Domain Adaptation of Deformable Part-Based Models

Jiaolong Xu, *Student Member, IEEE*, Sebastian Ramos, *Student Member, IEEE*,
David Vázquez, *Member, IEEE*, Antonio M. López, *Member, IEEE*

Abstract—The accuracy of object classifiers can significantly drop when the training data (source domain) and the application scenario (target domain) have inherent differences. Therefore, adapting the classifiers to the scenario in which they must operate is of paramount importance. We present novel *domain adaptation* (DA) methods for object detection. As proof of concept, we focus on adapting the state-of-the-art deformable part-based model (DPM) for pedestrian detection. We introduce an *adaptive structural SVM* (A-SSVM) that adapts a pre-learned classifier between different domains. By taking into account the inherent structure in feature space (*e.g.*, the parts in a DPM), we propose a *structure-aware A-SSVM* (SA-SSVM). Neither A-SSVM nor SA-SSVM needs to revisit the source-domain training data to perform the adaptation. Rather, a low number of target-domain training examples (*e.g.*, pedestrians) are used. To address the scenario where there are no target-domain annotated samples, we propose a *self-adaptive DPM* based on a self-paced learning (SPL) strategy and a Gaussian Process Regression (GPR). Two types of adaptation tasks are assessed: from both synthetic pedestrians and general persons (PASCAL VOC) to pedestrians imaged from an on-board camera. Results show that our proposals avoid accuracy drops as high as 15 points when comparing adapted and non-adapted detectors.

Index Terms—domain adaptation, deformable part-based model, pedestrian detection

1 Introduction

Training accurate vision-based object classifiers is essential to the development of reliable object detectors. The main focus for training such classifiers has been the search for the most appropriate image representations and learning machines. In this context, most of the methods for learning classifiers assume that the training data (*source domain*) and the data from the application scenario (*target domain*) are sampled from the same probability distribution. However, in many practical situations this is not the case since even changes in the sensor device can break such an assumption [1], [2]. In other words, a *dataset shift* can be present [3] which significantly impacts the accuracy of classifiers and therefore the overall reliability of the overall object detectors. Accordingly, *domain adaptation* (DA) techniques are crucial to maintain detection accuracy across domains.

One of the most successful object detection methods relies on the learning of a *deformable part-based model* (DPM) using HOG-style features and a latent SVM learning procedure [4]. The DPM can also account for different *components* which, for instance, can be used to more accurately model an object under different views. Therefore, the ability to adapt such a rich model between different domains is essential. Accordingly, the overall aim in this paper is to provide methods for performing domain adaptation of DPMs, empathizing the adaptation of the DPM structure.

We can see a DPM as a particular case of a structural model, where the components and parts define the structure. Accordingly, we formulate the learning of a DPM as a general latent *structural SVM* (SSVM) [5]–[7]. Therefore, we cast the DA of a DPM as a particular case of adapting general structural models. In this context, we propose an *adaptive structural SVM* (A-SSVM) method motivated by the adaptive SVM (A-SVM) [8]. Furthermore, since A-SSVM works irrespective of the model structure (*e.g.*, the parts and components in a DPM), we also propose a *structure-aware A-SSVM* (SA-SSVM) method. Remarkably, neither A-SSVM nor SA-SSVM need to revisit the training data from the source domain, instead a relatively low number of training examples (*i.e.*, object instances) from the target domain are used to adapt the structural model that has been initially learned in the source domain.

Although A-SSVM and SA-SSVM only require a few manually annotated target-domain examples for the adaptation, we also address the more challenging situation of even avoiding such manual annotations. In particular, we have devised an iterative method for automatically discovering and labeling samples in the target domain and re-training an adapted classifier with them using either A-SSVM or SA-SSVM. Our method applies a self-paced learning (SPL) strategy [9] to re-train an initial model with increasingly difficult target-domain examples in an iterative way

- *Jiaolong Xu and Antonio M. López are with the Computer Vision Center (CVC) and the Computer Science Department at the Universitat Autònoma de Barcelona. Sebastian Ramos and David Vázquez are with the CVC.*
E-mails: {jiaolong,sramosp,dvazquez,antonio}@cvc.uab.es

without requiring source-domain data. The proper definition of what is an *easy/difficult* sample (example or counter-example) is essential for the SPL. However, in general it turns out that discovering easy/difficult samples in a new domain is a non-trivial task. In this paper, we apply Gaussian Process Regression (GPR) for performing such a *sample selection*, which can also simplify the SPL optimization procedure proposed in [9]. We call our proposal the *self-adaptive DPM*.

As proof of concept, we apply the proposed techniques to *pedestrian detection*, a very relevant topic in computer vision. Classifying a candidate window as pedestrian or background turns out to be a very difficult task due to a combination of factors [10]–[12]. In short, these factors are the huge intra-class variability of both pedestrians and background, as well as the changing imaging and environmental conditions. Note that pedestrians are moving objects with varying morphology, pose and clothes; they can be in a large variety of indoor and outdoor scenarios; and for some applications (*e.g.*, driving assistance) images are acquired from a moving vehicle. Thus, pedestrians can be seen from different viewpoints at a range of distances and under uncontrolled illumination. Different approaches that address the pedestrian detection problem have been proposed in the last decade. Among them, the DPM is a state-of-the-art method for this application [12] and for object detection [4] in general.

We evaluate two different situations in the context of adapting a pedestrian DPM. We adapt our pedestrian classifiers learned with synthetic data (*i.e.*, root annotations are generated automatically) to operate on real-world images. Furthermore, we adapt the generic person classifier from the PASCAL VOC to detect people in INRIA data. In the former case the drop in accuracy without adaptation is presumably due to the fact that the synthetic and real-world data differ in appearance. In the latter case the drop in accuracy may be due to the large differences in typical views, poses and resolutions between training and testing data, which also represents a very challenging case. The conducted experiments show that our proposals avoid accuracy drops of as high as 15 percentage points when comparing adapted and non-adapted detectors.

The rest of the paper is organized as follows. In section 2 we overview the domain-adaptation related work, including approaches that focus on pedestrian detection. In section 3 we summarize the main ideas of the DPM and structural learning. In section 4 we explain our supervised domain adaptation proposals for DPMs, namely A-SSVM and SA-SSVM. In section 5 we present our self-adaptive DPM for working with unlabeled or weakly labeled target domains. In section 6 we assess the results of our proposals in the field of pedestrian detection. Finally, in section 7 we draw the main conclusions and future research lines.

2 Related work

The DA scenario has been explored for different applications by the machine learning community and recently it is becoming more attractive to the computer vision community. Focusing on DA of visual data, we review SVM-based methods, including supervised, semi-supervised and unsupervised approaches.

Supervised Methods: The most common approach consists of a weighted combination of SVMs learned in the source domain and SVMs learned in the target domain [17], [18], [21], [22]. The principal drawback of these methods is that they require both source and target domain training data for the adaptation, which makes it computationally expensive. It may even result in negative transfer (*i.e.*, the accuracy decreases for the target domain) as reported in [22]. Alternatively, a feature replication approach is proposed in [23], which jointly learns classifiers in both domains with augmented features, *i.e.*, source-domain data is also required. Another approach, the cross-domain SVM (CD-SVM) [24], selects the source domain support vectors that are close to the target domain and also adds new support vectors from the target domain to learn a new classifier. Nonetheless, in the case that the target domain data are scarce, the learned classifier may still be source domain oriented. The adaptive SVM (A-SVM) proposed in [8] learns a perturbation function that restricts the new decision boundary to be close to the original source boundary. Furthermore, several A-SVM variants have also been proposed, *e.g.*, least-squares SVMs based method [25] and projective model transfer SVM (PMT-SVM) [26].

In comparison to these works, our supervised methods, A-SSVM and SA-SSVM, share the advantages of A-SVM and PMT-SVM of not requiring source-domain training data for the adaptation process. Furthermore, our proposals take into account structure knowledge in feature space.

Semi-supervised / Unsupervised Methods: The domain transform SVM (DT-SVM) of [27] minimizes the distribution mismatch of labeled and unlabeled samples between different domains. The transductive SVM (T-SVM) is employed in [21] for improving the accuracy of classifiers trained with weakly labeled web images. The transform-based methods [28], [29] use labeled source and unlabeled target data to construct a manifold and learn a classifier from a projected space. In [30], transform component analysis [31] is used to adapt a car detector to the target domain but the overall accuracy may be limited by the holistic detector. Similar to us, the online DA of [32] also applies GPR for re-scoring, but it treats each testing image as an individual target domain, which implies that a sufficient number of examples is required per-image. This is the case in the original paper when applied to face images, but the adaptation may be poor if the target image contains very few examples.

TABLE 1
Comparison of DA methods for pedestrian detection

	Adaptation method	Model	Prior model	Require source data	Labeled target data	Unlabeled target data
Cao *et al.* [13]	Boosting	Holistic	no	yes	yes	no
Pang *et al.* [14]	Boosting	Holistic	no	yes	yes	no
Vázquez *et al.* [1], [15]	SVM	Holistic	no	yes	yes	no
Vázquez *et al.* [16]	T-SVM	Holistic	yes	yes	Optional	yes
Wang *et al.* [17], [18]	SVM	Holistic	yes	yes	Optional	yes
Xu *et al.* [19]	LDA, Boosting	Holistic	no	yes	yes	no
Donahue *et al.* [20]	PMT-SVM	Mixture root	yes	yes	yes	yes (traject.)
This paper	SSVM	Part-based	yes	no	Optional	yes

In contrast to previous approaches, our self-adaptive DPM uses non manually labeled target-domain samples (or weakly labeled samples when image-level labels are available [22]) that are automatically discovered by an iterative process and without requiring source-domain data. Unlike in [22], our method does not rely on motion segments. Compared to [32], our method leverages target domain examples from multiple images and further incorporates GPR for fine-level sample selection.

DA for Pedestrian Detection: Most of the related work on DA for computer vision tasks is focused on object recognition [33], while its application to object detection is quite limited. Table 1 briefly compares recently proposed adaptive detection methods (especially for pedestrian detection). Among these methods, [13], [14], [19] are boosting-based approaches while the others are SVM-based. The authors of [17], [18] use a weighted combination to adapt a generic pedestrian detector to a specific scene. Recently, the authors of [20] proposed a semi-supervised DA approach which combines an instance-constrained manifold regularization with the PMT-SVM, where a few labeled target domain examples are required. In [22], DA is applied to adapt an object detector from video to images. However, only a weighted combination of source and target classifier is explored for DPM.

