Pedestrian Detection: Exploring Virtual Worlds

Javier Marín
Computer Vision Center,
Universitat Autònoma de Barcelona, Spain

David Gerónimo, David Vázquez, Antonio M. López
Computer Vision Center and Computer Science Department,
Universitat Autònoma de Barcelona, Spain

1 Introduction

The objective of *advanced driver assistance systems* (ADAS) is to improve traffic safety by assisting the driver through warnings and by even automatically taking active countermeasures. Two examples of successfully commercialised ADAS are lane departure warnings and adaptive cruise control, which make use of either active (*e.g.*, radar) or passive (*e.g.*, cameras) sensors to keep the vehicle on the lane and maintain a safe distance from the preceding vehicle, respectively. One of the most complex safety systems are *pedestrian protection systems* (PPSs) (Bishop, 2005; Gandhi & Trivedi, 2007; Enzweiler & Gavrila, 2009; Gerónimo et al., 2010), which are specialised in avoiding vehicle-to-pedestrian collisions. In fact, this kind of accidents results in approximately 150000 injuries and 7000 killed pedestrians every year just in the European Union (UN-ECE, 2007). Similar statistics apply to the United States, while underdeveloped countries are increasing theirs year after year. In the case of PPSs, the most promising approaches make use of images as main source of information, as can be seen in the large amount of proposals exploiting them (Gerónimo et al., 2010). Hence, the core of a PPS is a forward facing camera that acquires images and processes them using Computer Vision techniques. In fact, the Computer Vision community has traditionally maintained a special interest in detecting humans, given the challenging topic it represents. Pedestrians are one of the most complex object to analyse: their variability in pose and clothes, distance to the camera, backgrounds, illuminations, occlusions make their detection a difficult task. In addition, the images are acquired from a mobile platform, so traditional human detection algorithms such as background subtraction are not applicable in a straightforward manner. Finally, the task has to be carried out in real-time.

State-of-the-art detectors rely on machine learning algorithms trained with labelled samples, *i.e.*, *examples* (pedestrians), and *counterexamples* (background). Therefore, in order to build robust pedestrian detectors the quality of the training data is fundamental. Last years various authors have publicly released their pedestrian datasets (Dalal & Triggs, 2005; Dollár et al., 2009; Enzweiler & Gavrila, 2009; Wojek et al., 2009; Gerónimo et al., 2010) which have gradually become more challenging (bigger number of samples, new scenarios, occlusions, etc.). In the last decade, the traditional research process has been to present a new database containing images from the world, and then researchers developed new and improved detectors (Dollár et al., 2011; Enzweiler & Gavrila, 2009; Tuzel et al., 2008; Lin & Davis, 2008). In this chapter, we explore the possibilities

that a virtual *computer generated* database, free of real-world images, can offer to this process.

The use of Computer Graphics in Computer Vision has been pursued during the last decade. Grauman *et al.* (Grauman et al., 2003) exploit computer generated images to train a probabilistic model to infer multi-view pose estimation. Such generated images are obtained by using a software tool. More specifically, the shape model information is captured through different multiple cameras obtaining the contour of the silhouettes simultaneously, and the structure information is formed by a fixed number of 3D body part locations. Later, shape and structure features are used together to construct a prior density using a mixture of probabilistic PCAs. Finally, given a set of silhouettes a new shape's reconstruction is obtained to infer the structure. Broggi *et al.* (Broggi et al., 2005) use synthesised examples for pedestrian detection in infrared images. More specifically, a 3D pedestrian model is captured from different poses and viewpoints in which the background is later modelled. Finally, instead of following a learning-by-examples approach to obtain a single classifier, a set of templates is used by a posterior pedestrian detection process based on template matching. Enzweiler *et al.* (Enzweiler & Gavrila, 2008) enlarge a set of examples by transforming the shape of pedestrians (labelled in real images) as well as the texture of pedestrians and background. The pedestrian classifier is learnt by using a discriminative approach (NNs with LRFs and Haar features with SVM are tested). Since these transformations encode a generative model, the overall approach is seen as a generative-discriminative learning paradigm. The generative-discriminative cycle is iterated several times in a way that new synthesised examples are added in each iteration by following a probabilistic selective sampling to avoid redundancy in the training set. These examples are later used to train a model to be used in a detector. Marín *et al.* (Marín et al., 2010) use a commercial game engine to create a virtual pedestrian dataset to train a synthetic model to test in real images. Then, they show the comparison between real and virtual models revealing the similarity of both detectors in terms of HOG features and SVM classifier.

