

From Virtual to Real World Visual Perception using Domain Adaptation – The DPM as Example

Antonio M. López

Computer Vision Center (CVC) and Dpt. Ciències de la Computació (DCC),
Universitat Autònoma de Barcelona (UAB)

antonio@cvc.uab.es

Jiaolong Xu

CVC and DCC, UAB

jiaolong@cvc.uab.es

José L. Gómez

CVC and DCC, UAB

jlgomez@cvc.uab.es

David Vázquez

CVC and DCC, UAB

dvazquez@cvc.uab.es

Germán Ros

CVC and DCC, UAB

gros@cvc.uab.es

December 30, 2016

Abstract

Supervised learning tends to produce more accurate classifiers than unsupervised learning in general. This implies that training data is preferred with annotations. When addressing visual perception challenges, such as localizing certain object classes within an image, the learning of the involved classifiers turns out to be a practical bottleneck. The reason is that, at least, we have to frame object examples with bounding boxes in thousands of images. A priori, the more complex the model is regarding its number of parameters, the more annotated examples are required. This annotation task is performed by human oracles, which ends up in inaccuracies and errors in the annotations (*aka* ground truth) since the task is inherently very cumbersome and sometimes ambiguous. As an alternative we have pioneered the use of *virtual worlds* for collecting such annotations automatically and with high precision. However, since the models learned with virtual data must operate in the real world, we still need to perform *domain adap-*

tation (DA). In this chapter we revisit the DA of a *deformable part-based model* (DPM) as an exemplifying case of virtual- to real-world DA. As a use case, we address the challenge of *vehicle detection* for driver assistance, using different publicly available virtual-world data. While doing so, we investigate questions such as: how does the domain gap behave due to virtual-vs-real data with respect to dominant object appearance per domain, as well as the role of photo-realism in the virtual world.

1 Need for Virtual Worlds

Since the 90's, *machine learning* has been an essential tool for solving *computer vision* tasks such as image classification, object detection, instance recognition, and (pixel-wise) semantic segmentation, among others [76, 13, 17, 54, 4]. In general terms, the best performing machine learning algorithms for these tasks are *supervised*; in other words, not only the raw data is required, but also *annotated information*, *i.e.* *ground truth*, must be provided to run the training

protocol. Collecting the annotations has been based on human oracles and collaborative software tools such as Amazon's Mechanical Turk [48], LabelMe [53], etc. It is known, that human-based annotation is a cumbersome task, with ambiguities, and inaccuracies. Moreover, not all kinds of ground truth can be actually collected by relying on human annotators, *e.g.* pixel-wise optical flow and depth.

The non-expert reader can have a feeling of the annotation effort by looking at Fig. 1, where we can see two typical annotation tasks, namely bounding box (BB) based object annotations, and delineation of semantic contours between classes of interest. In the former case, the aim is to develop an object detector (*e.g.* a vehicle detector); in the latter, the aim is to develop a pixel-wise multi-class classifier, *i.e.* to perform the so-called semantic segmentation of the image.

With the new century, different datasets were created with ground truth and put publicly available for research. Providing a comprehensive list of them is out of the scope of this chapter, but we can cite some meaningful and pioneering examples related to two particular tasks in which we worked actively, namely *pedestrian detection* and *semantic segmentation*; both in road scenarios for either *advanced driver assistance systems* (ADAS) or *autonomous driving* (AD). One example is the Daimler Pedestrian dataset [16], which includes 3,915 BB-annotated pedestrians and 6,744 pedestrian-free images (*i.e.* image-level annotations) for training, and 21,790 images with 56,492 BB-annotated pedestrians for testing. Another example corresponds to the pixel-wise class ground truth provided in [7] for urban scenarios; giving rise to the well-known *CamVid* dataset which considers 32 semantic classes (although only 11 are usually considered) and includes 701 annotated images, 300 normally used for training and 401 for testing. A few years after, the KITTI Vision Benchmark Suite [20] was an enormous contribution for the research focused on ADAS/AD given the high variability of the provided synchronized data (stereo images, LIDAR, GPS) and ground truth (object bounding boxes, tracks, pixel-wise class, odometry).

In parallel to these annotation efforts and the corresponding development of new algorithms (*i.e.* new

human-designed features, machine learning pipelines, image search schemes, etc.) for solving computer vision tasks, *deep learning* was finding its way to become the powerful tool that is today for solving such tasks. Many researchers would point out [30] as a main breakthrough, since deep *convolutional neural networks* (CNNs) showed an astonishing performance in the data used for the *ImageNet Large-Scale Visual Recognition Challenge* (ILSVRC). ImageNet [14] contains over 15 million of human-labeled (using Mechanical Turk) high-resolution images of roughly 22,000 categories. Thus, ImageNet was a gigantic human annotation effort. ILSVRC uses a subset of ImageNet with about 1,000 images of 1,000 categories; overall, about 1.2M images for training, 50,000 for validation, and 150,000 for testing. Many deep CNNs developed today rely on an ImageNet pre-trained deep CNN which is modified or fine-tuned to solve a new task or operate in a new domain. The research community agrees in the fact that, in addition to powerful GPU hardware to train and test deep CNNs, having a large dataset with ground truth such as ImageNet is key for their success. In this line, more recently, it was released MS COCO dataset [34], where per-instance object segmentation is provided for 91 object types on 328,000 images, for a total of 2.5M of labeled instances.

As a matter of fact, in the field of ADAS/AD we would like to have datasets with at least the variety of information sources of KITTI and the ground truth size of ImageNet/COCO. However, when looking at the ground truth of KITTI in quantitative terms, we can see that individually they are in the same order of magnitude than other ADAS/AD-oriented publicly available datasets (*e.g.* see the number of pedestrians BBs of KITTI and Daimler datasets, and the number of pixel-wise annotated images of KITTI and CamVid). A proof of this need is the recently released Cityscapes dataset [12] which tries to go beyond KITTI in several aspects. For instance, it includes 5,000 pixel-wise annotated (stereo) images covering 30 classes and per-instance distinction, with GPS, odometry and ambient temperature as meta-data. In addition, it includes 20,000 more images but where the annotations are coarser regarding the delineation of the instance/class contours. This kind of



Figure 1: Ground truth obtained by human annotation: *left*) framing the rectangular bounding box (BB) of vehicle instances; *right*) delineating the contours (silhouettes) between the different classes of interest contained in the image, even at instance level.

dataset is difficult to collect since driving through 50 cities covering several months and weather conditions was required. Moreover, providing such a ground truth can take from 30 to 90 minutes per image for a human oracle in case of fine-grained annotations and depending on the image content.

For the semantic segmentation task, Cityscapes goes one order of magnitude beyond KITTI and CamVid. However, it is far from the annotation numbers of ImageNet and MS COCO. The main reason is two-fold. On the one hand, data collection itself, *i.e.* Cityscapes images are collected from on-board systems designed for ADAS/AD not just downloaded from an internet source; moreover, metadata such as GPS and vehicle odometry is important, not to mention the possibility of obtaining depth from stereo. On the other hand, the annotations must be more precise since ultimately the main focus of ADAS/AD is on reducing traffic accidents. In any case, as we mentioned before, other interesting ground truth types are not possible or really difficult to obtain by human annotation, *e.g.* pixel-wise optical flow and depth (without active sensors); but eventually these are important cues for ADAS/AD based on visual perception.