The authors of [1], [15] investigated the adaptation of a holistic pedestrian model trained with virtual-world samples to operate on real-world images. Using a framework called V-AYLA, virtual-world samples and real-world ones are fused for training and adapting a model within the so-called *cool world*. In these works the focus is on relevant pedestrian descriptors (HOG and LBP [1], Haar and EOH [15]) as well as on the type of complementarity between virtual- and real-world data. Here we go beyond in several aspects: (1) we focus on a state-of-the-art pedestrian detection method, namely the DPM, providing not only adaptation of pedestrian descriptors but also of the deformable model and the multiple components (A-SSVM, SA-SSVM); (2) such an adaptation does not require the use of the cool word anymore, *i.e.*, the models are adapted by considering real-world backgrounds and a relatively few pedestrians; (3) the proposed self-adaptive DPM aims to avoid human intervention during the adaptation process.

The authors of [16] also investigated the use of an iterative unsupervised DA technique for the holistic pedestrian detector based on HOG/Lin-SVM. This technique is based on Transductive SVM and, in fact, has turned out to be rather time consuming since both labeled and unlabeled samples are used to learn during each iteration. In comparison, instead of using a fixed threshold, our self-adaptive DPM uses a combination of SPL and GPR to handle unlabeled target domain samples. Moreover, since we do not need source-domain data for the adaptation, the learning algorithm is faster than the one in [16].

3 DPM and Structural Learning

The DPM [4] is defined by one root filter and a pre-set number of part filters. Part filters operate at twice the resolution of the root filter. The root acts as reference and all other parts are connected to this reference (star model). To better capture intra-class variations, star models can be further combined into a mixture of components (*e.g.*, representing different views).

To detect objects in an image, a sliding window search is applied in the image pyramid. Suppose that the DPM has M components and that each component has K parts. Then, an object hypothesis is defined by $\mathbf{h} = [c, \mathbf{p}'_0, \ldots, \mathbf{p}'_K]', c \in [1, M]$, where $\mathbf{p}_j = [u_j, v_j, s_j]'$ specifies the position (u_j, v_j) and scale level s_j of part $j \in [0, K]$, $j = 0$ identifies the root. The DPM takes into account appearance features as well as part deformations. Given a candidate image window $\mathbf{x}$ and an associated hypothesis $\mathbf{h}$, for a single component c, the decision function can be written in terms of a dot product between the parameter vector $\mathbf{w}_c$ and the feature vector $\Phi_c(\mathbf{x}, \mathbf{h})$ as:

$$\mathbf{w}'_c \Phi_c(\mathbf{x}, \mathbf{h}) = \sum_{j=0}^{K} \mathbf{F}'_{cj} \phi_a(\mathbf{x}, \mathbf{h}) - \sum_{j=1}^{K} \mathbf{d}'_{cj} \phi_d(\mathbf{p}_j, \mathbf{p}_0) + b_c, \tag{1}$$

where $\phi_a(\mathbf{x}, \mathbf{h})$ represents the appearance feature vector (*e.g.*, HOG descriptors), and $\phi_d(\mathbf{p}_j, \mathbf{p}_0) =$

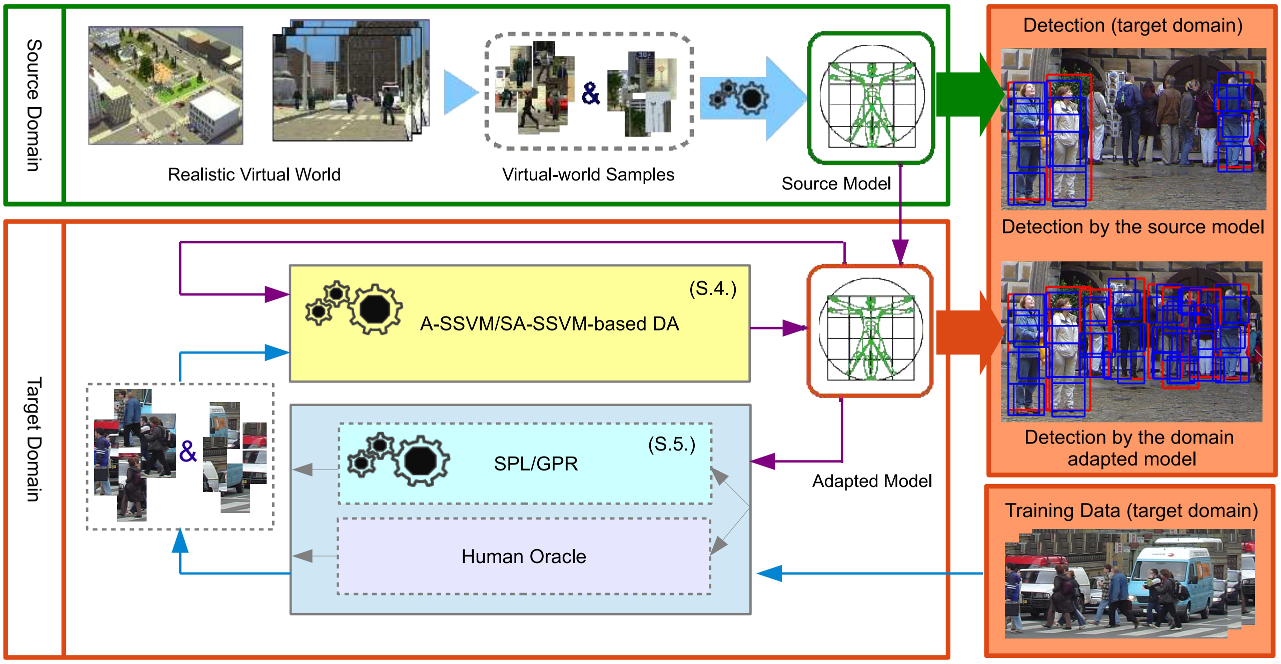

Fig. 1. Proposed framework for *domain adaptation* (DA) of the SVM-based *deformable part-based model* (DPM). The figure shows the adaptation of a DPM-based pedestrian detector from a virtual-world *source domain* to a real-world *target domain*. As DA module we propose an *adaptive structural* SVM (A-SSVM) and a *structure-aware* A-SSVM (SA-SSVM), see Sect. 4. A-SSVM and SA-SSVM require target-domain labeled samples (*e.g.*, a few pedestrians and background) that can be provided by a human oracle. Alternatively, we propose a strategy inspired by *self-paced learning* (SPL) and supported by a *Gaussian Process Regression* (GPR) for the automatic labeling of samples in unlabeled or weakly labeled target domains. The combination of SPL/GPR with either A-SSVM or SA-SSVM gives rise to our *self-adaptive DPM* (see Sect. 5).

$[dx_j, dx_j^2, dy_j, dy_j^2]'$ is the deformation function of part j with respect to part 0 (root). $\mathbf{F}_{cj}$ are the appearance parameters, $\mathbf{d}_{cj}$ is a four-dimensional vector specifying the coefficients of deformation cost, and b_c is the bias term. For the multiple component model, the *one-vs-rest* approach can be employed and the final decision function is written as:

$$f(\mathbf{x}) = \max_{\mathbf{h}} \mathbf{w}'\Phi(\mathbf{x}, \mathbf{h}), \tag{2}$$

where $\mathbf{w} = [\mathbf{w}'_1, \ldots, \mathbf{w}'_M]'$, $\Phi = [\mathbf{0}'_{n_1}, \ldots, \Phi'_c, \ldots, \mathbf{0}'_{n_M}]'$.

Thus, DPM training aims to learn an optimum $\mathbf{w}$ which encodes the appearance parameters and deformation coefficients. Suppose we are given a set of training samples $(\mathbf{x}_1, y_1, \mathbf{h}_1), \ldots, (\mathbf{x}_N, y_N, \mathbf{h}_N) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{H}$, where $\mathcal{X}$ is the input space, $\mathcal{Y} = \{+1, -1\}$ is the label space, and $\mathcal{H}$ is the hypothesis or output space. We write the features as joint feature vectors $\Phi(\mathbf{x}, \mathbf{h})$. In the DPM case [4], $\mathbf{h}$ is not given and is therefore treated as a latent variable during training.

The discriminative function of (2) can be learned by the max-margin method, *e.g.*, using latent SVM as in [4]. The latest version of the DPM (version 5.0) generalizes the SSVM and latent SSVM in a weak-label SSVM, which subsumes latent SVM as a special case [34]. Computing the optimum $\mathbf{w}$ for the score function (2) is equivalent to solving the following latent SSVM optimization problem:

$$\underbrace{\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max_{\widehat{y}, \widehat{\mathbf{h}}}[\mathbf{w}'\Phi(\mathbf{x}_i, \widehat{\mathbf{h}}) + L(y_i, \widehat{y}, \widehat{\mathbf{h}})]}_{convex}$$
$$\underbrace{-C\sum_{i=1}^{N} \max_{\mathbf{h}} \mathbf{w}'\Phi(\mathbf{x}_i, \mathbf{h})}_{concave}, \tag{3}$$

where parameter C is the relative penalty scalar parameter, $L(y_i, \widehat{y}, \widehat{\mathbf{h}})$ represents the loss function, $\widehat{y}$ the predicted label, and y_i the ground truth label. In particular, we use 0-1 loss for object detection, *i.e.*, $L(y_i, \widehat{y}, \widehat{\mathbf{h}}) = 0$ if $\widehat{y} = y_i$ and 1 otherwise. The latent SSVM optimization objective function (3) can be viewed as minimizing the sum of a convex and concave function and it can be solved by the coordinate descend method as in [4] or by the general Convex-Concave Procedure (CCCP) in [34], which is a simple iterative procedure that guarantees the convergence to a local minimum or a stationary point of the objective function. For a comprehensive explanation, we refer the reader to [5]–[7], [34].

4 Domain Adaptive DPM

Based on the DPM framework, we propose our Domain Adaptive DPM (DA-DPM), which is illustrated in Fig. 1. To adapt a DPM detector from a source domain to a different target domain, we first assume the supervised DA task. This means that both source and target domain labels are given. Let $\mathcal{D}_l^S$ denote the labeled source domain and $\mathcal{D}_l^T$ the labeled target domain. We assume that a DPM has been trained in the source domain, we denote by $\mathbf{w}^S$ the corresponding parameter vector. Thus, our goal is to adapt $\mathbf{w}^S$ to the target domain, using a relatively low number of target-domain labeled examples, so that we obtain a more accurate model $\mathbf{w}$ for the new domain.