More recently, Pishchulin *et al.* (Pishchulin et al., 2011a) employ a rendering-based reshaping method to generate a synthetic training set using real subjects (similar to (Enzweiler & Gavrila, 2008)) from only few persons and views. In this case, they collected a dataset of eleven subjects each represented in six different poses corresponding to a walking cycle. Eight different viewpoints are then used to capture each pose. Later, they gradually change the height of each pose (15 higher/15 smaller) to obtain 20400 positive examples in total. Finally, they explore how the number of subjects used during the training process affects the performance as well as the combination between real and synthesised models. The same authors (Pishchulin et al., 2011b) also explore the use of synthetic data obtained from a 3D human shape model in order to complement the image-based data. Such 3D human shape and pose model is obtained through a database of 3D laser scans of humans which describes shape and pose variations. Similar to (Pishchulin et al., 2011a) best performance is obtained when training models in different datasets and then combining them.

The reviewed proposals can be divided into the using synthesised examples coming from real data and the ones using only virtual examples. While promising detectors based on synthesised examples are still needed of real images in which models are obtained through sophisticated systems (Pishchulin et al., 2011a; Pishchulin et al., 2011b), the virtual worlds generated using just synthetic data offer a large number of available models and possibilities without using real data. In this chapter we focus on learning pedestrian models in such virtual worlds to be used in real world detection and how different settings perform when testing in real data. In particular, following the approach in (Marín et al., 2010) we learn such appearance using virtual samples in order to detect pedestrians in real images (Fig. 1). Besides, we extend the approach published with specific analysis on the required number of virtual models and training examples to get a satisfactory performance, and present new results on how the pose influences the performance.

The process is as follows. We record training sequences in realistic virtual cities and train appearance-based pedestrian classifiers using HOG and linear SVM, a baseline method for building such classifiers that remains competitive for pedestrian detection in the ADAS context (Dollár et al., 2011; Enzweiler & Gavrila, 2009). As Marín *et al.* we test such classifiers in the same publicly available dataset, Daimler AG (Enzweiler & Gavrila, 2009).

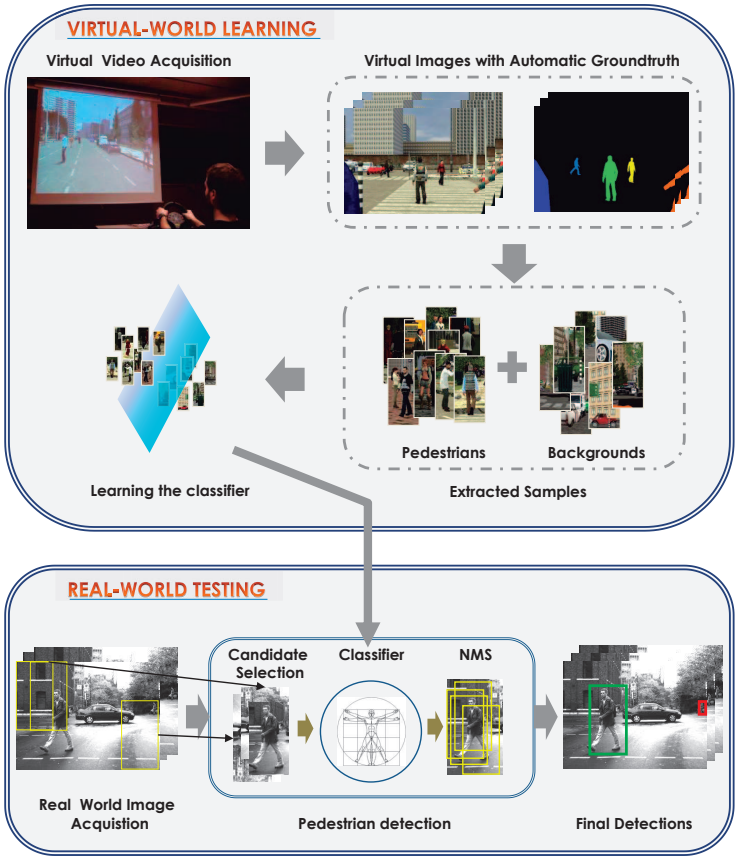


Figure 1: General Idea. This figure shows the pipeline followed in our experiments.

