Learning a Part-based Pedestrian Detector in Virtual World

Jiaolong Xu, David Vázquez, Antonio M. López *Member, IEEE,* Javier Marín and Daniel Ponsa

***Abstract*—Detecting pedestrians with on-board vision systems is of paramount interest for assisting drivers to prevent vehicle-to-pedestrian accidents. The core of a pedestrian detector is its classification module, which aims at deciding if a given image window contains a pedestrian. Given the difficulty of this task, many classifiers have been proposed during the last fifteen years. Among them, the so-called (deformable) part-based classifiers including multi-view modeling are usually top ranked in accuracy. Training such classifiers is not trivial since a proper aspect clustering and spatial part alignment of the pedestrian training samples are crucial for obtaining an accurate classifier. In this paper, first we perform automatic aspect clustering and part alignment by using virtual-world pedestrians, *i.e.*, human annotations are not required. Second, we use a mixture-of-parts approach that allows part sharing among different aspects. Third, these proposals are integrated in a learning framework which also allows to incorporate real-world training data to perform domain adaptation between virtual- and real-world cameras. Overall, the obtained results on four popular on-board datasets show that our proposal clearly outperforms the state-of-the-art deformable part-based detector known as latent SVM.**

***Index Terms*—computer vision, pedestrian detection, synthetic training data, multi-part model**

I. Introduction

ON-BOARD pedestrian detection is crucial to prevent accidents. Vision-based detectors consist of several processing stages [1], [2], namely the generation of image candidate windows, their classification as *pedestrian* or *background*, the refinement into a single detection of multiple ones arising from the same pedestrian, and the tracking of the detections for removing spurious ones or inferring trajectory information.

An accurate classification is fundamental. However, it turns out to be a difficult task due to the large intra-class variability of both pedestrians and background classes, as well as the imaging and environmental conditions. Note that pedestrians are moving objects which vary on morphology, pose, and clothes; there is a large diversity of scenarios; and images are acquired from a platform moving outdoors (*i.e.*, the vehicle), thus, pedestrians are seen from different viewpoints at a range of distances and under uncontrolled illumination.

Aiming at overcoming such a complexity, many pedestrian classifiers/detectors have been proposed during the last fifteen years. The reader is referred to [2] for a comprehensive review on pedestrian detection, to [1], [3] for accuracy comparisons of different proposals, as well as to [4], [5] where the focus is on reaching real-time processing. A first outcome of the work done so far in this field is that most accurate pedestrian classifiers are learned from pedestrian and background samples. For instance, this is the case of the well-known pedestrian classifier based on histograms of oriented gradients and linear support vector machines (HOG/Lin-SVM) [6].

Indeed, HOG/Lin-SVM approach was a milestone in the field of pedestrian detection. However, the most relevant contribution of [6] consists in devising HOG features, since the overall pedestrian classifier itself just follows a *holistic* approach and uses a linear frontier to separate pedestrians and background. Holistic approaches regard pedestrians as a whole, *i.e.*, no body-inspired parts are considered separately. Moreover, [6] proposes what we term as *single* holistic approach because the intra-class variability of the pedestrians is not explicitly considered. In other words, during the training of the pedestrian classifier all pedestrians are mixed, which tends to generate blurred features. In consequence, the learned classifier does not necessarily improves its accuracy by increasing and/or diversifying the training pedestrian samples [7].

In order to overcome this limitation, prior knowledge about the pedestrian class can be exploited. For instance, we can find multiple holistic ensembles accounting for different pedestrian view and pose combinations (*aspects* hereinafter), or single/multiple body-inspired part-based ensembles. Representative examples can be found in [4], [8]–[13]. In fact, the *deformable part-based model* (DPM) presented in [10] is one of the most popular state-of-the-art pedestrian/object detectors.

An advantage of DPMs is that pedestrian poses unseen during training are implicitly modeled through the allowed deformation, *i.e.*, the generalization capability of the corresponding classifiers increases. This is more effective if view-based DPMs can be used to build a mixture model, which is the case in [10] provided that the aspect ratio of the annotated pedestrian bounding boxes (BBs) correlates with major view differences (*e.g.*, frontal *vs.* side). A natural extension of this idea consists in allowing to share parts among different views, which increases the number of implicitly modeled aspects and reduces the number of overall parts to be learned and applied. Up to the best of our knowledge this approach has not been exploited in pedestrian detection for driver assistance. However, part-sharing has recently shown benefits in tasks such as object detection and pose estimation [7], [14]–[16].

Accordingly, for on-board pedestrian detection this paper proposes a new aspect-based mixture of DPMs with part-sharing. A key point of such a pedestrian model is to have pedestrian samples with reliable and rich annotations. In particular, for each pedestrian, its full-body BB is required

J. Xu, D. Vázquez, A.M. López, J. Marín and D. Ponsa are with the Departament de Ciències de la Computació and the Computer Vision Center (CVC), Universitat Autònoma de Barcelona, Spain. E-mail: jiaolong@cvc.uab.es

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

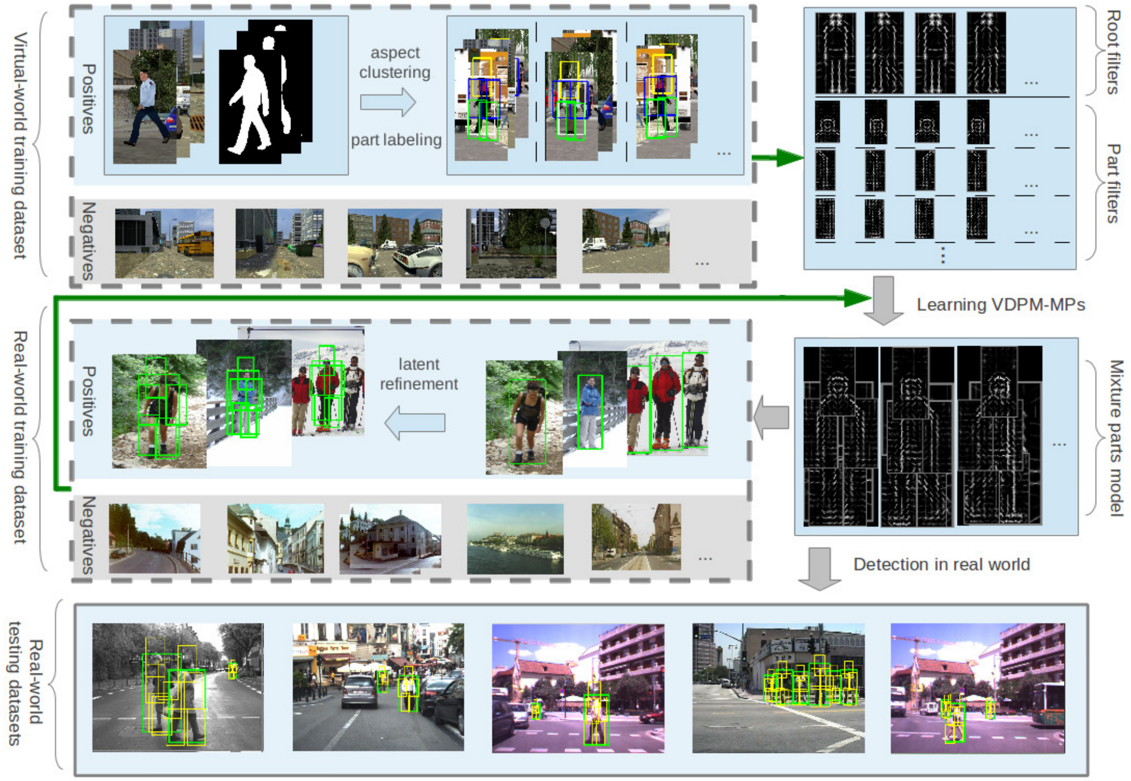

Fig. 1. Virtual world trained DPM with Mixture of Parts (VDPM-MP) framework for training an aspect-based mixture of DPMs with part-sharing. We refer to Algorithms 1 and 2 (Sect. IV) for more details.

along with the BB of its constituent parts, and its aspect label (*e.g.*, either rear-frontal, side-left, side-right). Collecting all this information by human annotation is a tiresome task prone to errors. Thus, other than [7], [8], [14], [16]–[22], we propose the use of a virtual-world with automatic pixel-wise pedestrian groundtruth. In our first work in this line [23] a single holistic pedestrian classifier trained with virtual-world data performed equally well in automotive real-world images than an equivalent one trained with real-world data. For building our pedestrian model, in this paper we also exploit part labeling (*i.e.*, part BBs) and aspect clustering, both automatically obtained from the pixel-wise groundtruth.