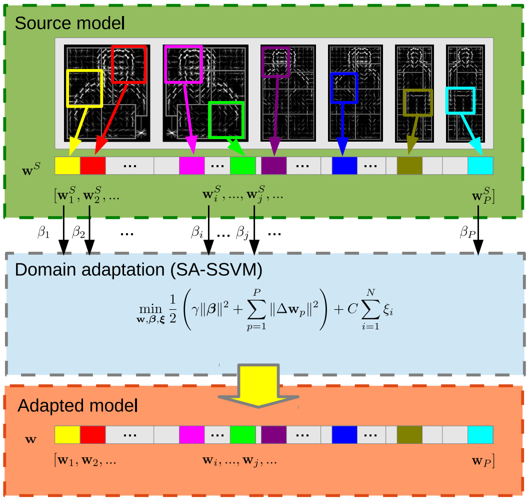
Source model
$\mathbf{w}^S$
···
$[\mathbf{w}_1^S, \mathbf{w}_2^S, ...$
$\mathbf{w}_i^S, ..., \mathbf{w}_j^S, ...$
$\mathbf{w}_P^S]$
β_1
β_2
β_i
β_j
β_P
Domain adaptation (SA-SSVM)
$\min_{\mathbf{w},\boldsymbol{\beta},\boldsymbol{\xi}} \frac{1}{2}\left(\gamma\|\boldsymbol{\beta}\|^2 + \sum_{p=1}^{P}\|\Delta\mathbf{w}_p\|^2\right) + C\sum_{i=1}^{N}\xi_i$
Adapted model
$\mathbf{w}$
$[\mathbf{w}_1, \mathbf{w}_2, ...$
$\mathbf{w}_i, ..., \mathbf{w}_j, ...$
$\mathbf{w}_P]$

Fig. 2. Domain adaptation for DPM: Structure-aware Adaptive Structural SVM (SA-SSVM).

4.1 Adaptive SSVM (A-SSVM)

Our first proposal is based on the adaptive SVM (A-SVM) [8], an effective DA algorithm that uses a prior model and learns a perturbation function based on a pre-trained source classifier. We extend it for structural learning, namely adaptive SSVM (A-SSVM).

Given the source model $\mathbf{w}^S$, the final classifier f^T is defined by

$$f^T(\mathbf{x}) = \max_{\mathbf{h}}[\mathbf{w}^{S'}\Phi(\mathbf{x},\mathbf{h}) + \underbrace{\Delta\mathbf{w}'\Phi(\mathbf{x},\mathbf{h})]}_{\Delta f(\mathbf{x})} \ , \qquad (4)$$

where $\Delta f(\mathbf{x})$ is called the perturbation function, $\Delta\mathbf{w} = \mathbf{w} - \mathbf{w}^S$, $\mathbf{w}^S$ is the prior model, and $\mathbf{w}$ the final adapted model. The basic idea is to learn a new decision boundary close to the original source decision one. The new decision function (4) can be obtained by solving the following optimization problem:

$$\min\nolimits_{\Delta\mathbf{w}} \mathcal{R}(\Delta\mathbf{w}) + C\mathcal{L}(\Delta\mathbf{w}, \mathcal{D}_l^T), \qquad (5)$$

where $\mathcal{R}$ is a regularizer, $\mathcal{L}$ represents the loss term on target data, and C is a penalty scalar parameter as in (3). Furthermore, (5) can be explicitly written as:

$$\begin{aligned} &\min\nolimits_{\mathbf{w},\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w} - \mathbf{w}^S\|^2 + C\sum\nolimits_{i=1}^{N}\xi_i \\ &\text{s.t.} \quad \forall i, y, \mathbf{h}, \quad \xi_i \geq 0, \quad \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^T \\ &\mathbf{w}'\Phi(\mathbf{x}_i, \mathbf{h}_i) - \mathbf{w}'\Phi(\mathbf{x}_i, \mathbf{h}) \geq L(y_i, y, \mathbf{h}) - \xi_i \ , \end{aligned} \qquad (6)$$

where y_i and $\mathbf{h}_i$ are the ground truth label and object hypothesis, y and $\mathbf{h}$ represent all the alternative output label and object hypothesis, and $\boldsymbol{\xi} = [\xi_1, \ldots, \xi_N]'$.

The regularization term shows that A-SSVM adapts the model learned in the source domain towards the target domain by regularizing the distance between $\mathbf{w}$ and $\mathbf{w}^S$. Equivalent to the optimization of SSVM [35], the primal form minimization problem of (6) has its closely related maximization dual form problem. By introducing the Lagrange multiplier $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_N]'$, we can analyse the DA in the dual form:

$$\begin{aligned} &\max\nolimits_{\boldsymbol{\alpha}} \sum\nolimits_{i,\overline{y},\overline{\mathbf{h}}} \alpha_i(\overline{y}, \overline{\mathbf{h}})[L(y_i, \overline{y}, \overline{\mathbf{h}}) - \mathbf{w}^{S'}\Delta\Phi_{i,\overline{\mathbf{h}}}] \\ &-\frac{1}{2}\sum\nolimits_{i,\overline{y},\overline{\mathbf{h}}}\sum\nolimits_{j,\widehat{y},\widehat{\mathbf{h}}} \alpha_i(\overline{y}, \overline{\mathbf{h}})\alpha_j(\widehat{y}, \widehat{\mathbf{h}})\Delta\Phi_{i,\overline{\mathbf{h}}}\Delta\Phi_{j,\widehat{\mathbf{h}}} \end{aligned} \ , \qquad (7)$$

where $\overline{y}, \overline{\mathbf{h}}, \widehat{y}, \widehat{\mathbf{h}}$ are alternative labels and object hypotheses other than the ground truth, and $\Delta\Phi_{i,\mathbf{h}} = \mathbf{w}^{S'}[\Phi(\mathbf{x}_i, \mathbf{h}_i) - \Phi(\mathbf{x}_i, \mathbf{h})]$. Comparing (7) to the dual form of the standard SSVM [35], the only difference comes from the fact that (7) contains the term $\mathbf{w}^{S'}\Delta\Phi_{i,\overline{\mathbf{h}}}$. Let $L_s = \mathbf{w}^{S'}\Delta\Phi_{i,\overline{\mathbf{h}}}$. Then $L_s < 0$ indicates that the output is incorrectly predicted by the source classifier in the target domain. Therefore, a larger α_i is preferred in order to maximize the dual form (7) and vice versa. Note that only the target-domain samples $\mathbf{x}_i \in \mathcal{D}_l^T$ are used during the training and α_i is equivalent to the weight of the vector $\mathbf{x}_i$. Thus, the A-SSVM tunes the model parameters towards the target-domain data.

4.2 Structure-aware A-SSVM (SA-SSVM)

The A-SSVM regularization constrains the new classification hyperplane should to not deviate far from the source one, and thus it requires that the source and the target domains have the same feature representation and similar feature distributions. This is very strict for a mixture component part-based model. First of all, it does not take into account the inherent structure knowledge of the model. Secondly, it may not be effective when the source and the target domains have significant differences in the feature space, *e.g.*, substantial differences in view or pose distribution. Since we use the joint feature map, *i.e.*, $\Phi(\mathbf{x}, \mathbf{h})$ for structure learning, the learned hyperplane parameters naturally encode the structural knowledge from the space $\mathcal{X} \times \mathcal{H}$. For example, by taking a deeper look at the learned DPM hyperplane, its corresponding parameter vector can be divided into blocks by the mixture components or parts. This motivates us to consider adapting a prior model with structural knowledge, namely our structure-aware A-SSVM (SA-SSVM).

Fig. 2 illustrates the SA-SSVM method with a person DPM. First, we learn the DPM in the source domain. This model, $\mathbf{w}^S$, consists of components: half body and full body, as well as persons seen from different viewpoints. Each component consists of parts: head, torso, etc. To adapt this DPM to a different domain, we decompose the structural model as $\mathbf{w}^S = [\mathbf{w}_1^{S'}, \ldots, \mathbf{w}_P^{S'}]'$, where P is the number of partitions. Note that each component, $\mathbf{w}_p^S$, may contain both appearance and deformation parameters. The decomposed model parameters are adapted to the target domain by different weights, denoted by $\beta_p, p \in [1, P]$ as in Fig. 2. In order to learn these adaptation weights, we further introduce a regularization term $\|\boldsymbol{\beta}\|^2$ in the objective function, and we use a scalar parameter γ to control the relative penalty to the hyperplane parameter regularization term.

We define $\Delta\mathbf{w} = [\Delta\mathbf{w}'_1, ..., \Delta\mathbf{w}'_P]'$, $\boldsymbol{\beta} = [\beta_1, \ldots, \beta_P]'$, where $\Delta\mathbf{w}_p = \mathbf{w}_p - \beta_p\mathbf{w}_p^S$, and $p \in [1, P]$. The regularization term of A-SSVM in (5) can be modified as:

$$\mathcal{R}\left(\mathbf{w}, \boldsymbol{\beta}, \mathbf{w}^S\right) = \frac{1}{2}\left(\gamma\|\boldsymbol{\beta}\|^2 + \textstyle\sum_{p=1}^P \|\Delta\mathbf{w}_p\|^2\right). \tag{8}$$

The SA-SSVM optimization is then formulated as:

$$\begin{array}{l} \min_{\mathbf{w},\boldsymbol{\beta},\boldsymbol{\xi}} \mathcal{R}\left(\mathbf{w}, \boldsymbol{\beta}, \mathbf{w}^S\right) + C\sum_{i=1}^N \xi_i \\ \text{s.t.} \quad \forall i, y, \mathbf{h}, \quad \xi_i \geq 0, \quad \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^T \\ \mathbf{w}'\Phi(\mathbf{x}_i, \mathbf{h}_i) - \mathbf{w}'\Phi(\mathbf{x}_i, \mathbf{h}) \geq L(y_i, y, \mathbf{h}) - \xi_i \ . \end{array} \tag{9}$$

There are two parameters to be optimized in the SA-SSVM objective function (9), *i.e.*, $\boldsymbol{\beta}$ and $\mathbf{w}$.