The obtained results are evaluated in a per-image basis and compared with the classifier obtained when using real samples for training. In this work, we specially focus on exploring the impact in the performance w.r.t the different number of virtual models, the total number of examples and the pose distribution.

The structure of the chapter is as follows. Section 2 introduces the datasets used for training (real world and virtual world ones) and testing (only real world images). Section 3 details the conducted experiments, which use the real- and virtual-pedestrian models in a complete detection system. Section 4 presents the results and corresponding discussions. Finally, section 5 summarizes the conclusions and future work.

2 Datasets

The lack of publicly available large datasets for pedestrian detection in the ADAS context has been a recurrent problem for years (Dollár et al., 2011; Enzweiler & Gavrila, 2009; Gerónimo et al., 2010). For instance, INRIA dataset (Dalal & Triggs, 2005) has been the most widely used for pedestrian detection. However, it contains photographic pictures in which people are mainly close to the camera and in focus. Moreover, there are backgrounds that do not correspond to urban scenarios, which are the most interesting and difficult ones for detecting

pedestrians from a vehicle.

Fortunately, three more adapted datasets for the ADAS context have recently been made publicly available. They have been presented by Caltech (Dollár et al., 2009), Daimler (Enzweiler & Gavrila, 2009), and the Computer Vision Center (Gerónimo et al., 2010). In the current work, we perform the experiments in Daimler dataset since it comes from one of the most relevant automotive companies worldwide, thus, we can expect the images to be quite representative for ADAS. Our proposed virtual dataset is also focused on ADAS images, but acquired from a virtual car in a computer graphics generated world. In the next sections we summarise the details of both datasets.

2.1 Real images

We summarize the main characteristics of Daimler's dataset. In fact, it consists of a training set and different testing sets.

2.1.1 Training set

The images of this set are grayscale and were acquired at different times of day and locations (Fig. 3).

Examples. The original training frames with pedestrians are not publicly available, but cropped pedestrians are. From 3915 manually labelled pedestrians, 15660 were obtained by applying small vertical and horizontal random shifts (*i.e.*, jittering) and mirroring, and then put publicly available. The size of each cropped example is 48×96 pixels, which comes from the 24×72 pixels of the contained pedestrian plus an additional margin of 12 pixels per side. All the original labelled pedestrians are at least 72 pixels high, thus, some of the samples come from downscaling but none from upscaling. All the samples contain pedestrians that are upright and not occluded.

Counterexamples. 6744 pedestrian-free frames were delivered. Their resolution is 640×480 pixels. Thus, to gather cropped counterexamples these frames must be sampled. Conceptually, the sampling process we use can be described as follows. We need counterexamples of the same dimensions than the cropped pedestrian examples, *i.e.*, 48×96 pixels. Therefore, we can select windows of size $48k^i \times 96k^i$ pixels, where k is the scale step (1.2 in our case) and $i \in \{0,1,2,...\}$, provided that they are fully contained in the image we are sampling. Then, we can downscale the counterexamples by a factor k^i using, for instance, bi-cubic interpolation. In practice, we implement this sampling idea by using a pyramid of the frame to be sampled and then by cropping windows of size 48×96 pixels at each layer (Dalal, 2006), which is closely related to the scanning strategy used by the final pedestrian detector (Sect. 3).

2.1.2 Testing set

The testing set consists in a sequence of 21790 grayscale frames of 640×480 pixels. The sequence was acquired on-board while driving during 27 minutes through urban scenarios. Moreover, this testing set does not overlap the training set. The testing set includes 56492 manually labelled pedestrians. The labels contain also an additional information indicating whether they are of *mandatory* detection or not. Basically, the pedestrians labelled as non-mandatory are those either occluded, not upright, or smaller than 72 pixels high. There are 2459 mandatory pedestrians in total. Frames of the training set can be seen with overlayed results in Sect. 4 (Fig. 11).

2.2 Virtual images

In order to obtain virtual images, the first step consists in building virtual scenarios. We have carried it out by using the video game Half-Life 2 (Valve Software Corporation, 2004). This game allows to run maps created

(a) (b)

Figure 2: Virtual image with corresponding automatically generated pixel-wise groundtruth for pedestrians. (a) represents the original image, and (b) the pixel-wise groundtruth image, where every pedestrian has a different color label.

with an editor named *Hammer* (included in the Valve's software package), as well as to add modifications (*a.k.a. mods*). We use Hammer to create realistic virtual cities with roads, streets, buildings, traffic signs, vehicles, pedestrians, different illumination conditions, etc. Once we start to *play*, the pedestrians and vehicles move through the virtual city by respecting physical laws (*e.g.*, pedestrians do not float and cannot be at the same place than other solid objects at the same moment) as well as by following their artificial intelligence (*e.g.*, vehicles move on the road).