In the last years the computer vision community has started to consider the decrease in accuracy of a classifier due to differences between training and testing data. In [24], [25], we show that between virtual- and real-world data this problem exists. However, we show also that it is not due to the particular difference between virtual- and real-world imaging but just because this phenomenon can appear between any two camera types, even if both operate in the real world. Moreover, we show how fusing virtual-world training data with a relatively few real-world training data allows to adapt virtual and real domains. While looking for the best *domain adaptation* method for our classifiers is out of the scope of this paper, we have devised our learning framework to allow such a world's fusion and we demonstrate its effectiveness too. For that we only require the full-body BB of the real-world pedestrians, *i.e.* neither their part BBs nor aspect labels.

Fig.1 summarizes our DPM-based proposal. Since we rely on virtual-world data and part-sharing is implemented as a mixture of parts, we term this proposal as VDPM-MP. We test it on four popular on-board datasets focusing on luminance images and HOG features. The results show that VDPM-MP outperforms the state-of-the-art DPM proposed in [10] based on HOG-inspired features and latent SVM (HOG/Lat-SVM).

In Sect. II we summarize the works most related to this paper. Sect. III explains the generation of virtual-world training samples with aspect clustering and part labeling. Sect. IV details the proposed framework for training pedestrian classifiers. Sect. V presents the qualitative evaluation of our proposal, assessing the contribution of the different ideas involved in it. Finally, the conclusions are drawn in Sect. VI.

II. Related Work

A major benefit of using prior knowledge strategies (*i.e.*, richer annotations) to *align* the pedestrian training samples is to obtain less blurred features and more accurate overall pedestrian classifiers as a consequence. For instance, aspect clustering avoids to mix frontal/rear viewed pedestrians with side viewed ones during training. DPMs avoid to mix different body parts. Aspect clustering plus fixed part models may have the same effect as DPMs, however, the capacity of modeling unseen poses may be reduced and then more pedestrians examples could be required for training.

In [8], a body-inspired part-based method is proposed and, notably, aspect-based clustering of the training data is also reported as a crucial step. In particular, an AdaBoost strong classifier ensembles 117 weak classifiers that account for different body parts and clusters. There are nine clusters and within each cluster thirteen fixed parts with overlap (head, trunk, head-trunk, trunk-legs, etc.) are considered, each part being described by a sort of simplified HOG features. In [9], a fixed part-inspired model is also used but without aspect clustering; the focus is on the features.

The state-of-the-art DPM-based pedestrian detector relies on the HOG/Lat-SVM framework [10]. In this approach there is a holistic filter called *root* and few body-part filters operating at twice the resolution of the root. The parts are not assumed to be fixed, but they are distributed according to a deformable star layout anchored to the root. In the current version of HOG/Lat-SVM (*i.e.*, V5) [26] it is possible to handle different models based on the root aspect ratio, called *components*. Each component has its own associated filters and layout.

In [11], a probabilistic mixture-of-experts framework is proposed where 24 different holistic classifiers (experts) account for four different pedestrian views (front, back, left, right) and three modalities (luminance, depth, optical flow), described by HOG and local binary patterns (LBP) features.

In [8] view information is manually provided (*how-to* details are not given), while in [11] aspect information is provided by a shape hierarchy built from manually delineated silhouettes. Note that manual delineation of silhouettes is a tiresome task. Thus, for reducing the annotation effort, in [10], [15], [16] an automatic aspect clustering criterion is used, namely, the aspect ratio of the manually annotated pedestrian BBs. However, this is a crude criterion that roughly allows to distinguish two view categories, namely frontal/rear *vs.* left/right, and only if the use of this criterion was taken into account during BBs annotation. Some works also incorporate clustering based on the features further used to describe the object, *e.g.*, HOG [7]. However, this implies clustering in high dimensional spaces, especially when combining different feature types, whereas it may lead to cluster-assignment inconsistencies in multi-expert approaches provided the feature type determines the expert. Moreover, in these cases clustering does not depend only on the objects of interest (*i.e.*, pedestrians here) but also in the background clutter. In short, for such non-human guided approaches clustering can become a difficult issue itself.

In [10], given the pedestrian BBs clustered by aspect, part alignment is automatically done per-view through the iterative mechanism of Lat-SVM training. However, it is necessary to provide a heuristic part initialization which may lead to a sub-optimum alignment. In fact, some works rely on manual part-level supervision as an effective method for obtaining more accurate DPMs [7], [14], [16], [18]–[21]. Of course, manually supporting part annotations is also a tiresome task.

In order to increase the size and variability on the training set, some works propose the application of geometric and photometric transformations to pedestrian examples manually annotated. Since such transformations require to have the pedestrian examples segmented, different pedestrian-background combinations are also generated for training. For instance, in [17] it is assumed the manual delineation of the initial silhouettes of the pedestrians. In [22], landmark points for 3D pose recovery and a semi-automatic segmentation of the 2D pedestrian examples need to be manually provided.

In comparison with the reviewed literature, the contributions of our VDPM-MP method are summarized as follows. DPM employs a coordinate-descent algorithm to learn the model parameters which is sensitive to initialization. However, given the difficulties of collecting rich pedestrian/object annotations by manual labeling, heuristic methods are used to initialize the parameters defining parts and views. VDPM-MP framework uses virtual-world pedestrians with automatically generated groundtruth regarding parts and views, *i.e.*, a richer pedestrian information than just BBs, which is mandatory to learn the VDPM-MP classifier. Therefore, allowing a better initialization of the learning process. Moreover, once a VDPM-MP is trained with virtual data, the resulting classifier can be refined for domain adaptation using real-world data, where only the full-body BB of each real-world pedestrian is required. In addition, following the spirit of [14], VDPM-MP is a more flexible DPM than the originally proposed in [10] since it allows part-sharing. As we will see in Sect. V, these two improvements (*i.e.*, better parameter initialization and part-sharing) allow VDPM-MP to significantly improve the accuracy of the original DPM without requiring to collect part and aspect information by manual labeling.

III. Virtual-World Training Data

A. Virtual-World Images

For this work we have improved the dataset of [23] using the same proprietary game engine (*i.e.*, Half-Life 2). The new images contain higher quality textures and more variability in cars, buildings, trees, pedestrians, etc. Unfortunately, we have no access to the 3D information processed by the game engine. However, a precise 2D segmentation (pixel-wise groundtruth) of the imaged pedestrians is automatically available. Hence, for automatically obtaining BBs, performing aspect clustering and part labeling, we process the 2D pedestrian-segmentation masks as explained in III-B and III-C. Therefore, our mechanism can also be used when manually drawn object silhouettes are available (*e.g.*, as in [17]).