Directly optimizing (9) is difficult using off-the-shelf tools. By re-arranging the feature and parameter representation, we convert (9) into a quadratic programming (QP) problem which can be solved by a standard SVM solver. We introduce a concatenated vector $\tilde{\mathbf{w}} = [\Delta\mathbf{w}', \sqrt{\gamma}\boldsymbol{\beta}']'$ and

$$\tilde{\Phi}(\mathbf{x}_i, \mathbf{h}) = [\Phi(\mathbf{x}_i, \mathbf{h})', \frac{1}{\sqrt{\gamma}}\Theta^S(\mathbf{x}_i)']', \tag{10}$$

where $\Theta^S(\mathbf{x}_i) = [\mathbf{w}_1^{S'}\Phi_1(\mathbf{x}_i, \mathbf{h}), ..., \mathbf{w}_P^{S'}\Phi_P(\mathbf{x}_i, \mathbf{h})]'$, and $\Phi_p(\mathbf{x}, \mathbf{h})$ stands for the features of part p given the candidate $\mathbf{x}$ and the hypothesis $\mathbf{h}$. Then, the optimization problem in (9) can be rewritten as follows:

$$\begin{array}{l} \min_{\tilde{\mathbf{w}},\boldsymbol{\beta},\boldsymbol{\xi}} \mathcal{R}(\tilde{\mathbf{w}}) + C\sum_{i=1}^N \xi_i \\ \text{s.t.} \quad \forall i, y, \mathbf{h}, \quad \xi_i \geq 0, \quad \forall(\mathbf{x}_i, y_i) \in \mathcal{D}_l^T \\ \tilde{\mathbf{w}}'\tilde{\Phi}(\mathbf{x}_i, \mathbf{h}_i) - \tilde{\mathbf{w}}'\tilde{\Phi}(\mathbf{x}_i, \mathbf{h}) \geq L(y_i, y, \mathbf{h}) - \xi_i \ , \end{array} \tag{11}$$

where $\mathcal{R}(\tilde{\mathbf{w}}) = \sum_{p=1}^P \|\tilde{\mathbf{w}}_p\|^2$ and $\tilde{\mathbf{w}}_p = [\Delta\mathbf{w}_p{}', \sqrt{\gamma}\beta_p]'$.

Note that the regularization term $\mathcal{R}(\tilde{\mathbf{w}})$ is convex and the loss term in (11) is also convex, thus the objective function of SA-SSVM is convex. In the following, we discuss several properties of the proposed SA-SSVM.

Part-level adaptation. In contrast to A-SVM, adaptive regularization is performed on partitions. Analogously to the A-SSVM decision function (4), we can write the SA-SSVM decision function as:

$$f^T(\mathbf{x}) = \max_{\mathbf{h}}[\sum_{p=1}^P \beta_p\mathbf{w}_p^{S'}\Phi_p(\mathbf{x}, \mathbf{h}) + \underbrace{\Delta\mathbf{w}'\Phi(\mathbf{x}, \mathbf{h})}_{\Delta f(\mathbf{x})}]. \tag{12}$$

Compared to (4), (12) decomposes the pre-learned classifier into a set of *part classifiers* and the final score is a weighted combination of the prior part classifiers and the perturbation functions. Thus, it takes into account the structural knowledge of the prior model.

Part-level regularization is also proposed in [36], however the parts are taken from multiple holistic templates for transfer learning and the new model is still a rigid holistic template. In contrast to [36], we consider the structure in the single prior model and perform decomposition to the part-based model. The part appearances as well as the deformation in the prior model are adapted in the new model. Using structural correspondence for DA was also proposed in [37]. Structural correspondence is learned with the extracted pivot features from source and target domains. However, the method is specially designed for cross-language text classification tasks.

Properties of γ. The regularization term $\gamma\|\boldsymbol{\beta}\|^2$ controls the adaptation degree of the model. As can be seen from the primal form of the objective function (8) and (9), when $\gamma \to \infty$ β_p is forced to be zero, due to the infinite penalty. Thus (9) converges to non-adaptive SSVM. As $\gamma \to 0$, the penalty on β_p is small, thus it adapts more to the prior model.

Feature augmentation. Note that the joint feature representation in (10) is a concatenation of $\Phi(\mathbf{x}_i, \mathbf{h})$ and the part responses of the source classifiers, as $\Theta^S(\mathbf{x}_i)$. Thus, for the adapted classifier $\tilde{\mathbf{w}}$, $\tilde{\Phi}(\mathbf{x}_i, \mathbf{h})$ is an augmented feature with responses in $\Theta^S(\mathbf{x}_i)$.

We can also analyze the properties of the dual form. Letting $\boldsymbol{\alpha}$ be the Lagrange multiplier, the dual form of the optimization problem (11) can be written as:

$$\begin{array}{l} \max_{\boldsymbol{\alpha}} \sum_{i,\overline{y},\overline{\mathbf{h}}} \alpha_i(\overline{y}, \overline{\mathbf{h}})L(y_i, \overline{y}, \overline{\mathbf{h}}) \\ -\frac{1}{2}\sum_{i,\overline{y},\overline{\mathbf{h}}}\sum_{j,\widehat{y},\widehat{\mathbf{h}}} \alpha_i(\overline{y}, \overline{\mathbf{h}})\alpha_j(\widehat{y}, \widehat{\mathbf{h}})\Delta\tilde{\Phi}'_{i,\overline{\mathbf{h}}}\Delta\tilde{\Phi}_{j,\widehat{\mathbf{h}}} \ , \end{array} \tag{13}$$

where the expression $\Delta\tilde{\Phi}'_{i,\overline{\mathbf{h}}}\Delta\tilde{\Phi}_{j,\widehat{\mathbf{h}}} = \Delta\Phi_{i,\overline{\mathbf{h}}}{}'\Delta\Phi_{j,\widehat{\mathbf{h}}} + \frac{1}{\gamma}(\mathbf{w}^{S'}\Delta\Phi_{i,\overline{\mathbf{h}}})(\mathbf{w}^{S'}\Delta\Phi_{j,\widehat{\mathbf{h}}})$ is defined by the labeled training data from the target domain. Thus, the kernel $\Delta\tilde{\Phi}'_{i,\overline{\mathbf{h}}}\Delta\tilde{\Phi}_{j,\widehat{\mathbf{h}}}$ takes into account both visual information from the new domain data and the partial responses of the pre-learned model, which can lead to better discriminative power. Again we see that γ controls the degree of adaptation, as $\gamma \to \infty$ indicates no adaptation and $\gamma \to 0$ indicates maximum adaptation.

Algorithm 1 Supervised DA-DPM

```
Input: $\mathbf{w}^S, \epsilon$, target-domain training samples:
$\mathcal{D}_l^T = \{(\mathbf{x}_i, y_i)\}, i \in (1, N)$.
Output: $\mathbf{w}$
0: $\mathbf{w} \leftarrow \mathbf{w}^S$
1: Repeat
2:     Update $\mathbf{h}_i^* = \arg\max_{\mathbf{h}} \mathbf{w}'\Phi(\mathbf{x}_i, \mathbf{h}), \forall i$.
3:     Update $\mathbf{w}$ by fixing the hidden variables to $\mathbf{h}_i^*$ and solving the DA optimization problem with A-SSVM (6) or SA-SSVM (11).
4: Until the objective function ((6) or (11)) cannot be decreased below tolerance $\epsilon$.
```

4.3 Supervised DA-DPM Algorithm

We apply the proposed A-SSVM and SA-SSVM algorithms to learn a domain adapted DPM. The A-SSVM and SA-SSVM are built on the SSVM, which assumes that the ground truth of all outputs $\mathbf{h}$ is given. To apply these techniques to the DPM, we incorporate the latent variables by decomposing the objective functions as the sum of convex and concave parts as in (3), thus we can employ the CCCP to solve the latent A-SSVM and SA-SSVM optimization problems. The procedure is formalized in Alg. 1. This algorithm has two main parts: (1) updating the hidden variables (step 3) by approximating the concave function with a linear upper bound; and (2) fixing the hidden variables and updating the parameters by solving a convex A-SSVM or SA-SSVM learning problem.

5 Self-adaptive DPM

5.1 Self-paced Learning (SPL)

To address a DA scenario without target-domain labeled data, we could directly apply the source detector to discover examples (positive samples) and counter-examples (negative samples) in the target domain, and then use them to run A-SSVM or SA-SSVM. However, these collected samples may contain a large number of false positives, due to the domain shift and the inherent detection error of any classifier. In that case, the DA method can get stuck in a local optimum with high training error due to the fact that the CCCP (and so Alg. 1) considers all samples simultaneously. A strategy analogous to SPL, which starts with the *easiest* samples and gradually considers more complex ones, can be employed to handle this problem.

In SPL, the *easy* samples are defined as those with the highest level of confidence [38], where such a confidence relies on a SVM-based classification score in our case (*e.g.*, the highest absolute value of the score could indicate higher classification confidence). At this point we face a scenario where we must apply a source-domain classifier in a target domain without labels. Therefore, we must distinguish between positive and negative target-domain samples and determine for which samples the decision was easy, all in presence of a domain shift. Accordingly, a simple threshold on the absolute value of the classification score is not an appropriate measure for determining if a sample is easy or not, because: (1) if the easy samples are selected too conservatively (high threshold), the adaptation would be poor since these samples are far away from the hyperplane margin and more likely source-domain oriented; and (2) if the easy samples are aggressively selected (low threshold), many mislabeled ones may be collected for the adaptation. Therefore, rather than defining *easiness* according to a fixed threshold directly applied to our SVM-based classification scores, we propose a more adaptive *sample selection* process based on a GPR.

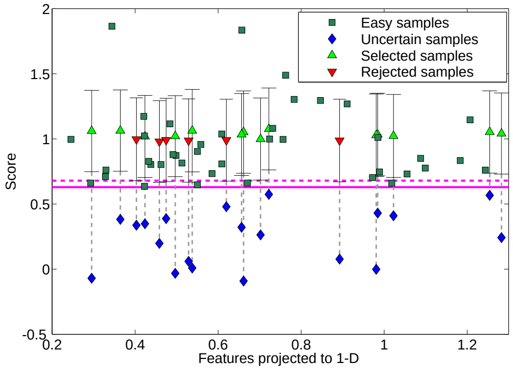

Fig. 3. Sample selection by GPR (see main text for details). The horizontal axis runs on the sample features projected to 1-D for visualization. The triangles are re-scored values from the diamonds, with vertical segments indicating the $\pm 3\sigma_{*,i}$ variance range. The solid horizontal line draws the threshold $\overline{r}$ and the dashed one $\overline{r} + \theta$. An uncertain sample is selected if its variance range is over $\overline{r} + \theta$.