In this ADAS/AD context, and due to the difficulties and relevance of having large amounts of data with ground truth for training, debugging and testing, roughly since 2008 we started to explore a different approach. In particular, the idea of using realistic virtual worlds (*e.g.* based on videogames) for train-

ing vision-based perception modules. The advantages were clear: (1) forcing the driving and data acquisition situations needed; (2) obtaining different types of pixel-wise ground truth (class ID, instance ID, depth, optical flow); (3) generating such data relatively fast (*e.g.* currently our SYNTHIA environment [50] can generate 10,000 images per hour with such ground truths, using standard consumer hardware); etc. Of course, such a proposal also came with doubts such as *can a visual model learned in virtual worlds operate well in real-world environments?*, and *does this depend on the degree of photo-realism?*. From our pioneering paper [36], where we used pedestrian detection based on HOG/Linear-SVM as proof-of-concept, to our last work, *i.e.* SYNTHIA [50], where we have addressed pixel-wise semantic segmentation via deep CNNs, we have been continuously exploring the idea of learning in virtual worlds to operate in real environments.

The use of synthetic data has attracted the attention of other researchers too, and more recently specially due to the massive adoption of deep CNNs to perform computer vision tasks and their data hungry nature. 3D CAD models have been used to train visual models for pose estimation, object detection and recognition, and indoor scene understanding [23, 1, 55, 59, 47, 57, 46, 56, 2, 10, 63, 44, 43, 45, 26, 28, 52, 37, 62, 61, 41, 6, 27]; a virtual racing circuit has been used for generating different types of pixel-wise ground truth (depth, optical flow and class ID) [25]; videogames have been used for train-

ing deep CNNs with the purpose of semantic segmentation [49] and depth estimation from RGB [58]; synthetic scenarios have been used also for evaluating the performance of different feature descriptors [29, 3, 74, 73, 75] and for training and testing optical and/or scene flow computation methods [39, 8, 42, 38], stereo algorithms [24], as well as trackers [64], even using synthetic clones of real-world areas of interest [18]; synthetic LIDAR-style data has been used for object detection too [31, 32]; finally, virtual worlds are being used for learning high-level artificial behavior such as playing Atari games [40], reproducing human behavior playing shooter games [35] and driving/navigating end-to-end [9, 82], even learning *unwritten* common sense [72, 83].

2 Need for Domain Adaptation

From the very beginning of our work, it was clear that there is a *domain gap* between virtual and real worlds. However, it was also clear that this was the case when using images coming from different (real) camera sensors and environments. In other words, the domain gap is not a virtual-to-real issue, but rather a more general sensor-to-sensor or environment-to-environment problem [70, 71]. Other researchers confirmed this fact too when addressing related but different visual tasks than ours [66]. Since then, training visual models in virtual worlds and applying *domain adaptation* techniques for their use in real-world scenarios come hand-by-hand for us. In fact, more authors have followed the approach of performing some explicit step of virtual- to real-world domain adaptation, without being an exhaustive list, the reader can address [32, 33, 63, 43] as illustrative examples.

We showed that virtual- to real-world domain adaptation is possible for holistic models based on the HOG+LPB/Linear-SVM paradigm [68] as well as on the Haar+EOH/AdaBoost one [69]. In the former case, proof-of-concept experiments adapting RGB-style synthetic images to far infrared ones (FIR) reported positive results too [60]. Moreover, for the *Deformable Part-based Model* (DPM) [17] we also proposed to use virtual worlds and domain adaptation [79, 77]. In most of the cases we focused on

supervised domain adaptation, *i.e.* a relatively few amount of annotated *target-domain* data (*i.e.* from the real world in our case) was used to adapt the model learned with *source-domain* data (from the virtual world). For the holistic models we focused on mixing the source and target data collected via *active learning* for model adaptation, we termed the corresponding feature space as *cool world*; while for DPM we focused on using just the source-domain model together with the target-domain data, *i.e.* without revisiting the source-domain data. In terms of modern deep CNNs, the former case would be similar to mixing source and target data in the mini-batches while the latter case is more in the spirit of the so-called fine-tuning.

In the rest of this chapter we are going to focus on DPM because it was the state-of-the-art for object detection before the breakthrough of deep CNNs. A priori it is a good proxy for deep CNNs regarding the specific experiments we want to address, after all deep CNNs eventually can require domain adaptation too [19, 67, 65, 11]. Obviously, being based on HOG-style features there is a point where much more data would not really translate to better accuracy [81], so we will keep training data in the order of a few thousands here. On the other hand, note that DPM can be reformulated as a deep CNN [22] for end-to-end learning. Moreover, the domain adaptation techniques we proposed for DPM [77], can be used as core technology for hierarchical domain adaptation¹ [78] as well as for weakly supervised incremental domain adaptation [80].

In particular, we are going to rely on our domain adaptation method for DPM termed as *Structure-aware Adaptive Structural SVM* (SA-SSVM), which gave us the best performance in [77]. In this chapter we compliment the experiments run in [77] mainly by addressing questions such as *the role of photo-realism in the virtual world*, as well as *how does the domain gap behave in virtual-vs-real data with respect to dominant object appearance per domain*. Moreover, for the sake of analyzing new use cases, instead of focusing on pedestrian detection using virtual data from

¹With this technique we won the 1st pedestrian detection challenge of the KITTI benchmark suite, a part of the *Recognition Meets Reconstruction Challenge* held in ICCV'13.

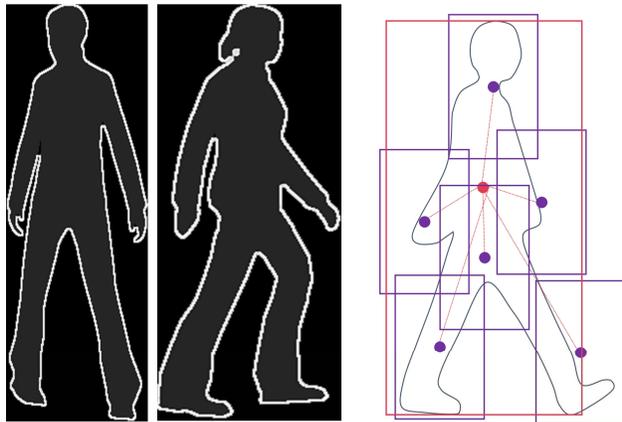


Figure 2: DPM for modeling pedestrians. There are two components (left, black background), and each component is encoded as a root and six parts (right, one component).

Half-Life 2 as in [77], here we focus on vehicle detection using different virtual-world datasets, namely Virtual KITTI [18], SYNTHIA [50], and GTA [49].

3 Domain Adaptation for DPM in a Nutshell

DPM encodes the *appearance* of objects' constituent *parts* together with a *holistic* object representation termed as *root*. In contrast to other part models, DPM allows the parts to be located at different positions with respect to the root. The plausible relative locations, known as *deformations*, are also encoded. Both appearance and deformations are learned. The appearance of the parts is learned at double the resolution than the root. The triplet root-parts-deformations is known as *component*. In order to avoid too blurred models DPM allows to learn a mixture of components. Different components use to correspond to very different object views or poses, specially when this implies very different aspect ratios of the corresponding root BB. See Fig. 2 for a pictorial intuition.