In order to acquire images in virtual scenarios we use the *mod* created by the company ObjectVideo. Taylor *et al.* (Taylor et al., 2007) show the usefulness of such a *mod* for designing and validating people tracking algorithms for video surveillance (static camera). A relevant functionality consists in providing pixel-wise groundtruth for human targets (Fig. 2). However, since the aim in (Taylor et al., 2007) is to test algorithms under controlled conditions, all the work is done with virtual scenarios without considering real world images. Thus, the work we present in this chapter is not actually related to (Taylor et al., 2007) apart from the use of the same Half-Life 2 *mod*.

In fact, we created an application to augment the functionalities of such a *mod* with the possibility of moving a virtual camera as if we were driving. In particular, in order to *drive* through a virtual city we introduced a camera with a given height as well as pitch, roll and yaw angles, and then we move it keeping these parameters constant. The only constraint that must be ensured is that these parameters are compatible with a camera forward facing the road from inside a vehicle, for instance, as if it was placed at the rear view mirror behind the windshield. Finally, in order to emulate the dataset of Daimler, we set the resolution of our virtual camera to 640×480 pixels.

As in (Marín et al., 2010) we created four virtual cities which, in fact, correspond to a single one in terms of graphical primitives, *i.e.*, we only changed some building, ground and object textures so that they look different as well as the overall illumination to emulate different daytimes. In order to introduce more variability in terms of pedestrians (aspect ratio and clothes) in these cities with respect to the ones in (Marín et al., 2010), we added new human models into the game, finally achieving a set of 60 different pedestrians (see Figure 4). Note that since they are articulated moving models seen from a moving camera, each virtual pedestrian can be imaged with different poses and backgrounds. Figure 3 plots samples of the virtual training set that we describe in the rest of

Figure 3: Examples and counterexamples taken from real (Daimler's dataset) and virtual images.

this section.

Examples. We recorded forty video sequences by driving through the virtual cities. In total we obtained 100075 frames, at 5fps, which corresponds to 5 hours, 33 minutes and 35 seconds. The virtual car was driven without any preferred *plan of route*. Along the way we captured images containing pedestrians in different poses and with different backgrounds. Since we can obtain the groundtruth of the virtual pedestrians automatically, we consider only those upright, non-occluded, and with a height equal or larger than 72 pixels in the captured images (pedestrians *taller* than 72 pixels require further down scaling as we will see) like in the training set of Daimler database. This gives us 7973 pedestrians to consider in order to construct the set of examples for training. It is worth mentioning that in order to have automatically labelled examples analogous to the manually labelled ones of Daimler's training set (*i.e.*, with the torso centered with respect to the horizontal axis), we cannot just take the bounding boxes corresponding to the pixel-wise groundtruth. Instead, we apply the following process to each virtual pedestrian:

1. For some pedestrian poses, the bounding box obtained from the pixel-wise groundtruth is such that the torso is not well centered in the horizontal axis, so we automatically correct this. More specifically, we project the pedestrian groundtruth into its horizontal axis. Then we take the location of the maximum of the projection as the horizontal center of the torso. Finally, we shift the initial pixel-wise bounding box so that its horizontal center matches the one of the torso.

2. Then, we modify the location of the sides of the pedestrian bounding box preserving the previous re-centering, but enforcing the same aspect ratio and proportional background margins than the pedestrians in the training set of Daimler (*i.e.*, 24/72 and 12/72, respectively). This is automatically achieved by simply applying standard rule of proportionality.

3. The bounding box at this point can still be larger than the canonical bounding box of the pedestrian examples of Daimler's training set, *et al.*, larger than 48×96 pixels. Thus, the final step consists in performing a down scaling using bi-cubic interpolation.

Counterexamples. In order to collect the counterexamples for training, we used the same four virtual cities than to obtain the examples, but now without pedestrians inside, *i.e.*, we drove through these *uninhabited* cities to collect pedestrian-free video sequences. We collect frames from these sequences in a random manner but assuring a minimum distance of five frames between any two selected frames, which is a simple way to increase variability.