5.2 Gaussian Process Regression (GPR)

Sample selection must collect object examples and counter-examples (background) from a training sequence of target-domain unlabeled images. The examples will be selected from the detections returned by the current detector (*i.e.*, the source-domain one or an intermediate target-domain adapted version of it). The counter-examples can be selected as background windows overlapping little with the detections (*e.g.*, we use a 10% overlapping threshold). Alternatively, images labeled as object-free (weak labeling) can be used for sampling counter-examples. Collecting examples from the target-domain detections following the SPL relies on a GPR as follows.

We define the thresholds $\overline{r}$ and $\underline{r}$, $\overline{r} \geq \underline{r}$, that divide detections into *conservative* and *aggressive* sets. The conservative set, $\mathcal{D}^{T_{\overline{r}}}$, contains the *easy* examples, defined as those detections with score above $\overline{r}$, *i.e.*,

$\mathcal{D}^{T_{\overline{r}}} = \{(\mathbf{x}_i, z_i) : z_i \geq \overline{r}\}$, where z_i is the classification score of detection $\mathbf{x}_i$. False positives are very unlikely in this set. The aggressive set, $\mathcal{D}^{T_{\underline{r}}}$, contains the detections with score above $\underline{r}$, *i.e.*, $\mathcal{D}^{T_{\underline{r}}} = \{(\mathbf{x}_i, z_i) : z_i \geq \underline{r}\}$. The aggressive set minus the conservative one, *i.e.*, $\mathcal{D}^{T_{\underline{r}\setminus\overline{r}}} = \{(\mathbf{x}_i, z_i) : \underline{r} \leq z_i < \overline{r}\}$, is a set of *uncertain* samples. It contains true positives but containing false positives is more likely than for $\mathcal{D}^{T_{\overline{r}}}$. In Fig. 3 the squares are in $\mathcal{D}^{T_{\overline{r}}}$ (easy examples) and the diamonds in $\mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}$ (uncertain samples).

A-SSVM and SA-SSVM assume that the target samples have error-free labels. Thus, assigning a proper class to the uncertain samples is important. Accordingly, we propose to use $\mathcal{D}^{T_{\overline{r}}}$ as confidently classified examples for predicting the scores of the samples in $\mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}$ according to a GPR [39]. In particular, we apply a standard linear regression with Gaussian noise, $z = \mathbf{w}'\Phi(\mathbf{x}) + \eta$, where $\Phi(\mathbf{x})$ is the feature vector, $\mathbf{w}$ is the weight vector and $\eta \sim \mathcal{N}(0, \sigma_z^2)$ is the noise term. In our case, the feature vector consists of the concatenation of the appearance and deformation features of the DPM, *i.e.*, ϕ_a and ϕ_d in Eq. (1). We assume a zero mean Gaussian prior on $\mathbf{w}$, *i.e.*, $\mathbf{w} \sim \mathcal{N}(0, \Sigma)$. We use $\mathbf{X}$ to denote the aggregated column vector input from the observed set $\mathcal{D}^{T_{\overline{r}}}$, and $\mathbf{X}_*$ is the analogous for $\mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}$. The joint density of the observed set and the noise-free function $\mathbf{f}_*$ on the test set $\mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}$ is given by

$$\begin{bmatrix} \mathbf{z} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} K(\mathbf{X},\mathbf{X}) + \sigma_z^2\mathbf{I} & K(\mathbf{X},\mathbf{X}_*) \\ K(\mathbf{X}_*,\mathbf{X}) & K(\mathbf{X}_*,\mathbf{X}_*) \end{bmatrix}\right), \quad (14)$$

where $K()$ is the kernel function for computing the covariance; we use a squared-exponential kernel [39]. The resulting predictive distribution $p(\mathbf{f}_*|\mathbf{X}, \mathbf{y}, \mathbf{X}_*)$ is a Gaussian with mean and covariance defined as:

$$\begin{aligned} \overline{\mathbf{f}}_{*,i} = &\quad K(\mathbf{x}_{*,i}, \mathbf{X})[K(\mathbf{X},\mathbf{X}) + \sigma_z^2\mathbf{I}]^{-1}\mathbf{y}, \\ \sigma_{*,i} = &\quad K(\mathbf{x}_{*,i}, \mathbf{x}_{*,i}) \\ &\quad -K(\mathbf{x}_{*,i}, \mathbf{X})[K(\mathbf{X},\mathbf{X}_*) + \sigma_z^2\mathbf{I}]^{-1}K(\mathbf{X}, \mathbf{x}_{*,i}). \end{aligned} \quad (15)$$

In analogy with [9], we use variables υ_i indicating if the i^{th} sample is selected ($\upsilon_i = 1$) or not ($\upsilon_i = 0$):

$$\upsilon_i = \begin{cases} 1, \left(\overline{\mathbf{f}}_{*,i} - 3\sigma_{*,i} \geq \overline{r} + \theta, \mathbf{x}_{*,i} \in \mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}\right) \vee \mathbf{x}_i \in \mathcal{D}^{T_{\overline{r}}}, \\ 0, \text{otherwise} \end{cases} \quad (16)$$

We use $\overline{\mathbf{f}}_{*,i} - 3\sigma_{*,i}$ to ensure that the confidence of the predicted output score is higher than 99.7%. The parameter $\theta > 0$ controls the degree of the acceptance for the samples in $\mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}$ and we use $\theta = 0.05$ in practice. The process is illustrated in Fig. 3.

5.3 Self-adaptive DPM Algorithm

Our self-adaptive DPM is sketched in Alg. 2. At each iteration, we apply GPR to $\mathcal{D}^{T_{\underline{r}}}, \mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}$ and compute the υ_i. Supervised DA-DPM (A-SSVM or SA-SSVM) relies on the easy examples and the selected uncertain ones ($\upsilon_i = 1$). Since $\overline{r}$ decreases by a factor of $\Delta > 0$ at each iteration (step 6), $\mathcal{D}^{T_{\overline{r}}}$ grows and more difficult examples are progressively selected. The training process runs until $\overline{r}$ reaches $\underline{r}$ or the objective function (*i.e.*, (6) for A-SSVM or (11) for SA-SSVM) cannot be decreased below a tolerance ϵ. We remark that our self-adaptive DPM computes the υ_i at step 3 in an explicit way (Eq. (16)), while in the SPL proposal of [9] solving a biconvex optimization problem is required for computing them (see Eq. (4) in [9]). In particular, [9] runs an alternative convex search (ACS).

Algorithm 2 Self-adaptive DPM

```
Input: $\mathbf{w}^S, \underline{r}, \overline{r}, \theta, \Delta, \epsilon$.
Output: $\mathbf{w}$
0: $\mathbf{w} \leftarrow \mathbf{w}^S$
1: Repeat
2: Collect $\mathcal{D}^{T_{\underline{r}}}, \mathcal{D}^{T_{\overline{r}}}$ in the target domain using $\mathbf{w}$.
3: Apply GPR to $\mathcal{D}^{T_{\overline{r}}}, \mathcal{D}^{T_{\underline{r}\setminus\overline{r}}}$ and update $\upsilon_i$ by (16).
4: Update $\mathbf{h}_i^* = \arg\max_{\mathbf{h}} \mathbf{w}'\Phi(\mathbf{x}_i, \mathbf{h})$.
5: Update $\mathbf{w}$ by fixing the hidden variables to $\mathbf{h}_i^*$
   and solving the corresponding DA optimization
   problem: A-SSVM (6) or SA-SSVM (11).
6: $\overline{r} \leftarrow \max(\overline{r} - \Delta,\ \underline{r})$.
7: Until $\overline{r} = \underline{r}$ or the objective function ((6) or (11))
   cannot be decreased below tolerance $\epsilon$.
```

6 Experiments

We built our DA framework based on the latest release of the DPM, *i.e.*, the DPM 5.0 framework [40]. We evaluate first the accuracy of our supervised DA-DPM proposals. We evaluate our self-adaptive DPM, showing its accuracy with and without the GPR.

As we are interested in pedestrian detection, all the experiments rely on public pedestrian datasets. We adapt a generic person detector from the PASCAL VOC 2007 Person dataset to the INRIA pedestrian dataset. In this case, the domain shift is mainly due to the differences in the data distributions in terms of viewpoints and poses. Moreover, following [1], we adapt a pedestrian classifier learned with synthetic data (virtual world) to operate on real-world images.

We use the Caltech evaluation framework [12] following the *reasonable* setting criterion, *i.e.*, detectable pedestrians are those taller than 50 pixels and without heavy occlusion. Thus, we assess the accuracy of a particular pedestrian detector by using per-image evaluation, *i.e.*, computing curves depicting the trade-off between *miss rate* and *number of false positives per image* (FPPI) on a logarithmic scale. For single *detection accuracy* we use one minus the average miss rate in the $[10^{-2}, 10^{0}]$ FPPI range. Moreover, since the target domains are sampled for collecting training data, each DA experiment is repeated five times.

As in [1], to compare our proposals with the state-of-the-art we apply a *paired Wilcoxon test* [41] on the accuracy measures collected from the experiments.

6.1 Implementation details

Our DA proposals can be seen as plug-ins for the DPM framework. The solver for optimizing A-SSVM and SA-SSVM is based on the quasi-Newton LBFGS method [42] as in the DPM 5.0 framework [40]. We also use a data-mining procedure to maintain a feature cache with the support vectors, and the minimization of the objective functions is restricted to the cache. Note that in the CCCP the data mining of the examples (*i.e.*, the pedestrians in these experiments) is performed on a constrained set. In particular, the valid examples detected by the model in each iteration are required to have at least 70% overlap with the ground truth bounding box. In the self-adaptive case there are no ground truth bounding boxes, so we designate the detected bounding box with the highest score as *ground truth*. The SPL is implemented to replace the original CCCP. In contrast to the CCCP which uses the entire dataset at each iteration, the SPL first takes the discovered easy examples and gradually collects the difficult ones. Moreover, we use the implementation of [39] to compute the GPR. For the parameters in Alg. 2, we fix $\underline{r}$ by -0.5. The initial value of $\bar{r}$ is estimated in the source domain, which ensures high detection accuracy ($> 90\%$), and we set $\Delta = 0.05$.

In practice, the optimization of the DA converges very fast. We use only two iterations for CCCP, and at each iteration we do data mining twice. For the SPL, we iterate three times and apply data mining twice in each loop. Note that our DA methods only require very few training examples, thus the training is very fast. For instance, training a DA-DPM with 100 pedestrians and $1,000$ negative images takes less than 20 minutes in a $3.60GHz \times 4$ modern desktop PC.