In practice, a DPM is encoded as a vector \mathbf{w} which

has to be learned. In the domain adaptation context, we term as \mathbf{w}^S the model learned with source-domain data (*e.g.* with virtual-world data). Our SA-SSVM domain adaptation method takes \mathbf{w}^S and relatively few annotated target-domain data (*e.g.* real-world data) to learn a new \mathbf{w} model which is expected to perform better in the target domain. The reader is referred to [77] for the mathematical technical details of how SA-SSVM works. However, let us explain the idea with the support of the example in Fig. 3; where \mathbf{w}^S consists of components: half body and full body, as well as persons seen from different viewpoints. Each component consists of root and parts (head, torso, etc). To adapt this DPM to a target domain, we decompose it as $\mathbf{w}^S = [\mathbf{w}_1^{S'}, \dots, \mathbf{w}_P^{S'}]'$, where P is the number of structures and u' stands for transpose of u . Note that each component, \mathbf{w}_p^S , may contain both appearance and deformation parameters (for roots only appearance). The decomposed model parameters are adapted to the target domain by different weights, denoted by $\beta_p, p \in \{1, P\}$; *i.e.* the SA-SSVM procedure allows domain adaptation for each of such structures separately by defining $\Delta \mathbf{w}_p = \mathbf{w}_p - \beta_p \mathbf{w}_p^S, p \in \{1, P\}$. In order to learn these adaptation weights, we further introduce a regularization term $\|\beta\|^2$ in the objective function, where $\beta = [\beta_1, \dots, \beta_P]'$, and we use a scalar parameter γ to control its relative penalty. Finally, C and the ξ_i are just the standard terms of a SVM objective function and N the number of target-domain samples used for the adaptation. After optimizing for the objective function (see mid box in Fig. 3), $\mathbf{w} = [\mathbf{w}_1', \dots, \mathbf{w}_P']'$ is the domain adapted DPM.

4 Experimental Results

4.1 Datasets

As we mentioned before, we are going to focus on vehicle detection for ADAS/AD applications. We use the training data of the KITTI car detection challenge [21]; which is split into two sets, one for actually training and the other for testing. Such a testing set will be the only one used here for that purpose, and we will follow the so-called *moderate*

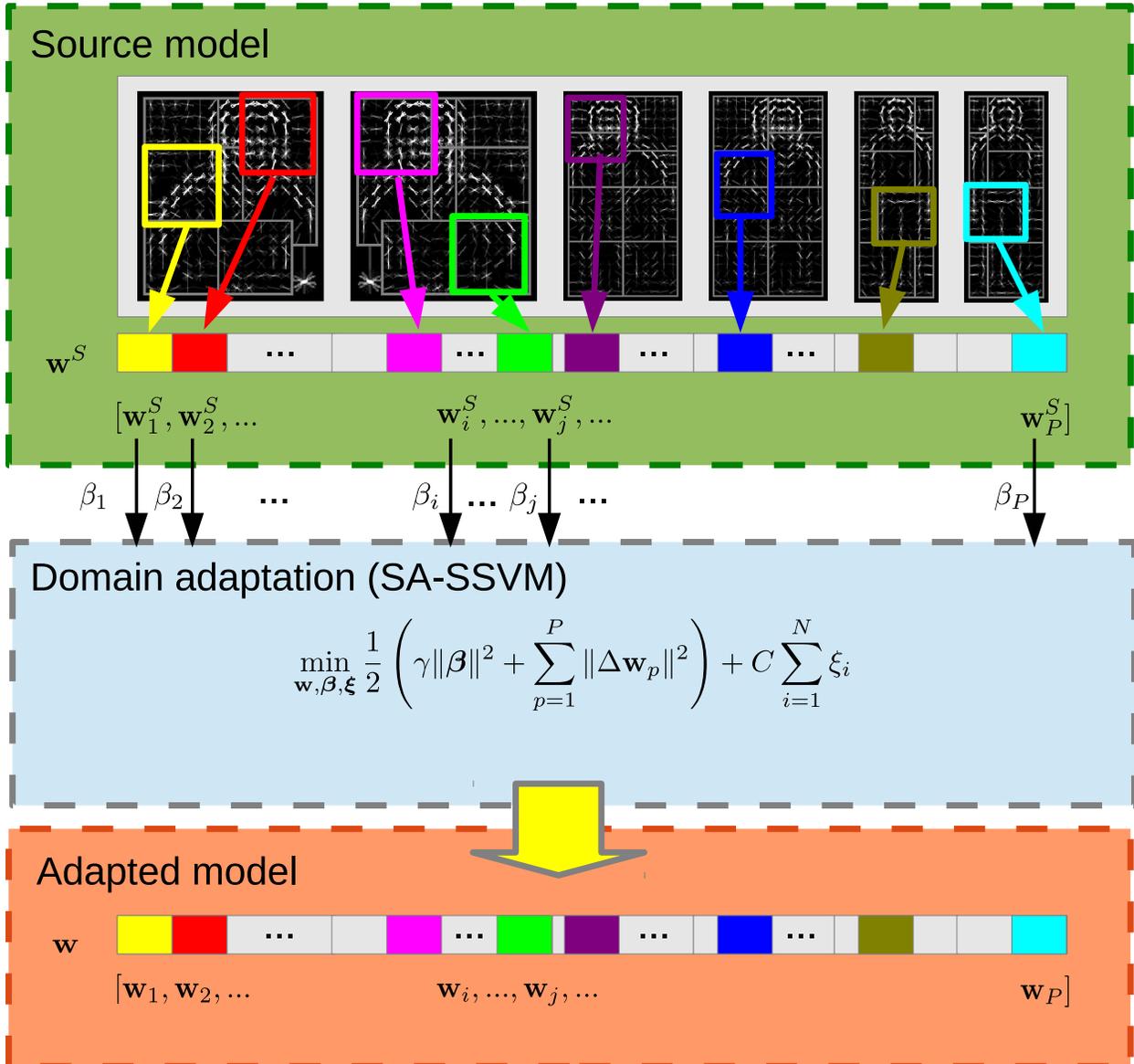


Figure 3: Domain Adaptation of DPM based on SA-SSVM (see main text for details).

setting when considering which vehicles are mandatory to detect. For training, we will consider four more datasets in addition to the mentioned split of the KITTI car detection challenge, thus, five in total. Namely, the KITTI car tracking dataset [21], its synthesized clone Virtual KITTI [18], SYNTHIA [50], and GTA [49]. Of course, SYNTHIA is a dataset with different types of ground truth, so we selected a set of cars and images for our experiments. In the case of GTA, we semi-automatically annotated with BBs a set of cars. Table 1 shows the number of samples in each dataset. Figures 4, 5, 6, and 7, show images sampled from KITTI-Det, KITTI Track with Virtual KITTI, SYNTHIA and GTA, respectively. Virtual KITTI and SYNTHIA are based on the same development framework, *i.e.* Unity3D². We can see that the images from GTA are more photo-realistic than the ones used in SYNTHIA and Virtual KITTI. SYNTHIA images are not always corresponding to a forward facing on-board virtual camera as is the case of Virtual KITTI and GTA. For more details about the datasets the reader can refer to the corresponding papers.

4.2 Protocol

In order to show the accuracy of the vehicle detectors we plot curves of *false positive per image* (FPPI) *vs miss rate* (MR) according to the Caltech protocol [15]; with an overlap of the 50% between detection and ground truth BBs. For training and testing we only consider *moderate* cases, which according to the definition given in the KITTI car detection challenge, are those vehicles non-occluded or just partially occluded (maximum truncation: 30%), and with a BB height ≥ 25 pixels. The vehicles mentioned in Tab. 1 refer to moderate cases.

Regarding DPM we use three components, each with eight parts. Part locations are initialized as 6×6 HOG-style cells (48×48 pixels) covering the root (at its double resolution version). Note that, in contrast to [79, 77], here we have not explored the use of the pixel-wise vehicle masks (available for virtual-world data) to provide a better initialization

of part locations during DPM learning. Thus, real- and virtual-world training data are used equally for learning source-domain DPMs.