B. Aspect Clustering

The silhouette of the pedestrians can be used to distinguish major aspect tendencies. The available segmentation of the virtual-world pedestrians allows to automatically delineate their precise silhouette. Thus, using a similarity function between silhouettes we can cluster them. A function that does not require point-wise matching between silhouettes is chamfer distance, which has already been successfully used for building shape-based pedestrian hierarchies from manually annotated silhouettes [27]. Given a binary template T and a binary image I, the T to I chamfer distance is defined as $Ch(T, I) = |T|^{-1} \sum_{t \in T} \min_{i \in I} \|t - i\|$, where $|T|$ denotes the area of T. In our case, both T and I are silhouettes. Since $Ch(T, I)$ is not a symmetric function in general, we use the symmetric version $S(X, Y) = Ch(X, Y) + Ch(Y, X)$.

Using $S(X, Y)$ we build a similarity distance matrix, $M(X, Y)$, for the silhouettes. Then, we can organize the pedestrians as a silhouette-based hierarchical cluster by relying on $M(X,Y)$ and K-medoids [28]. K-medoids selects a data point for each cluster center, which is important here since we will further use the *center pedestrians* for part labeling.

First, pedestrian BBs are automatically determined from the segmentation masks. The BBs are set with the same aspect ratio as the *canonical (detection) window* (CW). We crop pedestrians and masks according to the BBs. Then, all cropped windows (appearance and mask) are resized to the CW size.

Second, we exploit vertical symmetry to obtain an initial *alignment* of the pedestrians. In particular, we manually select one left-side viewed pedestrian, which is vertically mirrored to obtain its right-side counterpart (its mask is also mirrored). These two exemplars initialize K-medoids clustering for K=2. This procedure classifies our pedestrians as either left or right aspect. Frontal/rear aspects are assigned to one or another category depending on their aspect tendency. Now, the pedestrians classified as right-aspect are vertically mirrored and joined with the other category. Thus, we obtain a training set of pedestrians that are aspect-aligned in the left-*vs.*-right sense. Regarding the hierarchical clustering, this set of pedestrians constitutes the root level, *i.e.*, no clusters are available yet.

Third, we perform the hierarchical clustering. In particular, we generate a binary tree by iteratively applying K-medoids with K=2 and using $M(X,Y)$. In this case, K-medoids initialization is done just randomly. For instance, the first application of the procedure (2nd level of the hierarchy) divides the pedestrian examples of the root level as frontal/rear-*vs.*-left categories. The second application (3rd level) distinguishes different degrees of left skewness, and so on. Fig.2a and 2b show the average appearance and mask of pedestrians for the 2nd and 3rd levels of the hierarchy, as well as the *mirrored hierarchy* generated by vertically mirroring the pedestrian examples at each node of the binary tree.

C. Part Labeling

We assume the usual settings of the state-of-the-art part-based models [8], [10], [12], *i.e.*, a fixed number of parts annotated as rectangular sub-windows, where each part rectangle is of fixed size but where such size can vary from part to part. In the deformable case (DPM) the location of the parts changes from one pedestrian example to another. Since we focus on DPMs, we have to provide a procedure to automatically label the parts for each example. Currently we follow the hierarchical cluster described in III-B.

In particular, we select the pedestrian masks representative of the 2nd level clusters, *i.e.*, one exemplar for the frontal/rear aspect and another for the left one. We manually point the parts' centers of these two exemplars. For instance, we can roughly focus on head and extremities, *i.e.*, five parts, and then quickly clicking ten pixels to be these centers. The parts' centers are automatically propagated through the hierarchy, from the 2nd level to the bottom level. From level to level, the centers are propagated between the representatives of cluster nodes. The representatives of the bottom-level clusters propagate the centers to all the pedestrian examples within their respective clusters. Overall, by manually clicking ten points, we can obtain part labeling for thousands of pedestrians.

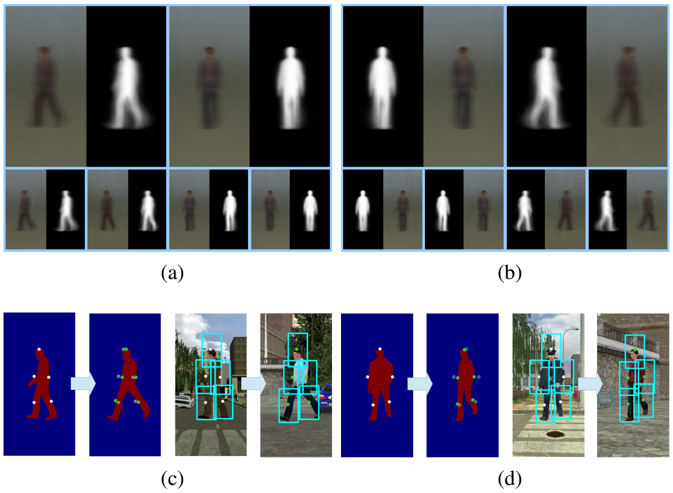
(a)
(b)
(c)
(d)

Fig. 2. (a) 2nd (top row) and 3rd (bottom row) levels of the silhouette-based hierarchy. The average appearance and segmentation mask of each cluster node are shown. (b) *Mirrored hierarchy*. (c) Part centers marked in the pedestrian mask representative of the left 2nd level cluster, and their automatic propagation to the representative of one of the left subcategories (a 3rd level cluster node). The respective parts are also shown on the corresponding appearance windows. (d) Analogous for the frontal/rear case.

Propagating the centers from one pedestrian example e_1 to another e_2 is done by a simple but effective procedure. For that we use the distance transform (DT) of the different examples. Since chamfer distance involves DT computation, all pedestrian DTs are already available from the hierarchical clustering. Let c_i^p be the center of the part p of the pedestrian example e_i, and D_j the DT of the example e_j. In order to map c_1^p into c_2^p, the new center c_2^p is defined as the silhouette pixel of e_2 which is at minimum distance $D_2(c_1^p)$ from c_1^p. If the condition holds for more than one pixel, we just choose one at random since they must be quite close and thus the choice will not affect the final pedestrian model. Fig.2c and 2d illustrate the idea. Note that, like for the hierarchical cluster, here we can define also the *vertically mirrored parts*.

IV. Pedestrian Classifier

A. Aspect-based Mixture of DPMs: Overall Idea

In this paper, pedestrians are modeled according to their full-body appearance, as well as by the appearance of n body-inspired parts. Such appearances are evaluated by corresponding learned image filters. The size of these filters can be different from part to part. However, each individual filter size is fixed. Contrarily, the location of the parts can vary with respect to the overall full-body location. There are part locations more plausible than others, therefore, there is a deformation penalty given by a deformation cost function. Overall, this is the description of a deformable part-based model (DPM). Moreover, in order to search for pedestrians at multiple resolutions, a pyramidal sliding window is assumed and, following [10], we also assume that parts are detected at twice the resolution of the root.

In addition, we consider different aspect models, thus, our pedestrian model actually is a *mixture model* of m components

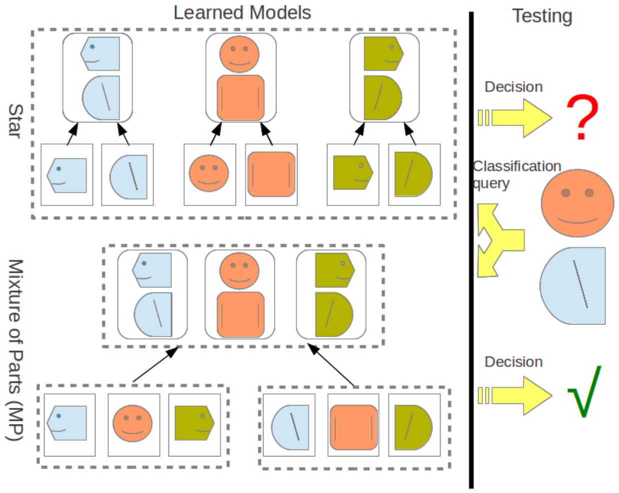

Fig. 3. Part-sharing allows to model unseen aspects. Imagine the testing sample at the right was not present in the training data. Then, the star model does not include a combination of parts (head/trunk here) corresponding to this testing sample. The MP models part combinations that were not seen during training, thus, it has more chances to rightly classify such a sample.