6.2 Experiment Settings

6.2.1 Datasets

Virtual-world pedestrians. We use a video game to collect realistic images from virtual urban scenarios as in [43]. We collect images at a resolution of 640×480 pixels, which contain objects from six principal classes under different illumination conditions, namely road, tree, building, vehicle, traffic sign and pedestrian. These images have an associated pixel-wise segmentation of the contained pedestrians, which allows us to automatically extract their ground truth bounding boxes. Overall, we obtained $2,000$ pedestrian samples for training, covering typical pedestrians views and poses as seen from an on-board vision system.

PASCAL VOC person. We use the PASCAL VOC 2007 Person dataset which contains a large number of general person images, including outdoor vertical full body persons and indoor half body ones, all of them with different poses and some of them highly occluded. The DPM person detector trained on VOC 2007 dataset has six components and eight parts (see the prior model in Fig. 2) and is publicly available from [40]. The components are trained with person samples of different aspect ratios and views.

TABLE 2
Different types of learned classifiers.

SRC	Trained with labeled source data.
TAR	Trained with labeled target data. For a fair comparison, we initialize the structure of TAR with the source DPM.
MIX	Trained with source and target labeled data.
(S)A-SSVM	Adapted by following Alg. 1, *i.e.* with a source model and labeled target data.
PMT-SSVM	A-SSVM variant that we have developed by extending the PMT-SVM [26] for DA of the DPM.
SA-SSVM-C	SA-SSVM variant where the DPM parameters are partitioned at component level.
U-SA-SSVM	Adapted with target images where the pedestrians are unlabeled, Alg. 2 is followed setting the SA-SSVM case and relying on the threshold $\bar{r}$ to select easy examples (GPR not applied).
U-SA-SSVM-GPR	As U-SA-SSVM but using the GPR.

Real-world pedestrians. We use popular pedestrian detection datasets, namely INRIA [44], ETH [45], KIT [46], Caltech [12] and CVC (N.02) [47]. Except INRIA, the other datasets are image sequences taken from on-board cameras. In particular, the ETH dataset contains three sub-sequences from on-board cameras, namely Bahnhof, Jelmoli, and Sunny Day. These sequences are taken in different scenarios and they are named here as ETH0, ETH1 and ETH2 respectively. In all cases pedestrians are standing, either walking or stopped. Caltech and INRIA have separate training and testing sets, for CVC we use the first four sequences for training and the other ten for testing, while for KIT and ETH training images are obtained by sampling the respective sequences keeping the remaining of the sequences for testing. It is worth mentioning that INRIA and Caltech can be considered as weakly labeled since their training sets are split into pedestrian-free images and images with annotated pedestrians. This is not the case for ETH, KIT and CVC.

6.2.2 Learned classifiers

We train the types of classifiers shown in Table 2. For TAR, MIX, A-SSVM, SA-SSVM, SA-SSVM-C, and PMT-SSVM we use 100 randomly selected target-domain training pedestrians. For MIX the full source dataset is used too. For U-SA-SSVM and U-SA-SSVM-GPR, we use 150 randomly selected target-domain training images which contain at least 100 pedestrians, but without considering manually annotated bounding boxes. The number of target-domain training images from which to collect background windows is fixed to 1000. For INRIA and Caltech these are pedestrian-free images. For the rest of the datasets these images contain pedestrians, thus background windows are obtained as those overlapping less than a 10% with the annotated pedestrian bounding boxes. The parameter γ in SA-SSVM is fixed by cross validation for all the experiments ($\gamma = 0.08$).

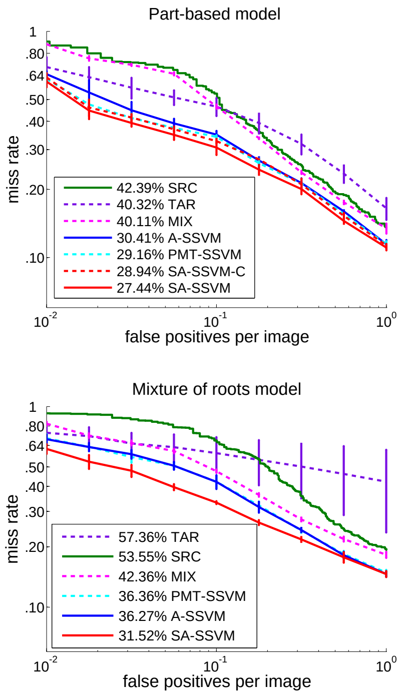

Fig. 4. Results of adapting PASCAL VOC 2007 DPM person detector to work on the INRIA pedestrian dataset. Percentages correspond to the average miss rate within the plotted FPPI range. Vertical segments illustrate the variance over five runs per experiment.

6.3 Experiments on supervised DA-DPM

6.3.1 PASCAL to INRIA

We adapt the general person DPM (six components, eight parts) based on PASCAL dataset to detect pedestrians in INRIA testing images. Figure 4 shows the accuracy of the different detectors. We evaluated pure mixture of roots (no parts) and part-based models.

6.3.2 Virtual to ETH, KIT, Caltech, and CVC

We adapt a pedestrian DPM (three components, five parts) trained with virtual-world data to operate on real-world datasets. For completeness, we include the original HOG/Lin-SVM holistic detector [44] and the DPM state-of-the-art one (Lat-SVM) [4] (three components, eight parts). Lin-SVM and Lat-SVM training uses the full INRIA training set. In fact, a widespread approach consists in training the classifiers using the INRIA training set and then testing on other datasets [12]. Accordingly, we have included analogous experiments. In particular, A-SSVM(*) and SA-SSVM(*) stand for adaptation to INRIA as a sort of *intermediate* domain. However, our interest is the *direct* adaptation to the *final* real-world domain, *i.e.*, to ETH0, ETH1, ETH2, KIT, Caltech, or CVC. The accuracy results of our proposals using intermediate and direct adaptation are listed in Table 3. In Fig. 5, we complete the accuracy results based on direct adaptation. Finally, Table 4 shows the adaptation accuracy for the pure mixture of roots (three roots, no parts) and the part-based models.

6.3.3 Discussion

According to Figs. 4 and 5, TAR shows poor accuracy and high variability, which is due to the low number of target pedestrians used for training. Even SRC performs clearly better than TAR in all cases except the PASCAL-to-INRIA part-based one, which is because the person poses on the PASCAL dataset are too different from those in the INRIA one, while the virtual-world (source) data covers poses similar to the real-world (target) data. MIX clearly outperforms SRC and TAR (INRIA Mixture of Roots, ETH, KIT, CVC) or at least does no harm (INRIA part-based, Caltech). These observations agree with the results of [1]. SSVM adaptations clearly outperform SRC and TAR. The same happens for MIX, except for the ETH1 case where PMT-SSVM and MIX perform similarly. However, we remark that, contrarily to SSVM adaptations, MIX requires re-training with the source data. Thus, we focus on analyzing SSVM DA.

Regarding SA-SSVM, we assessed whether to make it aware of the DPM structure of parts or of components. We used the PASCAL-to-INRIA adaptation problem since the domain shift is not only due to the use of different sensors, as in the virtual-to-real case, but also to large pose differences as mentioned before. In Fig. 4 top, we see that SA-SSVM accuracy (part aware) is 1.5 points better than SA-SSVM-C accuracy (component aware). Thus, in the rest of SA-SSVM experiments we used the part-aware setting.

Table 3 shows how the intermediate adaptations, A-SSVM(*) and SA-SSVM(*), outperform the SRC model in most of the cases. In fact, for CVC the accuracy of SA-SSVM matches Lat-SVM, while for ETH2, KIT and Caltech SA-SSVM clearly outperforms Lat-SVM ($\sim$ $12, 4,$ and 6 points, respectively), and for ETH1 and ETH2 Lat-SVM is still better ($\sim$ 7 and 6 points, respectively). We remind that Lat-SVM is trained with the full INRIA training set, *i.e.*, using $1,208$ pedestrians, while SA-SSVM only uses 100 ($\sim 8\%$) for the adaptation from the virtual-world (source) model. In any case, we see that the direct adaptations, A-SSVM and SA-SSVM, clearly outperform their intermediate counterparts, especially SA-SSVM. Thus, for the rest of experiments we assumed the direct adaptation setting.

The accuracy of A-SSVM and SA-SSVM have also been assessed for pure mixture-of-roots models. Since parts are not available, for SA-SSVM the component-aware strategy is used. In Fig. 4 bottom, we see the PASCAL-to-INRIA case, and in Table 4 the virtual-to-real one. Observe that A-SSVM and SA-SSVM clearly outperform SRC (Mix. of Roots Δ_a and Δ_{sa} in Table 4 show the respective accuracy gains for the virtual-to-real case), thus there is domain adaptation. However, as can be seen in Table 4, deformable part-based models achieve a higher relative gain than the mixture-of-roots models for the virtual-to-real case (Part-based

TABLE 3
DA from virtual to real world using different models (see main text). Average miss rate (%) is shown.