Regarding the application of SA-SSVM we have followed the settings reported in [77] as producing the best results. Namely, the adapted structures correspond to the root and parts, *i.e.* not to components; and we set $\gamma = 0.08$ and $C = 0.001$ (see Fig. 3). Since domain adaptation experiments (*i.e.* SA-SSVM based ones) require random sampling of the target domain training data, they are run three times and the mean FPPI-MR curve and standard-deviation based intervals are plotted (*i.e.* as in [77] but with three repetitions instead of five).

4.3 Experiments

According to the datasets listed in Tab. 1, we define the set of source-domain datasets to be $\mathcal{S} = \{\text{KITTI-Track, Virtual KITTI, SYNTHIA, SYNTHIA-Sub, GTA}\}$. The target-domain dataset is always KITTI-Det Test. KITTI-Det Train and KITTI-Det Test are coming from the same domain since they correspond to two splits we have done from the same original dataset. All the learned detectors are tested in KITTI-Det Test, and the difference among them is the data used for their training. Accordingly, the reported experiments are as follows:

- *SRC*: Training with a dataset $s \in \mathcal{S}$.
- *TAR-ALL*: Training based on the full KITTI-Det Train.
- *TARX*: Training with a subset of random images from KITTI-Det Train, in particular, only using the 100X% of the images.
- *SA-SSVM*: Training with a dataset $s \in \mathcal{S}$ plus the images used for the *TARX* shown in the same plot.

Following this pattern, Fig. 8 shows results for $X = 0.1$ (*i.e.* 10%), Fig. 9 shows results for $X = 0.5$ (*i.e.* 50%), and Fig. 10 shows results for $X = 1$ (*i.e.* *ALL*).

²See unity3d.com

Table 1: Used samples for each dataset. *Images* stands for the number of images and, from them, *Vehicles* stands for the number of annotated vehicles using a bounding box. Negative samples are selected from background areas of the same images. *KITTI-Det Test* and *KITTI-Det Train* refer to two splits of the training set of the KITTI car detection training set. *KITTI-Det Test* is the testing set used in all the experiments of this chapter, while the rest of datasets are used only for training. For *KITTI Track* and *Virtual KITTI*, we use sequences 1, 2, 6, 18, and 20 as the training datasets. *SYNTHIA-sub* refers to a subset randomly sampled from *SYNTHIA*.

	KITTI-Det Test	KITTI-Det Train	KITTI-Track	Virtual KITTI	SYNTHIA	SYNTHIA-Sub	GTA
Images	3163	3164	2020	1880	1313	675	580
Vehicles	12894	12275	12950	6867	2052	1023	1054



Figure 4: Images sampled from KITTI-Det.



Figure 5: Images sampled from KITTI-Track (left) and Virtual KITTI (right). Note how Virtual KITTI is a synthesized but realistic clone of KITTI-Track.

4.4 Discussion

The first observation comparing SRC and TAR-ALL results (note that they are constant across figures since do not depend on X) is that there is a large domain gap. Since we would like to annotate as less real-world data as possible, let us start our analysis for $X = 0.1$ (see Fig. 8).

The worst case is when $\text{SRC} \in \{\text{KITTI-Track}, \text{Virtual KITTI}\}$ since the average miss rate is ~ 17 points worse than for TAR-ALL. The best case is when $\text{SRC} \in \{\text{SYNTHIA}, \text{GTA}\}$, where the gap is of ~ 12 points. Note that this is not related to the number of vehicle samples since GTA has $\sim 1/6$ of vehicles than Virtual KITTI for training, SYNTHIA $\sim 1/3$ than Virtual KITTI, and Virtual KITTI in turn contains $\sim 1/2$ of the vehicles in KITTI-Track and in KITTI-Det Test. An analogous behavior is seen when ranking the datasets by number of vehicle-free images. In any case, both ~ 17 and ~ 12 points are significant accuracy drops.

For going deeper in the analysis of what can be the reason for the domain gap, we can compare the results of KITTI-Track *vs* Virtual KITTI. We

see that they are basically equal. Since Virtual KITTI is a synthesized clone of KITTI-Track, we think that the main reason of the accuracy drop is not the virtual-to-real nature of the training images, but the typical vehicle poses and backgrounds reflected in the training datasets, *i.e.* when comparing Virtual KITTI/KITTI-Track with KITTI-Det Train. In other words, KITTI-Det Train represents better KITTI-Det Test since we built them from the same data set. Note, in addition, that KITTI-Track come from the same camera sensor as KITTI-Det Train and Test, which does not avoid the accuracy gap. Moreover, both SYNTHIA and GTA come from virtual worlds and still produce a detector that performs better than when using KITTI-Track.

We observe also that leaving out the $\sim 90\%$ of the images in KITTI-Det Train ($X = 0.1$) causes a drop in accuracy of ~ 6 points. In other words, in this case once we have ~ 316 manually annotated images ($\sim 10\%$ of KITTI-Det Train), annotating $\sim 2,848$ more is required to push the DPM to its limit, which is only ~ 6 points better³. Active learning or ad hoc

³It is a fallacy to believe that, because good datasets are



Figure 6: Images sampled from SYNTHIA dataset. Note that they are not always corresponding to a forward facing virtual camera on-board a car.

heuristics can be tried to alleviate such manual annotation effort. However, we observe that pre-training the DPM with automatically collected virtual-world data and using SA-SSVM for adapting the model, makes such ~ 316 images already very valuable, since in all cases the domain-adapted vehicle detector improves both the result of TAR0.1 and SRC (only virtual-world data). We can see that the best case is for SYNTHIA, which reduces the domain gap to ~ 2 points from ~ 12 points, and improves the result of TAR0.1 in ~ 4 points. An additional observation is that pre-training (SRC) the detectors with virtual-world data also allows to use active learning techniques as we did in [68] and/or ad hoc heuristics as we did in [80] for annotating more complementary images (*i.e.* other more informative ~ 316 ones) or

big, then big datasets are good [5].

collecting more but without human intervention (*i.e.* self-annotation). We have not done it here for the sake of simplicity, but it is reasonable to think that this would reduce the domain gap even more.

Figure 11 compares vehicle detections based only on the SYNTHIA samples we are using in this chapter, and the result of applying SA-SSVM to them with TAR0.1, in both cases setting the threshold of the model classifier to operate in the FPPI=1 regime. Note how SA-SSVM allows to obtain better results.

TAR0.5 ($X = 0.5$; see Fig. 9) and TAR-ALL basically show the same performance, so $\sim 1,582$ images have been annotated without a reward in DPM performance. Of course, although an annotation-training-test loop can be followed to avoid useless vehicle annotations, a priori it is difficult to know when to stop such manual annotations. On the other hand, even using TAR0.5 data, starting with a pre-

Invited book chapter to appear in *Domain Adaptation in Computer Vision Applications*, Springer Series: Advances in Computer Vision and Pattern Recognition, Edited by Gabriela Csurka. Written during Summer 2016.



Figure 7: Images sampled from the GTA videogame.