(aspect-based mixture of DPMs). When using more than one component we have to decide whether to share parts among components or not (Fig. 3). In [10] parts are not shared among components, which corresponds to a star structure. Not sharing parts can lead to a large number of them, while sharing the parts reduces this number and allows to model aspect configurations not explicitly seen during training time. Part-sharing has been successfully used for pose estimation [14] and to share parts among different classes of objects [16]; manual part annotations are required in these works though.

B. DPM Definition

For describing our pedestrian model we mainly follow the notation of [10] since this is the state-of-the-art multi-component DPM which we take as baseline. We call H the pyramid of features built from the image under consideration. Let $p = (x, y, l)$ specify a position (x, y) in the l-th level of H. For instance, if H is a pyramid with HOG information, then $H(p)$ contains the features corresponding to a cell of HOG. We term as $\phi_a(H, p, w, h)$ the vector obtained by concatenating the feature vectors of a $w \times h$ subwindow of H with top-left corner at p in row-major order. Let F be a $w \times h$ filter, arranged as a vector, *i.e.*, as for the subwindows of H. We can compute the score of F at p as $F \cdot \phi_a(p)$, where hereinafter we have simplified the notation by assuming equal dimension of filters and subwindows, and the use of H underlying appearance features computation (for the deformation features defined later too). Following with the HOG example, note that each entry of F actually contains a vector of weights for the bins of the four histograms of a HOG cell. In other words, if we think in terms of the traditional HOG/Lin-SVM framework, the filter F contains the weights learned by the Lin-SVM procedure.

A DPM is then defined by a $(n + 2)$-tuple $(F_0, P_1, \ldots, P_n, b)$, where F_0 is the root filter of size $w_0 \times h_0$, P_i describes part i, and b is a real-valued bias term. In particular, $P_i = (F_i, v_i, d_i)$, where F_i is the filter of part i with size $w_i \times h_i$, v_i is a 2D vector that describes the relative *anchor* position of part i with respect to the root position, and d_i is a 4D vector of coefficients of a quadratic function, $\phi_d(dx, dy) = (dx, dy, dx^2, dy^2)$, that defines the cost of deviating from the anchor position (*i.e.*, the deformation cost).

Now, let $z = (p_0, \ldots, p_n)$ be a pedestrian hypothesis, *i.e.*, an assumption about where the root and the n parts are located within H, subject to $l_i = l_0 - \lambda$ for $i > 0$, where λ defines the number of levels needed to double the resolution. This hypothesis will be validated (*it is pedestrian*) or rejected by thresholding a score, say $S(z)$, which accounts for the scores of the appearance filters in their respective positions as well as the deformation cost of each part, plus the bias term, *i.e.*,

$$S(z) = \sum_{i=0}^{n} F_i \cdot \phi_a(p_i) - \sum_{i=1}^{n} d_i \cdot \phi_d(dx_i, dy_i) + b \ , \quad (1)$$

where (dx_i, dy_i) is the displacement of the i-th part relative to its anchor point. We can express (1) in compact form as

$$S(z) = \beta \cdot \psi(z) \ , \quad (2)$$

where β is a vector of model parameters and $\psi(z)$ is a vector with the appearance and deformation of hypothesis z as observed in H, *i.e.*,

$$\begin{aligned} \beta &= (F_0, \ldots, F_n, d_1, \ldots, d_n, b) \ , & (3) \\ \psi(z) &= (\phi_a(p_0), \ldots, \phi_a(p_n), & (4) \\ & \quad -\phi_d(dx_1, dy_1), \ldots, -\phi_d(dx_n, dy_n), 1) \ . & \end{aligned}$$

Based on $S(z)$ we can follow [10] to apply an efficient pedestrian search within an input image.

C. DPM Learning

The parameters in β must be learned from a set of labeled samples. For Lin-SVM learning with hinge loss, β can be obtained by solving the following optimization problem:

$$\begin{aligned} &\min_\beta \frac{1}{2} \parallel \beta \parallel^2 + C \textstyle\sum_{i=1}^{N} \xi_i \ , \\ &\text{s.t. } \ \forall i \in [1..N] : \xi_i \geq 0 \wedge (\beta \cdot \psi(s_i))\ell_i \geq 1 - \xi_i \ , \end{aligned} \quad (5)$$

where, $\{(s_1, \ell_1), \ldots, (s_N, \ell_N)\}$ is the set of training samples, with $\ell_i \in \{+1, -1\}$ labeling sample s_i as pedestrian $(+1)$ or background (-1). Here we assume $s_i = (p_{0,i}, \ldots, p_{n,i})$, *i.e.*, when $\ell_i = +1$ both the BBs of the root and parts of pedestrian i are provided (see Sect. III-C), while for $\ell_i = -1$ such BBs can be just sampled from background patches. Moreover, note that s_i is expressed with respect to the coordinates of the pyramid of features, H_{s_i}, computed from the original image containing the annotated pedestrian corresponding to s_i. These coordinates already encode resizing pedestrians to CW size.

If only pedestrian root BBs are annotated but not part BBs, this optimization problem is not convex and the Lat-SVM algorithm must be applied by treating root and part BBs as latent information [10]. Lat-SVM is basically a coordinate descent method where holding β fixed, the root and part BBs are optimized (manually annotated root BBs and ad hoc part BBs relative to such root BBs are used for initialization); then assuming that such BBs are right, β is optimized.

D. Mixture of DPMs: Star Model (VDPM-Star)

The basic DPM can be extended to account for m components (*e.g.*, *views*), *i.e.*, the new model can be thought as a mixture of DPMs. Each component has its associated β_c and $\psi_c(z)$ vectors, $1 \leq c \leq m$. The score function in this case can be defined as $S(z) = \max_{c\in[1..m]} \beta_c \cdot \psi_c(z)$. Now, the parameters to be learned take the form of a vector

$$\beta = (\beta_1, \ldots, \beta_m) \ . \tag{6}$$

Again, β can be obtained by solving the optimization problem in (5), where in this case

$$\psi(z') = (0, \ldots, 0, \psi_c(z), 0, \ldots, 0) \tag{7}$$

for $z' = (c, z)$. Note that $\psi(z')$ is a sparse vector, *i.e.*, all its entries are zero but those of the component c corresponding to z'. Accordingly, the training samples are of the form (s'_i, ℓ_i) with $s'_i = (c_i, s_i)$. Note that in our case the components are aspects and, thus, during training the aspect information (*i.e.*, c_i) of the pedestrian samples is known (see Sect. III-B), while for the background ones it can be set randomly. This mixture of DPMs just inherits the star structure of the basic DPM for each mixture component. Therefore, and given the fact that we rely on virtual-world data, we term it as VDPM-Star.

As in the single component case, given an image we use the new score $S(z)$ for finding pedestrians following [10].

E. Mixtures-of-Parts Model (VDPM-MP)

The star structure limits the parts to be connected to a single root component. Therefore, sharing parts among different components is not possible. Moreover, when increasing the number of components, the number of part filters grows accordingly. In contrast, models allowing part-sharing can avoid both problems. We follow the mixtures-of-parts (MP) idea presented in [14] for pose estimation, which is based on a tree organization. In particular, a node of the tree conveys different aspects of the same type of part, *e.g.*, one node can include different head aspects, another node can incorporate different trunk aspects, and so on. Moreover, there is a deformation cost between part aspects of child and father nodes.