	Training Set Cropped pedestrians (jitter and mirroring included) & Background frames	**Training process** 1st round: cropped pedestrians / cropped background & Bootstrapping: additional cropped background	**Testing sets**
Daimler	15660 & 6744	15660 / 15560 & All False Positives	Full set: 21790 frames
Virtual	1440, 4320, 12960 & 1219	1440, 4320, 12960 / 2438 & All False Positives	Mandatory set: 973 frames

Table 1: Training and testing settings for our experiments.

In fact, the images where taken with an initial resolution of 720×1280 pixels, and laler scaled to 480×640. In total we have 1219 frames without virtual pedestrians, so they can be sampled to gather virtual counterexamples. The sampling process is, of course, the same than the one previously described for Daimler (Sect. 2.1.1).

The video sequences were taken with the highest graphics quality allowed by the game engine. In particular, the changed settings were: the model detail, which controls the number of polygons and extra detailing; the texture detail; the color correction, which increases the realism; the antialiasing mode, which smooths the jagged lines when rendering; the high definition rate (HDR), which gives a higher vivid and contrasting lighting; and the filtering mode, which determines how clear is the detail of the textures as they fade into the distance to the camera.

3 Experiment design

3.1 Pedestrian detector components

In order to detect pedestrians we need a pedestrian classifier learnt from the training set by using specific *features* and a *learning machine*. With this classifier we *scan a given image* looking for pedestrians. Since multiple detections can be produced by a single pedestrian, we also need a mechanism to *select the best detection*. The procedures we use for features extraction, machine learning, scanning the images, as well as selecting the best detection from a cluster of them, are the same no matter if the classifier was learnt using virtual images or real ones (*et al.*, from Daimler). Let us briefly review which are these components in our case.

3.1.1 Features and learning machine.

The combination of the histograms of oriented gradients (HOG) features and linear SVM learning machine, proposed by Dalal *et al.* in (Dalal & Triggs, 2005), has been proven as a competitive method to detect pedestrians in the ADAS context (Enzweiler & Gavrila, 2009). Similar conclusions are also obtained when using a large ADAS-inspired dataset for testing in (Dollár et al., 2009). In fact, recent proposals that outperform HOG/linear-SVM when using the INRIA dataset include both HOG and linear SVM as core ingredients (Wang et al., 2009). Thus, we think that HOG/linear-SVM stands as a relevant baseline method for learning pedestrian classifiers, so we use it in our experiments. In particular, we follow the settings suggested in (Dalal & Triggs, 2005) for both HOG and linear SVM, as it is also done in (Enzweiler & Gavrila, 2009). A minor difference comes from the fact that in Daimler's datasets the images are grayscale while the virtual images are RGB. This issue is easily handled by just taking at each pixel the gradient orientation corresponding to the maximum gradient magnitude among the RGB channels (as in (Dalal & Triggs, 2005) for INRIA dataset).

3.1.2 Scanning strategy.

As in (Enzweiler & Gavrila, 2009) we use the widely extended *sliding window* strategy implemented through a pyramid to handle different detection scales (Dalal, 2006). We could consider the sliding window parameters

found in (Enzweiler & Gavrila, 2009) as the best in terms of pedestrian detection performance for the so-called *generic pedestrian detection* case with Daimler's testing set. However, at this stage of our research we decided to follow the settings proposed in (Dalal, 2006).

3.1.3 Selecting the best detection.

In order to group multiple overlapped detections and (ideally) provide one single detection per pedestrian we follow an iterative confidence- and overlapping- based approach, *i.e.* a kind of *non-maximum-suppression*. This technique, used by I. Laptev in (Laptev, 2009), consists of four basic steps: 1) create a new cluster with the detection of highest confidence; 2) recompute the cluster with the mean of the detections overlapping the new cluster; 3) iterate to step 2 until the cluster position does not change; 4) delete the detections contained in the cluster and iterate to 1 while there are detections.

For us a *pedestrian detector* consists of a pedestrian classifier, plus the above seen techniques of sliding window and non-maximum-suppression. Therefore, we are not considering tracking of pedestrians, but we think this does not affect the aim of this chapter.