Method/Dataset	ETH0	ETH1	ETH2	KIT	Caltech	CVC
Lin-SVM	59.6	64.8	74.3	76.5	68.5	63.9
Lat-SVM	49.6	48.9	55.2	58.0	63.3	41.8
Method/Dataset	ETH0	ETH1	ETH2	KIT	Caltech	CVC
SRC	63.1	58.8	48.5	53.3	64.9	45.9
A-SSVM(*)	59.6±1.3	56.4±0.9	45.0±1.5	51.5±1.4	59.5±1.8	43.2±0.8
SA-SSVM(*)	57.0±1.2	54.9±0.6	43.2±1.7	53.7±0.4	57.3±0.6	41.0±0.9
A-SSVM	56.0±0.5	53.2±0.5	37.6±0.8	45.0±0.5	58.6±0.6	32.5±0.7
SA-SSVM	**54.7**±0.3	**51.5**±0.4	**35.5**±0.3	**44.2**±0.7	**56.1**±0.6	**30.8**±0.3

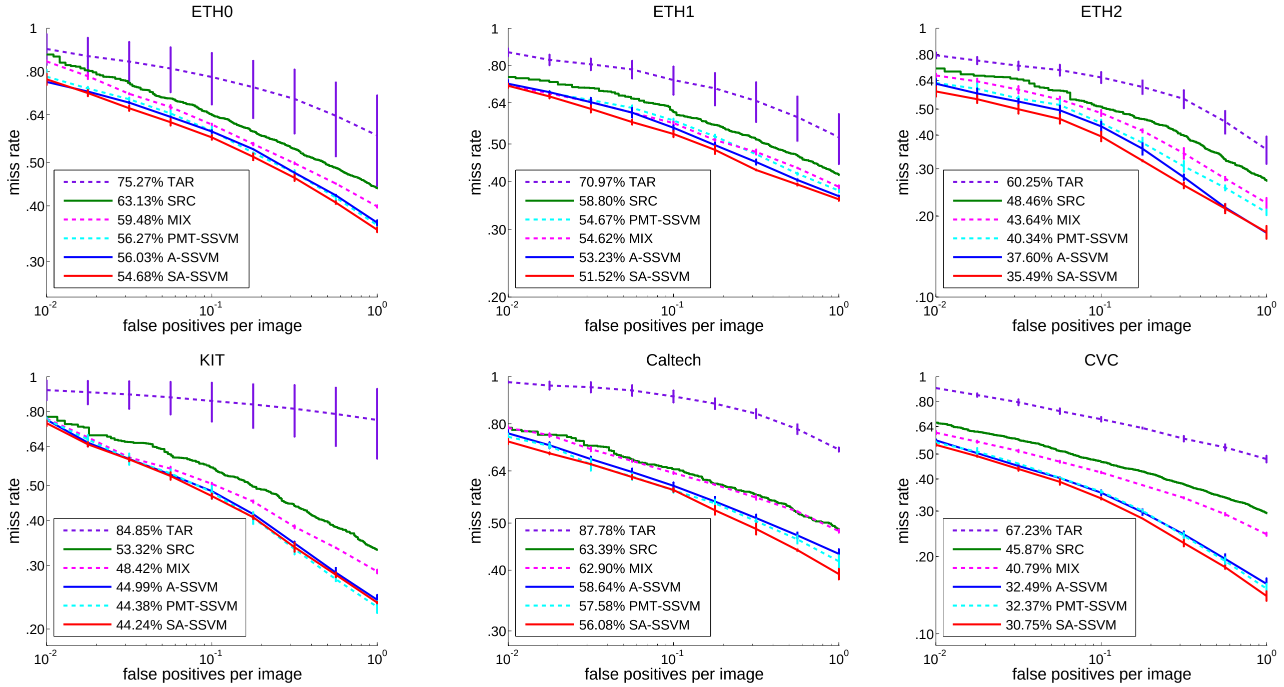

Fig. 5. Supervised adaptation of DPM from virtual world to specific real-world scenes.

Δ_a and Δ_{sa} in Table 4 show the corresponding accuracy gains, computed from SRC, A-SSVM and SA-SSVM of Table 3). The PASCAL-to-INRIA case is an exception, which is due to the fact that person views at PASCAL dataset are quite different than the ones in INRIA, and therefore strong adaptation can be expected already at component level. Note that in the virtual-to-real case the domain shift is mainly due to the *sensor type* but views and poses of the source and target domains are very similar. In fact, the same reason explains why A-SSVM and SA-SSVM report similar accuracy for the mixture-of-roots adaptation of the virtual-to-real case (Table 4), while SA-SSVM outperforms A-SSVM by almost 5 points in the PASCAL-to-INRIA case (Fig. 4 bottom). In any case, in absolute terms part-based adaptation (either with A-SSVM or SA-SSVM) clearly outperforms pure mixture-of-roots adaptation. Just comparing SA-SSVM from Tables 3 and 4, we see gains ranging from ~ 11 points for ETH0 to ~ 27 points for CVC.

At this point, we see that A-SSVM, SA-SSVM (part-aware), and PMT-SSVM direct adaptations operating on full DPM models clearly are the best performing methods. Accordingly, we focus on statistical comparisons on them. We use a paired Wilcoxon test by taking into account their respective part-based results for both PASCAL-to-INRIA and virtual-to-real adaptation problems. In this test, when comparing two adaptation methods, the null hypothesis is that they are equal. The hypothesis can be rejected if the p-value of the test is below 0.05 (the p-value running from 0 to 1). Since we test adaptations for seven datasets and each experiment is repeated five times, for each test run we have 35 pairs, which allows us to draw confident conclusions from the paired Wilcoxon test.

The conclusions are: (1) A-SSVM and PMT-SSVM perform equally (p-value = 0.63); (2) SA-SSVM outperforms A-SSVM (p-value = $2.5e^{-07}$) by 1.8 points; and (3) SA-SSVM outperforms PMT-SSVM (p-value = $6.6e^{-07}$) by 1.6 points. Thus, part-aware DA outperforms strategies that ignore model structure.

TABLE 4
DA from virtual to real world using different models (see main text). Average miss rate (%) is shown.

Mix. of Roots	ETH0	ETH1	ETH2	KIT	Caltech	CVC
SRC	69.1	70.3	65.8	70.8	72.6	64.9
A-SSVM	67.1±0.2	67.2±0.5	54.9±0.8	64.2±0.3	71.3±1.0	57.6±0.9
SA-SSVM-C	66.5±0.8	66.5±1.0	54.4±0.3	63.5±0.9	71.2±0.6	57.9±0.5
Δ_a	2.0±0.2	3.1±0.5	10.9±0.8	6.6±0.3	1.3±1.0	7.3±0.9
Δ_{sa}	2.6±0.8	3.8±1.0	11.4±0.3	7.2±0.9	1.4±0.6	7.0±0.5

Part-based	ETH0	ETH1	ETH2	KIT	Caltech	CVC
Δ_a	7.1±0.5	5.6±0.5	10.9±0.8	8.3±0.5	6.3±0.6	13.4±0.7
Δ_{sa}	8.4±0.3	7.3±0.4	13.0±0.3	9.1±0.7	8.8±0.6	15.1±0.3

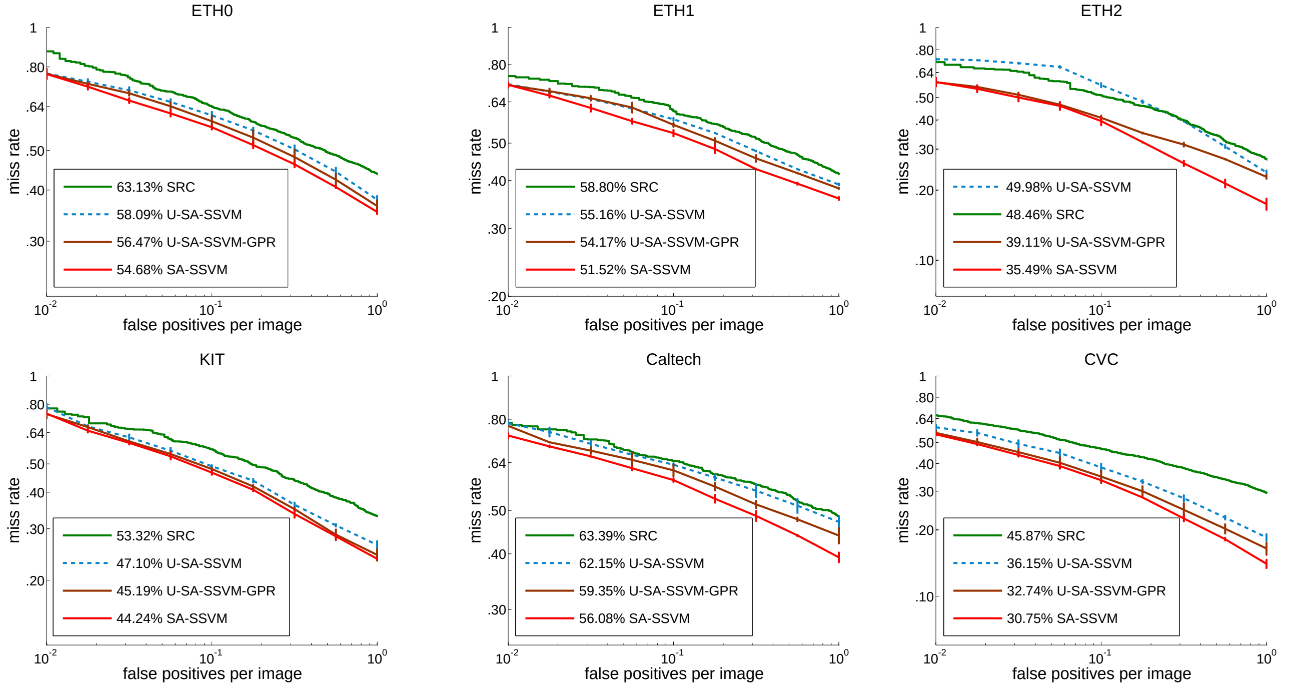

Fig. 6. Self-adaptive DPM from virtual world to specific real-world scenes.

6.4 Experiments on Self-adaptive DPM

We evaluate the self-adaptive DPM in the virtual-to-real case. We assume that real world predestrains come without bounding boxes. For Caltech there is a set of pedestrian-free images, while for ETH, KIT, and CVC this is not the case. We restrict our experiments to SA-SSVM since it has shown the best accuracy in the supervised case. Moreover, we evaluate the self-adaptive method with and without GPR.

6.4.1 Virtual to ETH, KIT, Caltech, and CVC

Fig. 6 shows the results on all the testing datasets. The paired Wilcoxon test shows that SA-SSVM improves on U-SA-SSVM-GPR by 2.1 points (p-value = $3e^{-06}$). This is mainly due to the difficulty that the self-adaptive DPM (U-SA-SSVM-GPR) faces for discovering target-domain pedestrians without introducing label noise such as false positive detections. In some datasets, the accuracy of the self-adaptive DPM is very close to the supervised DA-DPM (SA-SSVM), *e.g.*, in KIT and ETH0 less than two points. Analogously, the paired Wilcoxon test shows that U-SA-SSVM-GPR improves on U-SA-SSVM by 3.0 points (p-value = $6e^{-06}$). This demonstrates the effectiveness of using GPR rather than a fixed threshold.

6.4.2 Self-paced Learning with GPR

Fig. 7 illustrates the GPR-based pedestrian selection for three iterations. At each step the classifier is updated. Thus, all samples are re-scored at the next iteration (Alg. 2, step 2). The bounding boxes (BBs) drawn with continuous lines (yellow) denote easy examples (in $\mathcal{D}^{\mathcal{T}_{\mathcal{T}}}$), *i.e.*, the observations for the GPR. The BBs drawn with discontinuous lines are the uncertain detections (in $\mathcal{D}^{\mathcal{T}_{\mathcal{L}\setminus\mathcal{T}}}$). The light discontinuous lines (green) show the selected detections after the GPR ($\upsilon_i = 1$), while the dark discontinuous lines (red) denote the rejected ones ($\upsilon_i = 0$). All pedestrians detected in the first iteration, including the one initially rejected, are either classified as easy or selected (both types are the input for steps $4-5$ of Alg. 2) in the last iteration. In fact, two new detections are collected.