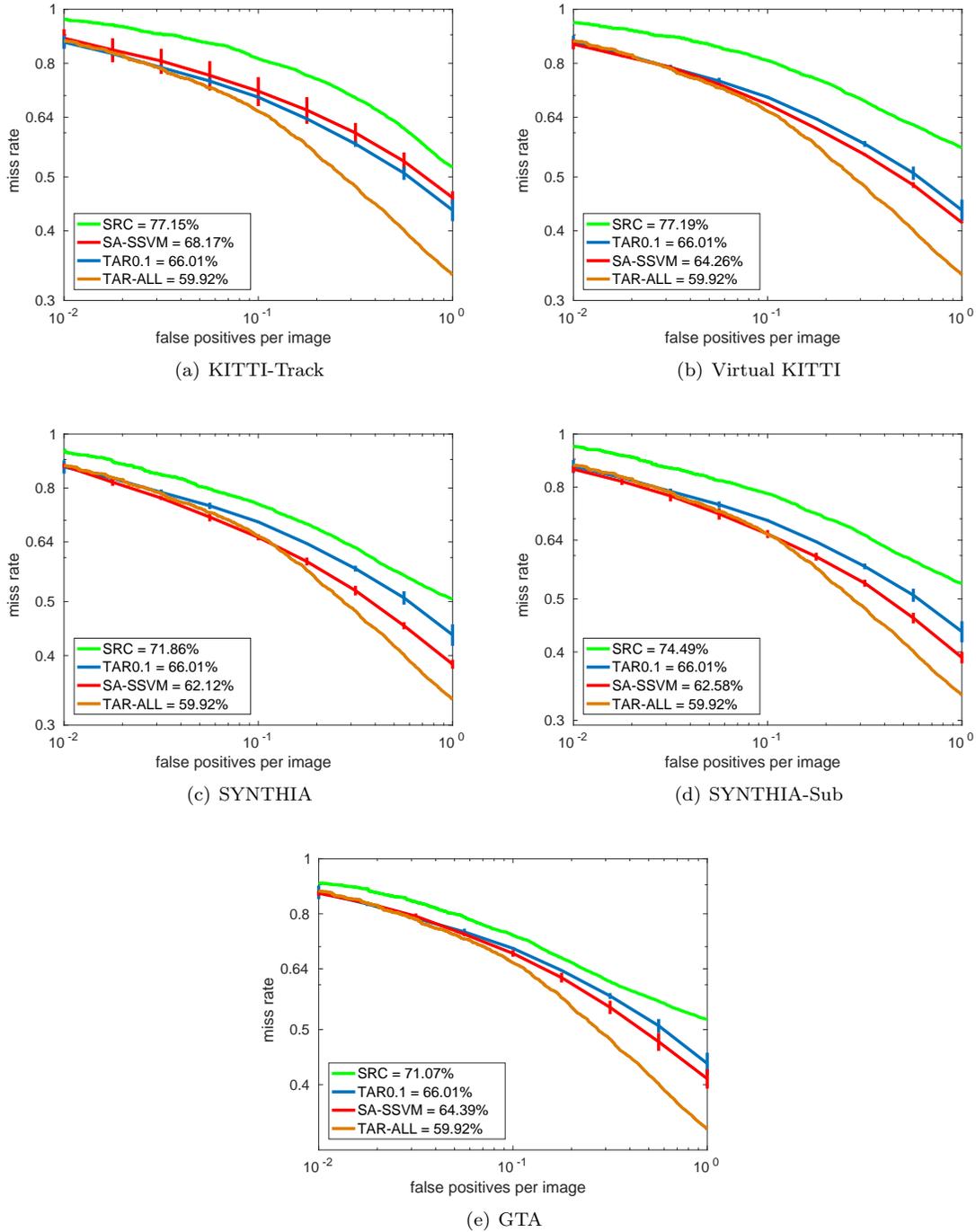


Figure 8: Results assuming $X = 0.1$ (see main text). In the box legend it is indicated the average miss rate for each experiment. Thus, the lower the better.

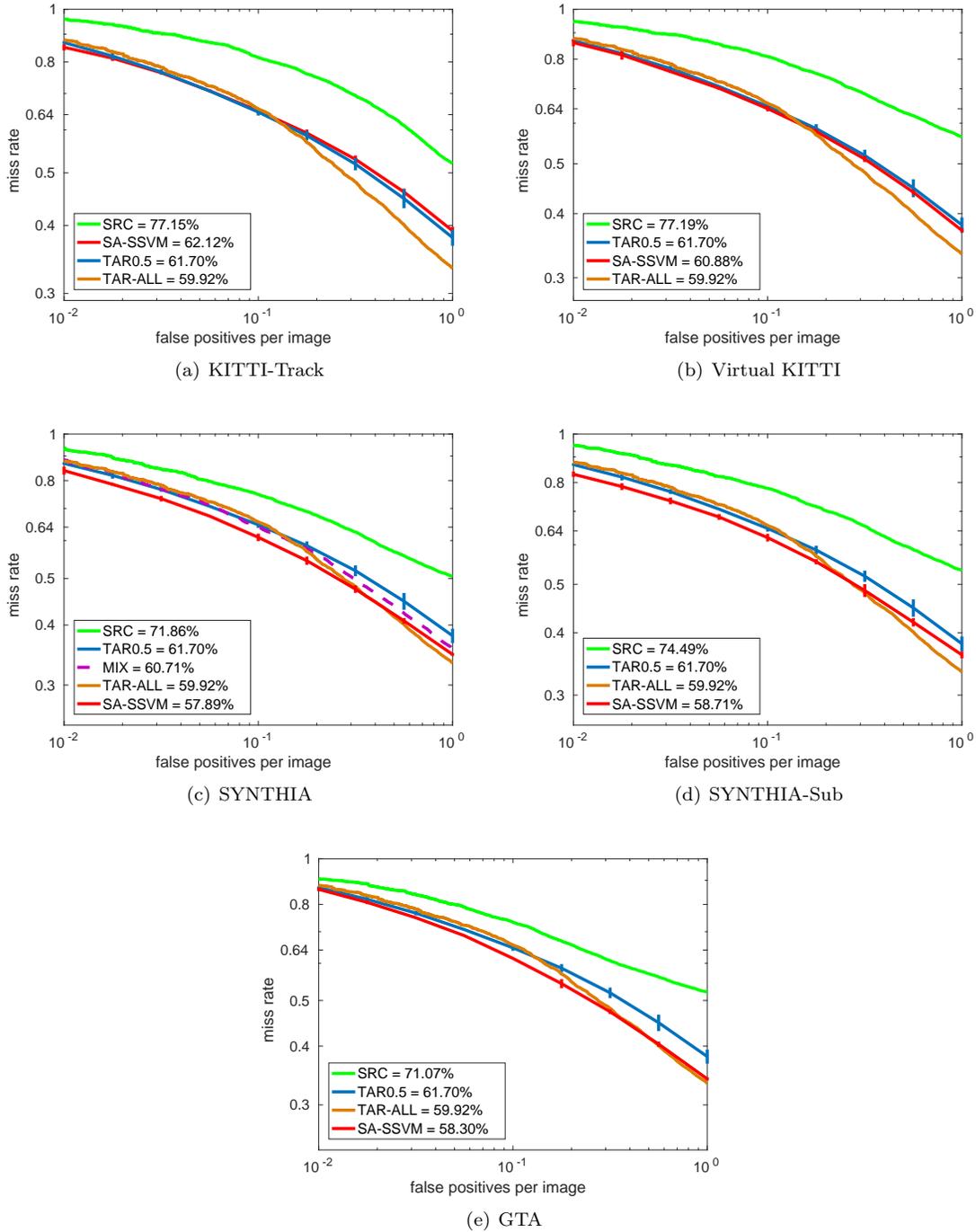


Figure 9: Results assuming $X = 0.5$ (see main text). In the box legend it is indicated the average miss rate for each experiment. Thus, the lower the better.