3.2 Training

3.2.1 Training with Daimler dataset

We train the HOG/linear-SVM classifier with the 15660 provided examples and collect also 15560 counterexamples by sampling the 6744 provided pedestrian-free images as explained in Sect. 2.1.1, which is the approach followed in (Enzweiler & Gavrila, 2009). In addition, we also apply one *bootstrapping* step, *i.e.*, with the first learnt classifier we run the corresponding pedestrian detector on the 6744 pedestrian-free frames and collect false positives to enlarge the number of counterexamples and retrain. This bootstrapping technique is known to provide better classifiers (Enzweiler & Gavrila, 2009; Munder & Gavrila, 2006; Dalal & Triggs, 2005) than simply train once. In our case, instead of collecting 15660 false positives, like in (Enzweiler & Gavrila, 2009), all the false positives considered by the initial classifier are collected during bootstrapping. Thus, the final classifier is trained using 15660 examples and over 90000 counterexamples. By following such technique we found to increase the performance of the final detector.

3.2.2 Training with virtual dataset

Learning a classifier by using the virtual training set is analogous to the Daimler case, *i.e.*, HOG/linear-SVM and one bootstrapping stage are used. In our experiments we do not only explore the feasibility of using virtual data for learning a reliable pedestrian detector, but also the total number and variability between pedestrian models that is required and how the pose influences the detection. First, we test how many different models are at least required to obtain similar performance as the real dataset. Later, given a fixed number of models, we assess the performance of different number of examples, in particular, 1440, 4320, and 12960. Next, different adapted pose distributions are generated and compared one another. And finally, a final detector is trained using real and virtual data.

In order to assess how robust the training process is to changes, when using different training examples coming from the same virtual world, we perform a random selection of the total number of labelled pedestrians into five subsets with the aim of conducting five trainings regarding each of the experiments described above in which every model is proportionality represented in each random subset. Then, from the 80000 total pedestrians annotated ($\geq$ 72 pixels tall) to conduct our experiments we extract a thousand of them -in this case 1080, the closest multiple of total number of pedestrians, 60 (see Figure 4)- per training subset. In order to emulate Daimler training set, we apply two jitters and a mirroring per jitter, which means that in total every subset contains 4320

Figure 4: The next figure shows a sample of each of the pedestrian models used in our virtual world. In total there are sixty different models.

examples. Note that the randomness when all the models are used comes in terms of pose and illumination, while in the case of reduced number of models, it comes also from the pedestrian models themselves (the ones that have been selected).

Moreover, to ensure a certain variability on the subsets of examples, every subset is generated so that every pedestrian model selected has a gap of five frames between examples, thus making the differences come from pose and background.

In order to generate the different pose distribution sets, we first define the aspect-ratio of the legs in each pedestrian sample. This is the horizontal projection length of the bottom of the shape-mask image -a quarter of the image- divided by the total length of the bottom window (see bottom right image in Figure 6 (a)). Then, we split the aspect-ratio into nine different ranges. Later, we define four different distributions in terms of aspect-ratio (Uniform, Normal050, Gamma025 and Gamma075 -the numbers 025, 050 and 075 related to the distribution names, represent the peak of each distribution-), which define the way we sample the training samples with respect to their pose. Finally, we randomly sample subsets based on the adapted distributions through the defined ranges. In Figure 6 (b) we show the sampling distributions for Gamma025, Gamma075, Normal050, and Uni-

Figure 5: A virtual pedestrian and its variability through different backgrounds and illuminations.

form. For instance, in Gamma025 more virtual pedestrians with closed legs than opened ones are selected, while in Gamma075 occurs the opposite.