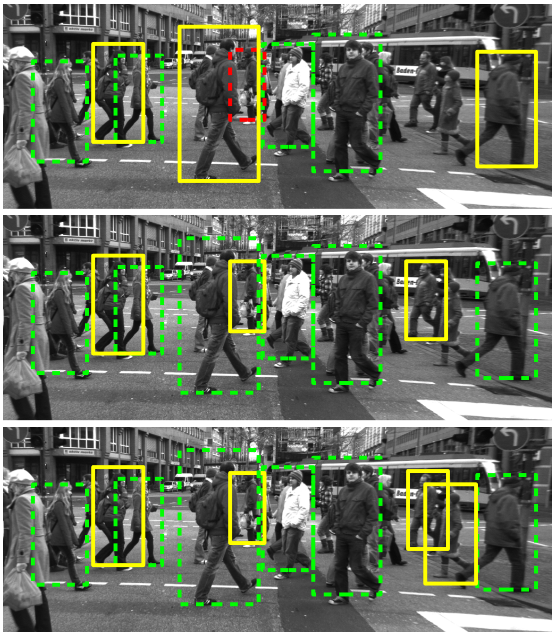

Fig. 7. Sample selection in U-SA-SSVM-GPR. See 6.4.2 for a complete explanation.

7 Conclusions

DA of DPM-based object detectors is of paramount interest for preserving their accuracy across different domains. Accordingly, we have presented two supervised DA-DPM methods (A-SSVM and SA-SSVM), which can be integrated into a self-adaptive DPM for new unlabeled or weakly labeled domains. Our DA methods do not require revisiting the source-domain data for adaptation, and only relatively little annotated data from the target domain is required to boost detection accuracy. In the case of the self-adaptive technique, samples from the target domain are automatically collected to adapt the model without any supervision, *i.e.* avoiding the need of human intervention. We have tested our proposals in the context of pedestrian detection performing a total of 384 train-test runs. Overall, two types of adaptation are evaluated: both from synthetic and general person domains, to real-world pedestrian images. As future work we plan three directions. First, to improve our current sample selection in the target domain. Second, to extend the current methods to work as online adaptive learning techniques. Third, to develop better DA models based on deep representations.

Acknowledgments

This work is supported by the Spanish projects TRA2011-29454-C03-01 and TIN2011-29494-C03-02, Jiaolong Xu's Chinese Scholarship Council (CSC) grant No.2011611023 and Sebastian Ramos' FPI Grant BES-2012-058280. Additionally, we thank Xu Hu from Oregon State Univ. for many fruitful discussions, and Dr. Andrew D. Bagdanov from the Computer Vision Center for helping us to improve paper readability.

References

[1] D. Vázquez, A. López, J. Marín, D. Ponsa, and D. Gerónimo, "Virtual and real world adaptation for pedestrian detection," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 797–809, 2014.

[2] A. Torralba and A.A. Efros, "Unbiased look at dataset bias," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 2011.

[3] J. Quiñonero-Candela, M. Sugiyama, A. Schwaighofer, and N.D. Lawrence, Eds., *Dataset shift in machine learning*, ser. Neural Information Processing. The MIT Press, 2008.

[4] 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.

[5] C.-N. J. Yu and T. Joachims, "Learning structural SVMs with latent variables," in *Int. Conf. on Machine Learning*, Montreal, Quebec, 2009.

[6] A. Vedaldi and A. Zisserman, "Structured output regression for detection with partial truncation," in *Advances in Neural Information Processing Systems*, Vancouver, Canada, 2009.

[7] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, "Latent hierarchical structural learning for object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, 2010.

[8] J. Yang, R. Yan, and A. Hauptmann, "Cross-domain video concept detection using adaptive SVMs," in *ACM Multimedia*, Augsburg, Germany, 2007.

[9] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, 2010.

[10] 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.

[11] 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.

[12] 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.

[13] X. Cao, Z. Wang, P. Yan, and X. Li, "Transfer learning for pedestrian detection," *Neurocomputing*, vol. 100, no. 0, pp. 51–57, 2013.

[14] J. Pang, Q. Huang, S. Yan, S. Jiang, and L. Qin, "Transferring boosted detectors towards viewpoint and scene adaptiveness." *IEEE Trans. on Image Processing*, vol. 20, no. 5, pp. 1388–400, 2011.

[15] D. Vázquez, A. López, D. Ponsa, and D. Gerónimo, "Interactive training of human detectors," *Multimodal Interaction in Image and Video Applications*, vol. 48, p. 169182, 2013.

[16] D. Vázquez, A. López, and D. Ponsa, "Unsupervised domain adaptation of virtual and real worlds for pedestrian detection," in *Int. Conf. in Pattern Recognition*, Tsukuba, Japan, 2012.

[17] M. Wang and X. Wang, "Automatic adaptation of a generic pedestrian detector to a specific traffic scene," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 2011.

[18] ——, "Transferring a generic pedestrian detector towards specific scenes," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.

[19] J. Xu, D. Vázquez, S. Ramos, A. López, and D. Ponsa, "Adapting a pedestrian detector by boosting LDA exemplar classifiers," in *IEEE Conf. on Computer Vision and Pattern Recognition – Workshop on Ground Truth*, Portland, OR, USA, 2013.

[20] J. Donahue, J. Hoffman, E. Rodner, K. Saenko, and T. Darrell, "Semi-supervised domain adaptation with instance constraints," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Portland, Oregon, USA, 2013.

[21] A. Bergamo and L. Torresani, "Exploring weakly-labeled web images to improve object classification: a domain adaptation approach," in *Advances in Neural Information Processing Systems*, Vancouver, BC, Canada, 2010.

[22] A. Prest, C. Leistner, J. Civera, C. Schmid, and V. Ferrari, "Learning object class detectors from weakly annotated video," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.

[23] H. D. III, "Frustratingly easy domain adaptation," in *Meeting of the Association for Computational Linguistics*, Prague, Czech Republic, 2007.

[24] W. Jiang, E. Zavesky, C. Shih-Fu, and A. Loui, "Cross-domain learning methods for high-level visual concept classification," in *IEEE Int. Conf. on Image Processing*, San Diego, CA, USA, 2008.

[25] F. Orabona, C. Castellini, B. Caputo, A. Fiorilla, and G. Sandini, "Model adaptation with least-squares svm for adaptive hand prosthetics," in *IEEE Int. Conf. on Robotics and Automation*, Kobe, Japan, 2009.

[26] Y. Aytar and A. Zisserman, "Tabula rasa: Model transfer for object category detection," in *Int. Conf. on Computer Vision*, 2011.

[27] L. Duan, I. Tsang, D. Xu, and S. Maybank, "Domain transfer svm for video concept detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami Beach, FL, USA, 2009.

[28] R. Gopalan, R. Li, and R. Chellappa, "Domain adaptation for object recognition: An unsupervised approach," in *Int. Conf. on Computer Vision*, Barcelona, Spain, 2011.

[29] B. Gong, Y. Shi, F. Sha, and K. Grauman, "Geodesic flow kernel for unsupervised domain adaptation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Providence, RI, USA, 2012.

[30] F. Mirrashed, V. Morariu, S. Behjat, R. Feris, and L. Davis, "Domain adaptive object detection," in *WACV*, Washington, DC, USA, 2013.

[31] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," in *International Joint Conference on Artificial Intelligence*, Pasadena, California, USA, 2009.

[32] V. Jain and E. Learned-Miller, "Online domain adaptation of a pre-trained cascade of classifiers," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Colorado Springs, CO, USA, 2011.

[33] K. Saenko, B. Hulis, M. Fritz, and T. Darrel, "Adapting visual category models to new domains," in *European Conf. on Computer Vision*, Hersonissos, Heraklion, Crete, Greece, 2010.

[34] R. Girshick, "From rigid templates to grammars: Object detection with structured models," Ph.D. dissertation, The University of Chicago, Chicago, IL, USA, 2012.

[35] C. N. Yu, "Improved learning of strucutral support vector machines: Training with latent variables and non-linear kernels," Ph.D. dissertation, Cornell University, Ithaca, NY, USA, 2011.

[36] Y. Aytar and A. Zisserman, "Enhancing exemplar SVMs using part level transfer regularization," in *British Machine Vision Conference*, Surrey, UK, 2012.

[37] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Int. Conf. on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006.

[38] K. Tang, V. Ramanathan, F.-F. Li, and D. Koller, "Shifting weights: Adapting object detectors from image to video." in *Advances in Neural Information Processing Systems*, Lake Tahoe, Nevada, USA, 2012.

[39] C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

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

[41] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bulletin*, vol. 1, no. 6, pp. 80–83, 1945.

[42] M. Schmidt, "Minconf - projection methods for optimization with simple constraints in matlab," http://www.di.ens.fr/ mschmidt/Software/minConf.html.

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

[44] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, San Diego, CA, USA, 2005.

[45] C. Wojek, S. Walk, and B. Schiele, "Multi-cue onboard pedestrian detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami Beach, FL, USA, 2009.

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

[47] 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. 583–595, 2010.

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

Sebastian Ramos received the B.Sc. degree in Electronic Engineering from the National University of Colombia in 2013. He is currently working towards the Ph.D. degree in Computer Science at the Computer Vision Center (CVC) in Universitat Autònoma de Barcelona (UAB). He was awarded a German scholarship to visit Technical University Munich from 2010 to 2012. During that period he was part of the Institute of Automatic Control Engineering (LSR) at TU Munich and the Robotics Technologies Group at Siemens AG Munich. His research interests include visual scene understanding for autonomous systems, transfer learning and discrete learning and inference. He is a student member of the IEEE.

David Vázquez received the B.Sc. degree in Computer Science from the Universitat Autònoma de Barcelona (UAB) in 2008. He received his M.Sc. in Computer Vision and Artificial Intelligence in 2009 and his Ph.D. degree in 2013 at the Computer Vision Center (CVC/UAB). He is currently a research scientist at CVC. His research interests include pedestrian detection, virtual worlds, domain adaptation and active learning. He is a member of the IEEE.

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 has been giving 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. He is a member of the IEEE.