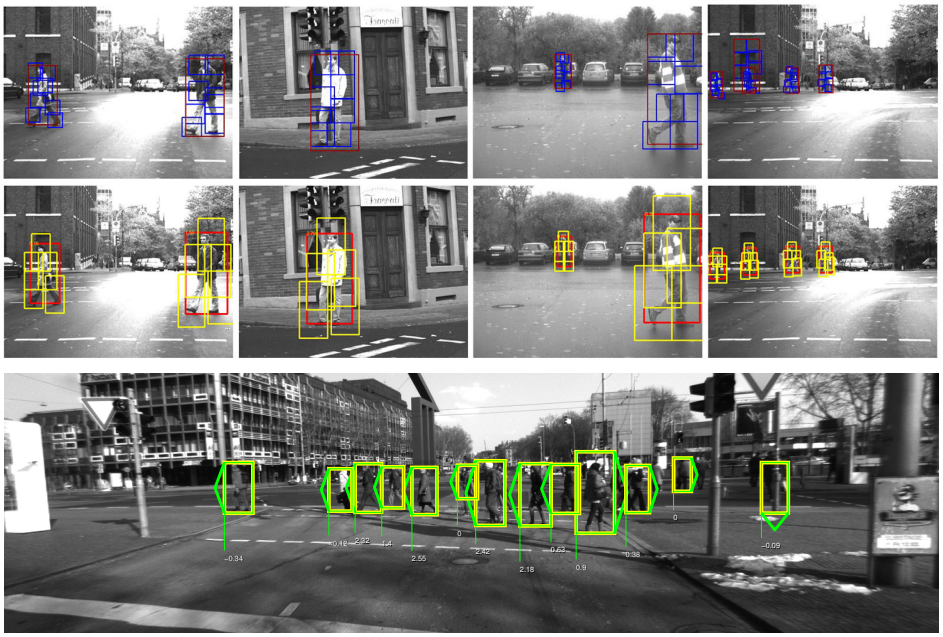

Fig. 5. Detections on Daimler and Karlsruhe pedestrian datasets. Row 1: detections of latent SVM V4 on Daimler. Row 2: detections of our model on Daimler. Row 3: the viewpoint output on Karlsruhe.

to learn a more exact body configuration which can be used to train a richer model. The detection speed and pedestrian direction estimation can also be further improved in our future work. For example, geometry constrains [21] or even stixels [22], [23] can be used to reduce the sliding window candidates, thus the speed is expected to be substantially improved.

ACKNOWLEDGMENT

This work was supported by the Spanish projects TRA2011-29454-C03-01 and TIN2011-29494-C03-02, as well as the Chinese Scholarship Council (CSC) grant NO.2011611023.

REFERENCES

[1] D. Gerónimo, A. López, A. Sappa, and T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 7, pp. 1239–1258, 2010.

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

[3] S. Munder and D. M. Gavrila, "An experimental study on pedestrian classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 28, pp. 1863–8, 2006.

[4] M. Enzweiler and D. M. Gavrila, "A mixed generative-discriminative framework for pedestrian classification," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2008, pp. 1–8.

[5] S. Munder, C. Schnorr, and D. Gavrila, "Pedestrian detection and tracking using a mixture of view-based shapetexture models," *IEEE Trans. on Intelligent Transportation Systems*, pp. 333–343, 2008.

[6] K. Goto, K. Kidono, Y. Kimura, and T. Naito, "Pedestrian detection and direction estimation by cascade detector with multi-classifiers utilizing feature interaction descriptor," in *IEEE Intelligent Vehicles Symp.*, 2011, pp. 224–229.

[7] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: single-frame classification and system level performance," in *IEEE Intelligent Vehicles Symp.*, 2004, pp. 1–6.

[8] A. Mohan and C. Papageorgiou, "Example-based object detection in images by components," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 4, pp. 349–361, 2001.

[9] L. Bourdev, "Poselets: Body part detectors trained using 3D human pose annotations," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[10] P. Felzenszwalb and D. P. Huttenlocher, "Pictorial Structures for Object Recognition," *Int. Journal on Computer Vision* , vol. 61, no. 1, pp. 55–79, 2005.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 32, no. 9, pp. 1627–45, 2010.

[12] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 1–20, 2011.

[13] J. Marin, D. Vázquez, D. Gerónimo, and A. López, "Learning appearance in virtual scenarios for pedestrian detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010.

[14] C. Hou and H. Ai, "Multiview pedestrian detection based on vector boosting," in *Asian Conf. on Computer Vision*, 2007, pp. 210–219.

[15] D. Park and D. Ramanan, "Multiresolution models for object detection," in *European Conf. on Computer Vision*, 2010, pp. 1–14.

[16] P. Felzenszwalb and R. Girshick, "Cascade object detection with deformable part models," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[17] M. Pedersoli, A. Vedaldi, and J. González, "A coarse-to-fine approach for fast deformable object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[18] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection survey and experiments," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 12, pp. 2179–95, 2009.

[19] A. Geiger, C. Wojek, and R. Urtasun, "Joint 3d estimation of objects and scene layout," in *Advances in Neural Information Processing Systems*, Granada, Spain, December 2011.

[20] M. Enzweiler and D. M. Gavrila, "Integrated pedestrian classification and orientation estimation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2010, pp. 982–989.

[21] H. Cho, P. E. Rybski, A. Bar-Hillel, and W. Zhang, "Real-time pedestrian detection with deformable part models," in *IEEE Intelligent Vehicles Symp.*, 2012, pp. 1035–1042.

[22] M. Enzweiler, M. Hummel, D. Pfeiffer, and U. Franke, "Efficient stixel-based object recognition," in *2012 IEEE Intelligent Vehicles Symposium*, 2012, pp. 1066–1071.

[23] R. Benenson and M. Mathias, "Pedestrian detection at 100 frames per second," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2012.