In this paper we incorporate several contributions with respect to [14]. First, rather than using just a collection of constituent parts, we also use a root which is treated as a special part. Second, in this case we also detect parts at twice the resolution of the root. Third, as in the DPM seen so far, the deformation cost of all the parts are with respect to the root. Thus, our model is a tree with only two layers. In the first layer (root node of the tree) we have different pedestrian root aspects. In the second layer, we have different nodes, each one being dedicated to a different type of part (*i.e.*, view induced head aspects, left-arm aspects, etc.). Fig.3 conceptualizes the idea. Note how any part aspect can be combined with any root aspect. Thus, the variety of modeled pedestrians that were not explicitly seen during training is larger than for star-like models [7], while increasing the number of aspects of a given part (*e.g.*, root aspects) does not require doing the same for the other parts.

Interestingly, by defining the proper β and ψ vectors, the learning of β drives us to (5) again. First, we define

$$\beta = (\Gamma_0, \ldots, \Gamma_n, \Delta_1, \ldots, \Delta_n, b) \ , \tag{8}$$

where

$$\Gamma_i = (F_i^1, \ldots, F_i^k), 0 \leq i \leq n \ , \tag{9}$$

conveys the appearance filters of part i ($i = 0$ refers to the root) for k aspects, while

$$\Delta_i = (d_i^{1,1}, \ldots, d_i^{1,k}, \ldots, d_i^{k,1}, \ldots, d_i^{k,k}), 1 \leq i \leq n \ , \tag{10}$$

are the deformation cost parameters of part i with respect to the root, where $d_i^{a_i,a_0}, 1 \leq a_i, a_0 \leq k$, stands for the deformation cost parameters of the aspect a_i of part i with respect to the aspect a_0 of the root. We note that, without losing generality, in this work we use the same number of aspects (*i.e.*, k) for each type of part provided it is relatively low (*e.g.*, four in the experiments of Sect. V), otherwise it is straightforward to consider different k_i.

Accordingly, we can define the feature vector $\psi(z)$ as

$$\begin{aligned} \psi(z) \ &= \ (\Phi_a(p_0), \ldots, \Phi_a(p_n), \\ &\quad -\Phi_d(\delta x_1, \delta y_1), \ldots, -\Phi_d(\delta x_n, \delta y_n), 1) \ , \end{aligned} \tag{11}$$

where

$$\Phi_a(p_i) = (\phi_a(p_i^1), \ldots, \phi_a(p_i^k)), 0 \leq i \leq n \ , \tag{12}$$

contains the appearance features at $p_i^{a_i}$, $1 \leq a_i \leq k$, *i.e.*, the location p_i for the different aspects a_i. Now, we define the vector $\delta x_i = (dx_i^{1,1}, \ldots, dx_i^{1,k}, \ldots, dx_i^{k,1}, \ldots, dx_i^{k,k})$ and analogously for δy_i, where $(dx_i^{a_i,a_0}, dy_i^{a_i,a_0}), 1 \leq a_i, a_0 \leq k$, stands for the displacement of aspect a_i of part i with respect to aspect a_0 of the root. Accordingly, we have

$$\begin{aligned} \Phi_d(\delta x_i, \delta y_i) = (&\phi_d(dx_i^{1,1}, dy_i^{1,1}), \ldots, \phi_d(dx_i^{1,k}, dy_i^{1,k}), \\ &\ldots, \phi_d(dx_i^{k,1}, dy_i^{k,1}), \ldots, \phi_d(dx_i^{k,k}, dy_i^{k,k})), \\ &1 \leq i \leq n \ . \end{aligned} \tag{13}$$

Again, training samples are of the form $s'_i = (c_i, s_i)$. The c_i label is used as aspect index. When forming the $\psi(s'_i)$ vectors for the optimization in (5), all appearance related entries are zero but those indexed by aspect index c_i. Regarding the deformation cost entries, the situation is analogous but taking into account that the displacement of each part of s'_i must be related to all roots, not only to the root whose aspect is indexed by c_i. Note that displacements from any aspect of any part to any root aspect can be computed because during the training all the examples are used according to the CW size. Accordingly, we obtain feature vectors of the form

$$\begin{aligned} \psi(s'_i) \ &= \ (0, \ldots, \phi_a(p_0^{c_i}), \ldots, 0, \ldots, \phi_a(p_n^{c_i}), \ldots, 0, \\ &\quad 0, \ldots, \phi_d(dx_1^{c_i,1}, dy_1^{c_i,1}), \ldots, \phi_d(dx_n^{c_i,1}, dy_n^{c_i,1}), \\ &\quad \ldots, \phi_d(dx_1^{c_i,k}, dy_1^{c_i,k}), \ldots, \phi_d(dx_n^{c_i,k}, dy_n^{c_i,k}), \\ &\quad \ldots, 0, 1) \ . \end{aligned} \tag{14}$$

We remark that in this case the annotation of the parts and the aspects is strictly necessary. In [14] a manual process is followed to obtain such a rich ground truth, while here we use

Algorithm 1 VDPM Optimization

```
*Assume $\beta$ and $\psi$ defined by* (6) *and* (7) *for VDPM-Star, or by* (8) *and* (14) *for VDPM-MP. Inputs $\mathcal{S}^+$ and $\mathcal{S}^-$ stand for positive and negative training data, respectively, while $\mathrm{D}_{in}$ is an initial pedestrian detector.*
**Optimize**($\mathcal{S}^+, \mathcal{S}^-, \mathrm{D}_{in}$)
$\mathrm{D}_{out} \leftarrow \mathrm{D}_{in}$
**while** the optimization does not finish **do**
    **1.** Compute the $\psi$'s as follows:
        **1.a.** Run **Detect**($\mathrm{D}_{out}, \mathcal{S}^+$) to obtain the $\phi_a$'s and $\phi_d$'s of the pedestrians.
        **1.b.** Run **HardNeg**($\mathrm{D}_{out}, \mathcal{S}^-$) to obtain the $\phi_a$'s and $\phi_d$'s of the background examples.
    **2.** Using the $\psi$'s, solve (5) to obtain $\beta$.
    **3.** Update $\mathrm{D}_{out}$ according to the new $\beta$.
**end while**
**return** $\mathrm{D}_{out}$
```

Algorithm 2 VDPM Training

```
*Mandatory*: $\mathcal{V}$, virtual-world data with pixel-wise ground truth for pedestrians as well as pedestrian-free images.
*Optional*: $\mathcal{R}^+$, real-world data with root BB annotations for pedestrians, and $\mathcal{R}^-$ with pedestrian-free images.
**1. Automatic annotation steps.**
**1.a.** Obtain $\mathcal{V}^-$, the pedestrian-free virtual-world images.
**1.b.** Obtain $\mathcal{V}_0^+$ as the complement of $\mathcal{V}^-$ in $\mathcal{V}$.
**1.c.** Obtain $\mathcal{V}^+$ from $\mathcal{V}_0^+$ by performing the automatic annotation of aspects (Sect. III-B) and parts (Sect. III-C).
**2. Build an initial part-based pedestrian detector**
**2.a.** *Appearance classifiers*: using $\{\mathcal{V}^+, \mathcal{V}^-\}$ train each root and parts' initial appearance classifiers.
**2.b.** *Anchor points*: use $\mathcal{V}^+$ to fit a Gaussian mixture model (currently a GMM of five components, *i.e.*, one per part) to the cloud of points generated by considering the centers of the part BBs, independently for each aspect. The mean of each Gaussian is taken as the anchor point of a part.
**2.c.** Build an initial part-based pedestrian detector, $\mathrm{D}_0$, using the appearance classifiers and their anchor locations.
**3. Train the VDPM-MP**
$\mathrm{D}_{out} \leftarrow$ **Optimize**($\mathcal{V}^+, \mathcal{V}^-, \mathrm{D}_0$)
**4. [optional] Virtual to real world domain adaptation**
$\mathrm{D}_{out} \leftarrow$ **Optimize**($\mathcal{R}^+, \mathcal{R}^-, \mathrm{D}_{out}$)
**return** $\mathrm{D}_{out}$
```

TABLE I
STATISTICS OF THE DATA SETS USED IN THIS PAPER.