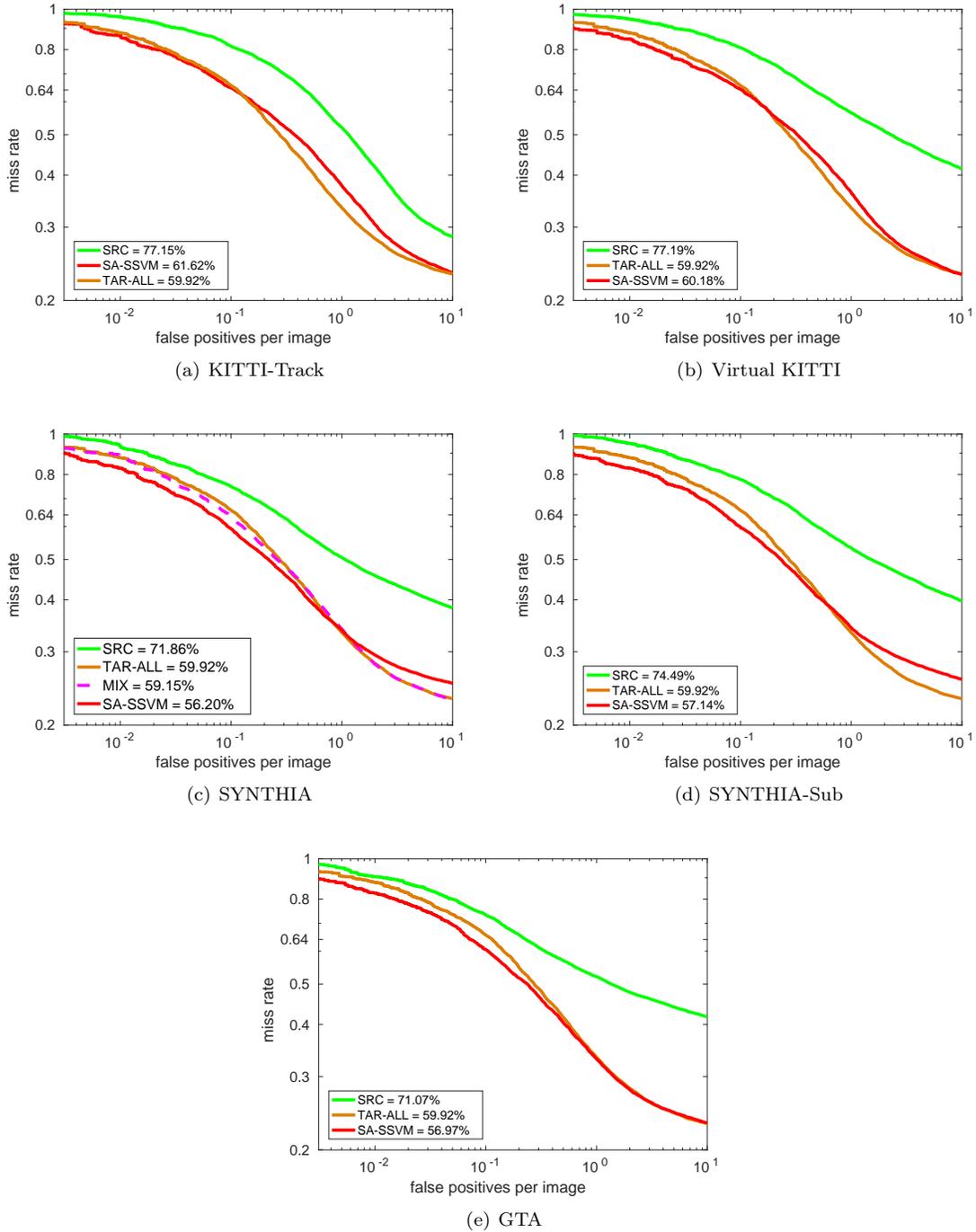


Figure 10: Results assuming $X = 1$ (ALL; see main text). In the box legend it is indicated the average miss rate for each experiment. Thus, the lower the better.



Figure 11: Vehicle detections when operating in the FPPI=1 regime. Left: DPM based on the SYNTHIA data considered in this chapter (SRC). Middle: Using the TAR0.1 version of KITTI-Det Train. Right: Adapting SRC by using TAR0.1 when applying SA-SSVM.

trained model (SRC) on either SYNTHIA or GTA allows to improve the performance of TAR-ALL, in the case of SYNTHIA by ~ 2 points with respect to TAR0.5. Looking at SYNTHIA-Sub and GTA, which has a similar number of samples (see Tab. 1), we can argue that GTA could probably reach the same performance than SYNTHIA if we would have the double of GTA vehicles. In any case, what it is remarkable is that it is more effective to have DPMs pre-trained in virtual worlds than just doubling the number of manually annotated target-domain images, *i.e.* at least assuming a manual annotation procedure free of prior knowledge about the current vehicle detector.

Even for $X = 1$ (see Fig. 10), *i.e.* combining the data used to train TAR-ALL and SRC, pre-training in virtual worlds is able to improve the performance of TAR-ALL alone. SYNTHIA provides us ~ 3 points of improvement with respect to TAR-ALL, being the overall best result. Using GTA as SRC eventually can provide such improvement too (again, by extrapolation of its performance when comparing to SYNTHIA-Sub).

In summary, the results presented and discussed so far reinforce the take home messages we highlighted in our previous works [36, 68, 77]; namely, combining models/data pre-trained in virtual worlds with a reasonable but low amount of real-world data through domain adaptation, is a really practical paradigm worth to explore for learning different kinds of models. As a matter of fact, according to the literature reviewed in sections 1 and 2, nowadays this approach is being widely adopted by the computer vision community.

Another interesting question that we did not addressed before refers to the degree of photo-realism, *i.e.* if a higher degree would imply to learn more accurate models and eventually not even requiring domain adaptation. This is a very important question since a extreme photo-realism may require hours for rendering a few images, while the degree of photo-realism of the virtual worlds presented here is achieved in real time using a standard modern gamer PC.

In our previous works we already saw that domain adaptation was required even when you train and test with real-world cameras. In other words, domain gap was due to sensor differences (no matter if one of the

sensors operates in real or virtual worlds) and the nature of the scenario where train and test images are acquired (typical illumination, background, and pose/view of dynamic objects). Because of this, our believe was that a more photo-realistic world would be just another sensor, still different from real-world, and therefore domain gaps would persists. Note that the experiments presented in this chapter reinforce this hypothesis: (1) using Virtual KITTI and KITTI-Track gives rise to SRC and domain-adapted detectors of similar performance in all the cases, *i.e.* despite the fact that KITTI-Track relies on the same real-world sensor than KITTI-Det Train and Test, while Virtual KITTI consists of synthesized data; (2) moreover, despite the fact that GTA contains images more photo-realistic than SYNTHIA, when using a similar number of samples (SYNTHIA-Sub) we see that the performance of the corresponding SRC and the domain-adapted detectors is basically the same.

Recent works [41, 75] reinforce the idea that, once a basic photo-realism is achieved (*i.e.* beyond Lambertian illumination and simplistic object materials), adding more and more photo-realism do not have a relevant impact. Thus, in our opinion from the evidences collected so far, Virtual KITTI and SYNTHIA are sufficiently photo-realistic for the tasks we are addressing (*i.e.* vision-based object detection and image semantic segmentation).

Another interesting point of analysis is if it is better to just mixing virtual- and real-world data or fine-tuning a pre-trained model on virtual-world data with real-world samples. The former is what we called *cool world* [70, 68], while SA-SSVM is an example of the later. Because of that we have run a *MIX* experiment with SYNTHIA and TAR-ALL, which can be seen in Fig. 10(c). In this case, we have just mixed the data and run an standard DPM learning procedure. Note that the result is ~ 3 points worse than using SA-SSVM. Moreover, the training time of MIX is much longer than the one of SA-SSVM, since it uses samples from both domains and training from scratch also requires more iterations to converge. If we extrapolate these experiments to the deep CNNs paradigm, a priori we would think than fine-tuning is the proper approach. However, when working in [50, 51], *i.e.* in semantic segmentation based on deep

CNNs, using the appropriate mini-batch scheme to weight the relevance of the samples as a function to their domain (virtual or real), we obtained better performance than with fine-tuning. Therefore, regarding this topic, we have no clear conclusions yet. Of course, the advantage of fine-tuning would be avoiding to revisit the source data; thus, this is a point to keep researching.

Overall, our research and the research presented so far by the computer vision community, led us to insist in the adoption of the paradigm where virtual worlds and domain adaptation techniques are used to train the desired models. Moreover, we think that the degree of photo-realism like the presented already in datasets such as Virtual KITTI and SYNTHIA is sufficient for this task. In addition, although in this chapter we have focused on DPM-based vehicle detection, we think the conclusions can be extrapolated to other computer vision tasks where the visual appearance is important (*e.g.* object detection in general and semantic segmentation). Of course, it is worth to note that at this moment the best results on the KITTI car detection challenge are dominated by approaches based on deep CNNs, providing astonishing high performances in the moderate setting, far beyond DPM approaches. Such benchmark seems to be challenging enough for DPM, but still is a small proportion of the real-world and this will be the real challenge for deep CNNs. Therefore, we also think that our conclusions will be extrapolated from DPM to other powerful models such as deep CNNs when addressing more challenging scenarios; note that in Sect. 2 we have mentioned already that even deep CNNs require domain adaptation. On the other hand, what is expected is that deep CNNs would require less domain adaptation than DPM since they are models with more capacity to generalize across domains.

5 Conclusion

In this chapter we have shown how virtual worlds are effective for training visual models when combined with domain adaptation techniques. Although we have focused on DPM and vehicle detection as

proof-of-concept, we believe that the conclusions extrapolate to other visual tasks based on more complex models such as deep CNNs. We have presented results which suggest that extreme photo-realism is not necessary, *i.e.* the degree of photo-realism already achieved in datasets such as Virtual KITTI and SYNTHIA is sufficient, provided domain adaptation would be necessary even when relying on more photo-realistic datasets (here GTA).

Looking into the future, we think a best practice would be to design sets of relatively controlled virtual-world scenarios, designed to train, debug and test visual perception and other AI capabilities (Virtual KITTI and SYNTHIA are examples of this). In other words, with the knowledge accumulated so far, we do not bet for building a gigantic virtual world to try to avoid domain gap issues. This would be really difficult to build and handle. We prefer to pursue domain adaptation to save any existing virtual-to-real world gap. However, we think the research must go into the direction of unsupervised domain adaptation for allowing the systems trained in virtual worlds to self-adapt to real-world scenarios. An example in this line is the approach we presented in [80], where manual annotations were not required to train a domain adapted pedestrian detector for an on-board moving camera setting. However, this approach performs the adaptation off-line, which can be perfectly right for many applications (*e.g.* adapting pre-trained surveillance systems to different places), but the real challenge is to do it on-line.

Acknowledgments Authors want to thank the next funding bodies: the Spanish MEC Project TRA2014-57088-C2-1-R, the People Programme (Marie Curie Actions) FP7/2007-2013 REA grant agreement no. 600388, and by the Agency of Competitiveness for Companies of the Government of Catalonia, ACCIO, the Generalitat de Catalunya Project 2014-SGR-1506 and the NVIDIA Corporation for the generous support in the form of different GPU hardware units.

References

- [1] Ankur Agarwal and Bill Triggs. A local basis representation for estimating human pose from

- cluttered images. In *Asian Conference on Computer Vision (ACCV)*, 2006.
- [2] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3D chairs: exemplar part-based 2d-3d alignment using a large dataset of CAD models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [3] Mathieu Aubry and Bryan C. Russell. Understanding deep features with computer-generated imagery. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 35(8):1798–1828, 2013.
- [5] Tamara L. Berg, Alexander Sorokin, Gang Wang, David A. Forsyth, Derek Hoiem, Ian Endres, and Ali Farhadi. It’s all about the data. *Proceedings of the IEEE*, 98(8):1434–1452, 2010.
- [6] Erik Bochinski, Volker Eiselein, and Thomas Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. In *Advanced Video and Signal-based Surveillance (AVSS)*, 2016.
- [7] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(20):88–89, 2009.
- [8] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, 2012.
- [9] Chenyi Chen, Ari Seff, Alain L. Kornhauser, and Jianxiong Xiao. DeepDriving: Learning affordance for direct perception in autonomous driving. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [10] Liang-Chieh Chen, Sanja Fidler, and Raquel Yuille, Alan L. Urtasun. Beat the MTurkers: Automatic image labeling from weak 3D supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [11] Brian Chu, Vashisht Madhavan, Oscar Beijbom, Judy Hoffman, and Trevor Darrell. Best practices for fine-tuning visual classifiers to new domains. In *European Conference on Computer Vision (ECCV), Transferring and Adapting Source Knowledge in Computer Vision (TASK-CV)*, 2016.
- [12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [13] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893, 2005.
- [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [15] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: an evaluation of the state of the art. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 34(4):743–761, 2012.
- [16] Markus Enzweiler and Dariu M. Gavrilă. Monocular pedestrian detection: Survey and experiments. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 31(12):2179–2195, 2009.
- [17] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based

- models. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 32(9):1627–1645, 2010.
- [18] Adrien Gaidon, Qiao Wang, Yann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [19] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.
- [20] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The KITTI dataset. *International Journal of Robotics Research (IJRR)*, 32(11):1231–1237, 2016.
- [21] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [22] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 437–446, 2015.
- [23] Kristen Grauman, Gregory Shakhnarovich, and Trevor Darrell. Inferring 3D structure with a statistical image-based shape model. In *IEEE International Conference on Computer Vision (ICCV)*, 2003.
- [24] Ralf Haeusler and Daniel Kondermann. Synthesizing real world stereo challenges. In *German Conference on Pattern Recognition (GCPR)*, 2013.
- [25] Haltakov Haltakov, Christian Unger, and Slobodan Ilic. Framework for generation of synthetic ground truth data for driver assistance applications. In *German Conference on Pattern Recognition (GCPR)*, 2013.
- [26] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. SynthCam3D: Semantic understanding with synthetic indoor scenes. *CoRR*, arXiv:1505.00171, 2015.
- [27] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Understanding real world indoor scenes with synthetic data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [28] Hironori Hattori, Vishnu Naresh Boddeti, Kris M. Kitani, and Takeo Kanade. Learning scene-specific pedestrian detectors without real data. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [29] Biliana Kaneva, Antonio Torralba, and William T. Freeman. Evaluation of image features using a photorealistic virtual world. In *IEEE International Conference on Computer Vision (ICCV)*, 2011.
- [30] Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton. ImageNet classification with deep Convolutional Neural Networks. In *Annual Conference on Neural Information Processing Systems (NIPS)*, 2012.
- [31] Kevin Lai and Dieter Fox. 3D laser scan classification using web data and domain adaptation. In *Robotics: Science and Systems*, 2009.
- [32] Kevin Lai and Dieter Fox. Object recognition in 3D point clouds using web data and domain adaptation. *International Journal of Robotics Research (IJRR)*, 29(8):1019–1037, 2010.
- [33] Wenbin Li and Mario Fritz. Recognizing materials from virtual examples. In *European Conference on Computer Vision (ECCV)*, 2012.
- [34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft

- COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755, 2014.
- [35] Joan M. Llargues, Juan Peralta, Raul Arrabales, Manuel González, Paulo Cortez, and Antonio M. López. Artificial intelligence approaches for the generation and assessment of believable human-like behaviour in virtual characters. *Expert Systems With Applications*, 41(16):7281–7290, 2014.
- [36] Javier Marín, David Vázquez, David Gerónimo, and Antonio M. López. Learning appearance in virtual scenarios for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [37] Francisco Massa, Bryan C. Russell, and Mathieu Aubry. Deep exemplar 2D-3D detection by adapting from real to rendered views. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [38] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [39] Stephan Meister and Daniel Kondermann. Real versus realistically rendered scenes for optical flow evaluation. In *Conference on Electronic Media Technology (CEMT)*, 2011.
- [40] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari with deep reinforcement learning. In *Annual Conference on Neural Information Processing Systems (NIPS), Workshop on Deep Learning*, 2013.
- [41] Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. How useful is photo-realistic rendering for visual learning? *CoRR*, arXiv:1603.08152, 2016.
- [42] Naveen Onkarappa and Angel D. Sappa. Synthetic sequences and ground-truth flow field generation for algorithm validation. *Multimedia Tools and Applications*, 74(9):3121–3135, 2015.
- [43] Pau Panareda-Busto, Joerg Liebelt, and Juergen Gall. Adaptation of synthetic data for coarse-to-fine viewpoint refinement. In *BMVA British Machine Vision Conference (BMVC)*, 2015.
- [44] Jeremie Papon and Markus Schoeler. Semantic pose using deep networks trained on synthetic RGB-D. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [45] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3D models. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [46] Bojan Pepik, Michael Stark, Peter Gehler, and Bernt Schiele. Teaching 3D geometry to deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [47] Leonid Pishchulin, Arjun Jain, Christian Wojek, Mykhaylo Andriluka, Thorsten Thormählen, and Bernt Schiele. Learning people detection models from few training samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [48] Amazon Mechanical Turk. <http://www.mturk.com>.
- [49] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Koltun Vladlen. Playing for data: Ground truth from computer games. In *European Conference on Computer Vision (ECCV)*, 2016.
- [50] German Ros, Laura Sellart, Joanna Materzyska, David Vázquez, and Antonio M. López. The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

- [51] German Ros, Simon Stent, Pablo F. Alcantarilla, and Tomoki Watanabe. Training constrained deconvolutional networks for road scene semantic segmentation. *CoRR*, arXiv:1603.08152, 2016.
- [52] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137:24–37, 2015.
- [53] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision (IJCV)*, 77(1–3):157–173, 2008.
- [54] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International Journal of Computer Vision (IJCV)*, 105(3):222–245, 2013.
- [55] Scott Satkin, Michael Goesele, and Bernt Schiele. Back to the future: Learning shape models from 3D CAD data. In *BMVA British Machine Vision Conference (BMVC)*, 2010.
- [56] Scott Satkin, Jason Lin, and Martial Hebert. Data-driven scene understanding from 3D models. In *BMVA British Machine Vision Conference (BMVC)*, 2012.
- [57] Johannes Schels, Jörg Liebelt, Klaus Schertler, and Rainer Lienhart. Synthetically trained multi-view object class and viewpoint detection for advanced image retrieval. In *International Conference on Multimedia Retrieval (ICMR)*, 2011.
- [58] Alireza Shafaei, James J. Little, and Mark Schmidt. Play and learn: Using video games to train computer vision models. In *BMVA British Machine Vision Conference (BMVC)*, 2016.
- [59] Jamie Shotton, Andrew Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipmanand, and Andrew Blake. Real-time human pose recognition in parts from a single depth image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [60] Yainuvis Socarras, Sebastian Ramos, David Vázquez, Antonio M. López, and Theo Gevers. Adapting pedestrian detection from synthetic to far infrared images. In *IEEE International Conference on Computer Vision (ICCV), Visual Domain Adaptation and Dataset Bias (ICCV-VisDA)*, 2013.
- [61] Hao Su, Charles R. Qi, Yangyan Yi, and Leonidas Guibas. Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [62] Hao Su, Fan Wang, Yangyan Yi, and Leonidas Guibas. 3D-assisted feature synthesis for novel views of an object. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [63] Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *BMVA British Machine Vision Conference (BMVC)*, 2014.
- [64] Geoffrey R. Taylor, Andrew J. Chosak, and Paul C. Brewer. OVVV: Using virtual worlds to design and evaluate surveillance systems. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.
- [65] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. A deeper look at dataset bias. In *German Conference on Pattern Recognition (GCPR)*, 2015.
- [66] Antonio Torralba and Alexei A. Efros. Unbiased look at dataset bias. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [67] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4068–4076, 2015.

- [68] David Vázquez, Antonio M. López, Javier Marín, Daniel Ponsa, and David Gerónimo. Virtual and real world adaptation for pedestrian detection. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 36(4):797 – 809, 2014.
- [69] David Vázquez, Antonio M. López, Daniel Ponsa, and David Gerónimo. Interactive training of human detectors. In Angel D. Sappa and Jordi Vitrià, editors, *Multimodal Interaction in Image and Video Applications Intelligent Systems*, pages 169–184. Springer, 2013.
- [70] David Vázquez, Antonio M. López, Daniel Ponsa, and Javier Marín. Cool world: domain adaptation of virtual and real worlds for human detection using active learning. In *Annual Conference on Neural Information Processing Systems (NIPS), Workshop on Domain Adaptation: Theory and Applications*, 2011.
- [71] David Vázquez, Antonio M. López, Daniel Ponsa, and Javier Marín. Virtual worlds and active learning for human detection. In *International Conference on Multimodal Interaction (ICMI)*, 2011.
- [72] Ramakrishna Vedantam, Xiao Lin, Tanmay Batra, C. Lawrence Zitnick, and Devi Parikh. Learning common sense through visual abstraction. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.
- [73] V.S.R. Veeravasrapu, Rudra Narayan Hota, Constantin Rothkopf, and Ramesh Visvanathan. Model validation for vision systems via graphics simulation. *CoRR*, arXiv:1512.01401, 2015.
- [74] V.S.R. Veeravasrapu, Rudra Narayan Hota, Constantin Rothkopf, and Ramesh Visvanathan. Simulations for validation of vision systems. *CoRR*, arXiv:1512.01030, 2015.
- [75] V.S.R. Veeravasrapu, Constantin Rothkopf, and Ramesh Visvanathan. Model-driven simulations for deep convolutional neural networks. *CoRR*, arXiv:1605.09582, 2016.
- [76] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.
- [77] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M. López. Domain adaptation of deformable part-based models. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 36(12):2367–2380, 2014.
- [78] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M. López. Hierarchical adaptive structural SVM for domain adaptation. *International Journal of Computer Vision (IJCV)*, 119(2):159–178, 2016.
- [79] Jiaolong Xu, David Vázquez, Antonio M. López, Javier Marín, and Daniel Ponsa. Learning a part-based pedestrian detector in a virtual world. *Transactions on Intelligent Transportation Systems (ITS)*, 15(5):2121–2131, 2014.
- [80] Jiaolong Xu, David Vázquez, Krystian Mikolajczyk, and Antonio M. López. Hierarchical online domain adaptation of deformable part-based models. In *International Conference on Robotics and Automation (ICRA)*, 2016.
- [81] Xiangxin Zhu, Carl Vondrick, Charless C. Fowlkes, and Deva Ramanan. Do we need more training data? *International Journal of Computer Vision (IJCV)*, 119(1):76–92, 2016.
- [82] Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J. Lim, and Abhinav Gupta. Target-driven visual navigation in indoor scenes using deep reinforcement learning. *CoRR*, arXiv:1609.05143, 2016.
- [83] C. Lawrence Zitnick, Ramakrishna Vedantam, and Devi Parikh. Adopting abstract images for semantic scene understanding. *Transactions of Pattern Recognition and Machine Analyses (PAMI)*, 38(4):627–638, 2016.