Testing sets ⇝	Daimler	Daimler*	TUD	Caltech*	CVC02
Images	21,790	973	508	4,024	4,363
≥ 50 pix ped.	6,090	1,483	1,207	1,014	5,016

Training sets ⇝	INRIA	Virtual
Pedestrian-free images	1,218	2,000
Pedestrian examples	1,208	2,500

TABLE II
EVALUATION OF COMPONENTS CLUSTERING METHODS FOR LAT-SVM V5. AVERAGE MISS RATE % IS SHOWN FOR FPPI IN $[10^{-2}, 10^{0}]$.

Clustering Method	Daimler*	TUD	Caltech*	CVC02
Symmetry, c=2	32.1	72.7	68.1	56.2
HOG K-means, c=4	31.0	73.8	68.6	57.2
Our, c=4	29.3	72.7	**64.9**	**49.6**
Our, c=8	**28.4**	**70.0**	**64.5**	50.9

the virtual world for automatically obtaining it. Accordingly, we term this pedestrian model as VDPM-MP. With the VDPM-MP $S(z)$, we search pedestrians in images following [10].

F. Training Framework

Algorithms 1 and 2 summarize the training of our VDPMs. We have coded it within the Lat-SVM V5 framework so that comparisons with such a state-of-the-art method are fair.

Algorithm 1 is at the core of Lat-SVM. **HardNeg()** is the data mining procedure used in [10] for collecting hard negatives. **Detect()** has the purpose of self-annotating components (aspects) and parts in training pedestrians, *i.e.*, estimating the ψ's of (5) during Lat-SVM learning. Hence, we can adopt **HardNeg()**, while the use of **Detect()** is different depending on whether we already have BB annotations for aspects and parts (*e.g.*, as for virtual-world data), or we only have root BBs (*e.g.*, as usually for real-world data).

Accordingly, for the step 3 in Alg. 2, the **Detect()** function only needs to return the aspect and part annotations computed in the step 1 of the same algorithm. However, we have found useful to lead the current detector to search for the best detection (highest score) overlapping up to a certain amount with the provided annotations. In particular, we set to 60% such overlapping for roots and parts individually. This flexibility can be understood as a sort of *online jittering* of the training pedestrians. Augmenting the training set with jittered pedestrians is employed in [17] to be more shift invariant because, for the sake of speed, during pedestrian detection the image is explored according to a stride longer than one pixel. We do this process online, thus our pedestrian training set is not augmented. We have seen that this operation leads to gains between two or three percentage points of accuracy.

For real-world data with only root BBs, **Detect()** is exactly the corresponding step of Lat-SVM V5 training. This means that the current detector is used for collecting aspects and part annotations, but without using the prior annotation information available when training with virtual-world data (the 60% overlapping rule). Thus, step 4 of Alg. 2 consists in training with Lat-SVM V5, but initializing the process with a VDPM detector (Star or MP) based only on virtual-world data. Since VDPM detectors are accurate, they provide a good initialization for the optimization process. The rational behind this optional step is to prepare our framework for domain adaptation based on incorporating real-world data [24], [25].

Finally, the initial part-based detector of step 2 in Alg. 2 follows our proposal in [13]. Thus, we obtain an aspect-based mixture of DPMs with a star structure, with the root and parts trained independently from each other.

V. Experimental Results

A. Datasets and Evaluation Protocol

Since our interest is pedestrian detection for cars, we validate our proposals in different datasets acquired on-board, namely Daimler [1], TUD-Brussels [29], Caltech-Testing [3], and CVC02 [30]. Thus, different camera types and cities are covered. Table I provides relevant statistics of these datasets. Daimler* refers to the *mandatory* set of Daimler we used in [23]. Caltech* refers to the *reasonable* set of Caltech [3].

As evaluation protocol we run the widely used *Caltech* per-image evaluation [3], *i.e.*, false positives per image (FPPI) *vs.* miss rate. Detected pedestrians must be of height ≥ 50 pixels.

Most pedestrian detectors evaluated in [3] are trained with INRIA training set [6]. Thus, for comparing our proposals with respect to them, we use such INRIA data for adapting virtual world to real one. Regarding domain adaptation, here we only focus on combining all the available virtual and real data, *i.e.*, we leave for future work to incorporate either our active learning strategies [25] or the non-supervised ones [24].

Our virtual-world training set contains 2,500 pedestrians and 2,000 pedestrian-free images[1]. Our VDPMs use the root and five parts: shoulder-head, left and right trunk-arms, left and right legs. Lat-SVM V5 uses a 8-part configuration. The root window (*i.e.*, the CW) size is of 48×96 pixels. For detecting pedestrians of height up to 50 pixels, we upscale the images with bilinear interpolation. Part windows are of 24×48 pixels.

Automatic aspect clustering is done once for a desired number of clusters. For the numbers tested in the presented experiments, our clustering procedure (Sect. III-B) roughly takes five minutes for the 2,500 virtual-world pedestrians using MatLab code running on an Intel Xeon CPU E5420 @2.5GHz. The part labeling of the same pedestrians (Sect. III-C) is also done once. By using MatLab code running on the mentioned processor, the part labeling takes around five minutes as well.

The testing of the pedestrian detectors presented in these experiments is always done by running the corresponding part of the Lat-SVM V5 framework. Since the BB prediction post-processing incorporated within this framework requires further training, it is skipped for all tests. In other words, the location of a detected pedestrian directly corresponds to the location of the root. Overall, training and testing of all detectors is done under the same conditions for fair comparison.

B. Results and Discussion

First we assess the accuracy of the component clustering methods for Lat-SVM V5. We train with our virtual-world data. The results are shown in Table II. Our virtual-world pedestrian examples have a fixed aspect ratio, thus the default Lat-SVM V5 clustering method is equivalent to consider two symmetric components. For completeness, we also include K-means clustering of HOG features [31]. Note how our clustering performs better than the rest. Setting $c = 8$ components tends to perform slightly better than $c = 4$. However, since the difference is small, in the following we assume $c = 4$ (3rd level of the hierarchy in Fig. 2a-2b) to obtain a faster detector.

TABLE III
VDPM-Star *vs.* VDPM-MP comparison. Average miss rate % is shown for FPPI in $[10^{-2}, 10^{0}]$.

VDPM (training sets)	Daimler*	TUD	Caltech*	CVC02
Star (V.)	25.1	70.7	**61.5**	48.8
MP (V.)	**24.3**	**65.9**	63.3	**47.5**
Star (V.+INRIA)	21.6	65.7	55.8	42.5
MP (V.+INRIA)	**18.2**	**61.3**	**53.0**	**36.3**

TABLE IV
DPMs average miss rate % is shown for FPPI in $[10^{-2}, 10^{0}]$.

DPM (training sets)	Daimler*	TUD	Caltech*	CVC02
Lat-SVM V5 (V.)	29.3	72.7	64.9	49.6
VDPM-MP (V.)	**24.3**	**65.9**	**63.3**	**47.5**
Lat-SVM V5 (INRIA)	24.7	**60.0**	59.5	42.6
Lat-SVM V5 (V.+INRIA)	23.4	69.6	58.9	42.9
VDPM-MP (V.+INRIA)	**18.2**	61.3	**53.0**	**36.3**

TABLE V
As Table IV substituting INRIA by CVC02 and Caltech†. The '—' avoids testing with the training pedestrians.

DPM (training sets)	Daimler*	TUD	Caltech*	CVC02
Lat-SVM V5 (Caltech†)	57.5	75.4	—	52.5
VDPM-MP (V.+Caltech†)	**18.5**	**60.2**	—	**36.6**
Lat-SVM V5 (CVC02)	60.2	81.1	60.0	—
VDPM-MP (V.+CVC02)	**20.9**	**56.6**	**50.6**	—

Next we compare Star and MP VDPMs using our aspect clustering, with and without domain adaptation. Table III shows the results. Note how effective is combining the virtual- and real-world data: accuracy improves from 4 to 11 percentage points depending on the dataset, MP clearly outperforming Star. Without the combination, VDPMs perform similarly.

Table IV compares Lat-SVM V5 with VDPM-MP. VDPM-MP uses our aspect clustering. Lat-SVM V5 uses the same clustering input when the training data is the virtual-world one, while it applies its own clustering algorithm when the training uses only INRIA. Note how, using the same aspect clustering and the virtual-world data, VDPM-MP reports better accuracy than Lat-SVM V5. This is because VDPM-MP is more flexible than the star model of Lat-SVM V5 and relies on a better initialization of the parts. The same happens combining virtual- and real-world data. Overall, if we compare Lat-SVM V5 trained with INRIA to our VDPM-MP trained with INRIA plus our virtual-world data, we see a large decrease in average miss rate, (~ 6 points for Daimler, Caltech and CVC02).

Fig.4 draws results for the full Daimler set and different Caltech subsets. We add CrossTalk [5] since it recently reported state-of-the-art results on Caltech. CrossTalk uses a holistic pedestrian model learnt by mining many feature channels using AdaBoost style. Note how in the reasonable setting of Caltech the average accuracy of CrossTalk is comparable to VDPM-MP at the moment, while looking only at close pedestrians (*Large* label corresponds to pedestrians over 100 pixels height, *i.e.*, closer than 18 m [3]) VDPM-MP outperforms CrossTalk in 10.5 points, which is very important in driving scenarios. This is in agreement with the fact that DPMs

[1] Publicly available within `www.cvc.uab.es/adas/`

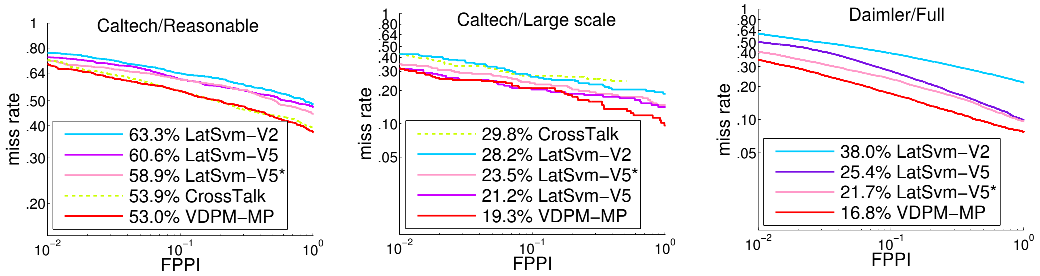

Fig. 4. LatSvm-V2 and LatSvm-V5 are trained with INRIA training dataset, while LatSvm-V5* and VDPM-MP are trained Virtual+INRIA training data.

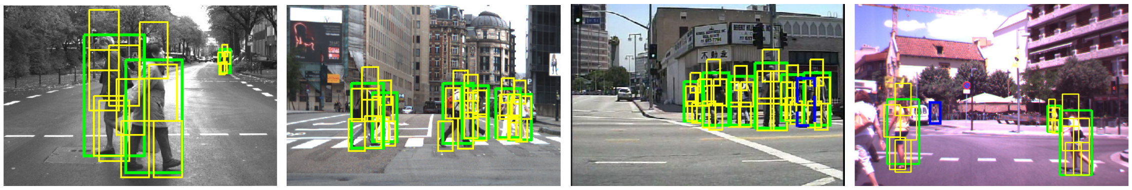

Fig. 5. Detections at FPPI = 0.1 (Daimler/TUD/Caltech/CVC02) for our VDPM-MP trained with Virtual+INRIA data. Blue BBs indicate miss detections. Green ones are root right detections, with corresponding detected parts as yellow boxes.

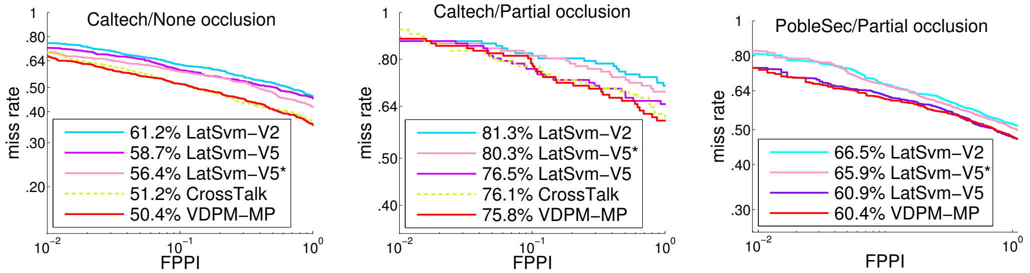

Fig. 6. Behavior of different detectors with respect to occluded pedestrians.

are expected to work better at higher resolutions than holistic models. Finally, Fig. 5 shows qualitative results of VDPM-MP.

For the sake of completeness we have devised a new set of experiments where we have changed the real-world dataset. We have appended the *reasonable* pedestrians of both the training and testing sets of Caltech to obtain a new training set, namely Caltech†, which contains 2,721 pedestrians (roughly twice as much as the INRIA training set). Note that Caltech* $\subset$ Caltech†. We have also used the CVC02 dataset as training set (it contains 5,016 reasonable pedestrians, see Table I). The obtained results, shown in Table V, confirm that our approach clearly outperforms Lat-SVM V5.

For assessing classifiers' accuracy for occluded pedestrians we incorporated the experiments in Fig. 6. We tested on the *partial occlusion* set of Caltech and in our own one [32]. The former containing 102 partially occluded pedestrians over 50 pix height, the latter containing 577. Note how our VDPM-MP clearly outperforms Latent SVM V5 in the non-occluded pedestrians, while for the occluded ones these methods perform analogously. In fact, the accuracy under partial occlusion tends to decrease compared to the non-occlusion case, showing that DPMs may require mechanisms of occlusion detection and re-scoring as we proposed in [32] for holistic models or, alternatively, explicitly incorporating additional components trained with partially occluded pedestrians as special aspects.

Finally, we assessed the processing time of the training and testing frameworks. The training is conducted in an Intel Xeon(R) CPU E51620 of 8 cores at 3.60GHz. The code has parts in C++ and in MatLab, training in parallel the part filters. DPM and VDPM methods consume a similar time to learn the pedestrian models, *i.e.*, between 11 and 12 hours in average for the presented experiments. For testing we have incorporated the proposal of [33] to speed up our linear part filters. Then, using the same CPU as for training, our current C++ implementation runs in the range of 6 to 10 fps.

VI. Conclusions

We have shown how virtual-world data can be used for learning pedestrian DPMs. Using our VDPM-MP proposal and combining virtual- and real-world data, we clearly outperform the state-of-the-art DPM, *i.e.*, Lat-SVM V5. Our automatic aspect clustering and part labeling have two main outcomes. On the one hand, we obtain a more precise initialization for the training optimization procedure. On the other hand, we can train a DPM with part-sharing and aspect clustering. As to the best of our knowledge this is the first work showing how to effectively train such a model by using virtual-world data. As future work we plan to reduce the number of real-world examples required for domain adaptation by testing our approaches in [24], [25]. We want also to improve detection accuracy for partially occluded pedestrians.

References

[1] M. Enzweiler and D. M. Gavrila, “Monocular pedestrian detection: Survey and experiments,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2179–2195, 2009.

[2] D. Gerónimo, A. M. López, A. D. Sappa, and T. Graf, “Survey of pedestrian detection for advanced driver assistance systems,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1239–1258, 2010.

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

[4] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, “Pedestrian detection at 100 frames per second,” in *CVPR*, Providence, RI, USA, 2012.

[5] P. Dollár, R. Appel, and W. Kienzle, “Crosstalk cascades for frame-rate pedestrian detection,” in *ECCV*, Firenze, Italy, 2012.

[6] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR*, San Diego, CA, USA, 2005.

[7] X. Zhu, C. Vondrick, D. Ramanan, and C. C. Fowlkes, “Do we need more training data or better models for object detection?” in *BMVC*, Surrey, UK, 2012.

[8] A. Shashua, Y. Gdalyahu, and G. Hayun, “Pedestrian detection for driving assistance systems: Single–frame classification and system level performance,” in *IV*, Parma, Italy, 2004.

[9] I. Parra, D. Fernández, M. Sotelo, L. Bergasa, P. Revenga, J. Nuevo, M. Ocaña, and M. García, “Combination of feature extraction method for SVM pedestrian detection,” *IEEE Trans. on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 292–307, 2007.

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

[11] M. Enzweiler and D. M. Gavrila, “A multi-level mixture-of-experts framework for pedestrian classification,” *IEEE Trans. on Image Processing*, vol. 20, no. 10, pp. 2967–2979, 2011.

[12] 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 *IV*, Baden-Baden, Germany, 2011.

[13] J. Xu, D. Vázquez, A. López, J. Marín, and D. Ponsa, “Learning a multiview part-based model in virtual world for pedestrian detection,” in *IV*, Gold Coast, Australia, 2013.

[14] Y. Yang and D. Ramanan, “Articulated pose estimation with flexible mixtures-of-parts,” in *CVPR*, Colorado Springs, CO, USA, 2011.

[15] P. Patrick and M. Everingham, “Shared parts for deformable part-based models,” in *CVPR*, Colorado Springs, CO, USA, 2011.

[16] I. Endres, V. Srikumar, M.-W. Chang, and D. Hoiem, “Learning shared body plans,” in *CVPR*, Providence, RI, USA, 2012.

[17] M. Enzweiler and D. Gavrila, “A mixed generative-discriminative framework for pedestrian classification,” in *CVPR*, Anchorage, AK, USA, 2008.

[18] L. Bourdev and J. Malik, “Poselets: Body part detectors trained using 3D human pose annotations,” in *ICCV*, Kyoto, Japan, 2009.

[19] M. Sun and S. Savarese, “Articulated part-based model for joint object detection and pose estimation,” in *ICCV*, Barcelona, Spain, 2011.

[20] S. Branson, P. Perona, and S. Belongie, “Strong supervision from weak annotation: Interactive training of deformable part models,” in *ICCV*, Barcelona, Spain, 2011.

[21] H. Azizpour and I. Laptev, “Object detection using strongly-supervised deformable part models,” in *ECCV*, Firenze, Italy, 2012.

[22] L. Pishchulin, A. Jain, M. Andriluka, T. Thormahlen, and B. Schiele, “Articulated people detection and pose estimation: reshaping the future,” in *CVPR*, Providence, RI, USA, 2012.

[23] J. Marín, D. Vázquez, D. Gerónimo, and A. M. López, “Learning appearance in virtual scenarios for pedestrian detection,” in *CVPR*, San Francisco, CA, USA, 2010.

[24] D. Vázquez, A. López, and D. Ponsa, “Unsupervised domain adaptation of virtual and real worlds for pedestrian detection,” in *ICPR*, Tsukuba, Japan, 2012.

[25] D. Vázquez, J. Marín, A. López, D. Ponsa, and D. Gerónimo, “Virtual and real world adaptation for pedestrian detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, August 2013.

[26] R. Girshick, P. Felzenszwalb, and D. McAllester, “Discriminatively trained deformable part models, release 5,” http://people.cs.uchicago.edu/ rbg/latent-release5/.

[27] D. Gavrila, “A bayesian, exemplar-based approach to hierarchical shape matching,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, no. 8, pp. 1408–1421, 2007.

[28] H. S. Park and C. H. Jun, “A simple and fast algorithm for K-medoids clustering,” *Expert systems with applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[29] C. Wojek, S. Walk, and B. Schiele, “Multi-cue onboard pedestrian detection,” in *CVPR*, Miami Beach, FL, USA, 2009.

[30] D. Gerónimo, A. Sappa, D. Ponsa, and A. López, “2D-3D based on-board pedestrian detection system,” *Computer Vision and Image Understanding*, vol. 114, no. 5, pp. 1239–1258, 2010.

[31] S. Divvala, A. Efros, and M. Hebert, “How important are ”deformable parts” in the deformable parts model?” in *ECCV– Workshop on Parts and Attributes*, Florence, Italy, 2012.

[32] J. Marín, D. Vázquez, A. López, J. Amores, and L. Kuncheva, “Occlusion handling via random subspace classifiers for human detection,” *IEEE Trans. on Systems, Man, and Cybernetics– Part B*, May 2013.

[33] C. Dubout and F. Fleuret, “Exact acceleration of linear object detectors,” in *ECCV*, Firenze, Italy, 2012.

Jiaolong Xu received the B.Sc. degree in Information Engineering in 2008, and the M.Sc. degree in Information and Communication Engineering in 2010, both at the National University of Defence Technology (NUDT), China. Currently, he is a PhD student of the Advanced Driver Assistance Systems (ADAS) group at the Computer Vision Center (CVC) in the Universitat Autònoma de Barcelona (UAB). His research interests include pedestrian detection, virtual worlds, and machine learning.

David Vázquez received the B.Sc. degree in Computer Science from the UAB. He received his M.Sc. in Computer Vision and Artificial Intelligence at CVC/UAB in 2009. As member of the CVC's ADAS group, he obtained the Ph.D. degree in 2013. His research interests include pedestrian detection, virtual worlds, domain adaptation and active learning.

Antonio M. López received the B.Sc. degree in Computer Science from the Universitat Politècnica de Catalunya (UPC) in 1992 and the Ph.D. degree in Computer Vision from the UAB in 2000. Since 1992, he gives lectures in the UAB, where he is now Associate Professor. In 1996, he participated in the foundation of the CVC, where he has held different institutional responsibilities. In 2003 he started the CVC's ADAS group, presently being its head.

Javier Marín received the B.Sc. degree in Mathematics from the Universitat de les Illes Balears (UIB) in 2007. He received his M.Sc. in Computer Vision and Artificial Intelligence at the UAB in 2009. As member of the CVC's ADAS group, he obtained the Ph.D. degree in 2013. His research interests include pedestrian detection, virtual worlds, and random subspace methods.

Daniel Ponsa received the B.Sc. degree in Computer Science and the Ph.D. degree in Computer Vision from the UAB in 1996 and 2007, resp. He did his Ph.D. as member of the CVC's ADAS group. He is currently an Assistant Professor at the UAB as well as member of the CVC's ADAS group. His research interests include tracking, motion analysis, and machine learning, for driver assistance and UAVs.