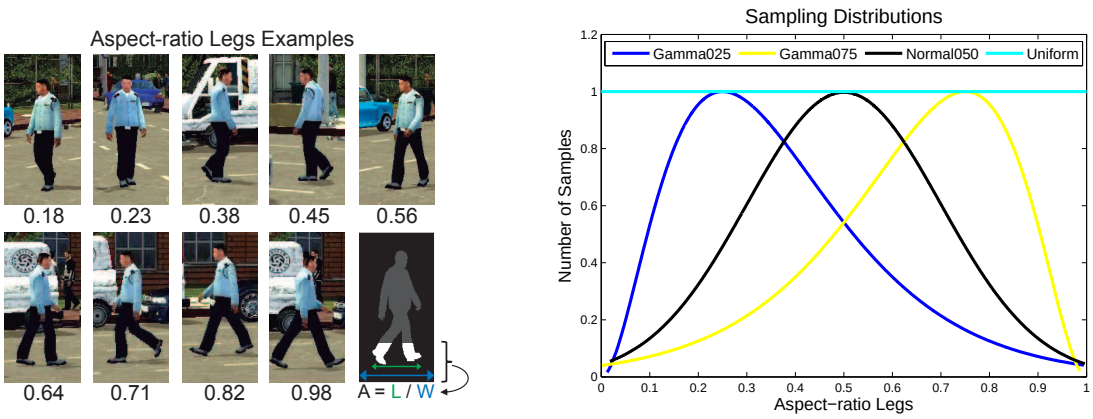
3.3 Testing

In order to reduce the computational time of the experiments, rather than using the 21790 testing frames from Daimler, we rely on a representative but reduced testing set as performed in (Marín et al., 2010). Specifically, those frames in which there is at least a mandatory pedestrian to detect (2.1.2) are first selected, then one every two frames is taken out. This final set of frames is considered as *mandatory testing set*. There are 973 of such frames and they contain 1193 mandatory pedestrians.

We use the mandatory testing set to evaluate all the pedestrian detectors associated to the classifier learnt with Daimler's training set and the same for the other detectors related to the virtual training set.

4 Results

In this section we describe the obtained results following the settings summarised in the previous section. To assess the performance of the different pedestrian detectors we use *per-image evaluation*. We choose this evaluation methodology as it has been recommended in (Dollár et al., 2009). Besides, *per-image* evaluation has also been used in Daimler's dataset for testing. In our case, instead of plotting curves depicting the tradeoff between detection rate (*i.e.*, the percentage of mandatory pedestrians that are actually detected) and the number of false positives per image (FPPI) in logarithm scale, we choose to plot missrate detection rate versus FPPI both in logarithm scale, similar as (Dollár et al., 2009). Moreover, the range chosen for plotting such curves is restricted to $[10^{-1}, 10^{0}]$, which means that we focus our interest in those regions that allow between one false positive per ten images and one FPPI. As a quantitative value in order to compare the performance between curves we compute



(a) (b)

Figure 6: Aspect-ratio legs distributions. (a) Examples with different aspect-ratio legs values of the same model - the bottom right image in (a) shows how the aspect-ratio is computed. (b) Sampling distributions: Gamma025, Gamma075, Normal050, and Uniform.

the area under the curve (AUC) average in such range.

In our experiments we advocate to use PASCAL criterion as it is still the most common criterion used in object recognition. Let define W_d as a pedestrian detector output detection window, and let W_l be a window labelled as mandatory pedestrian. We define the ratio areas: $r(W_d, W_l) = a(W_d \cap W_l)/a(W_d \cup W_l)$. Then, if there is a W_l for which $r(W_d, W_l) > 0.5$, W_d is considered a true positive, otherwise, W_d is a false positive. Undetected mandatory pedestrians count as false negative, *i.e.*, those W_l for which there is no W_d with $r(W_d, W_l) > 0.5$. If given a W_l more than one W_d passes the true positive criterion (*i.e.*, multiple detections), only one of them is considered, and the rest are considered as false positives. Note that such criterion also affects training because of the bootstrapping.

Figure 7 shows the obtained performance curves of the virtual and real approaches following the PASCAL criterion. In Figure 7 (a) the mean and the standard deviation of the 5-experiment-based of the virtual based training are shown. The virtual approach is conducted by using a fixed number of 4320 positive samples with all the pedestrian models proportionally represented. The standard deviation illustrates how robust the approach is when generating different random subsets, being its value 0.66 points. In Figure 7 (b) the best curve and the worst curve are plotted versus the Daimler baseline approach. As it can be seen, the results reveal the similarity of both datasets. The difference between the best virtual-world-based curve and the real-world-based one less than one point, and comparing the worst virtual-world-based curve and the real-world-based one such difference is still less than five points when comparing the AUC average. Altogether, the results, as presented in (Marín et al., 2010), allow us to contemplate that the differences of the learnt virtual-world-based detector and the real-world-based one are minimal in terms of performance. Figure 11 shows some qualitative results when using the real-world-based and virtual-world-based detectors. As expected, both detectors are quite similar in particular detections, however, they seem to slightly differ in the false positives.

Next we show the performances when using different number of models in the training procedure. Figure 8 (a) reproduces the different curves when using 2, 5, 10, 15, 30 and 60 pedestrians models. In this case, the mean of each experiment related to the number of pedestrian models is shown. As the AUC average indicates, the performance tends to saturate when 15 models are used. Note that these models have been captured through different illumination, background and pose (see Figure 5). In figure 8 (b), we can see that when the number of

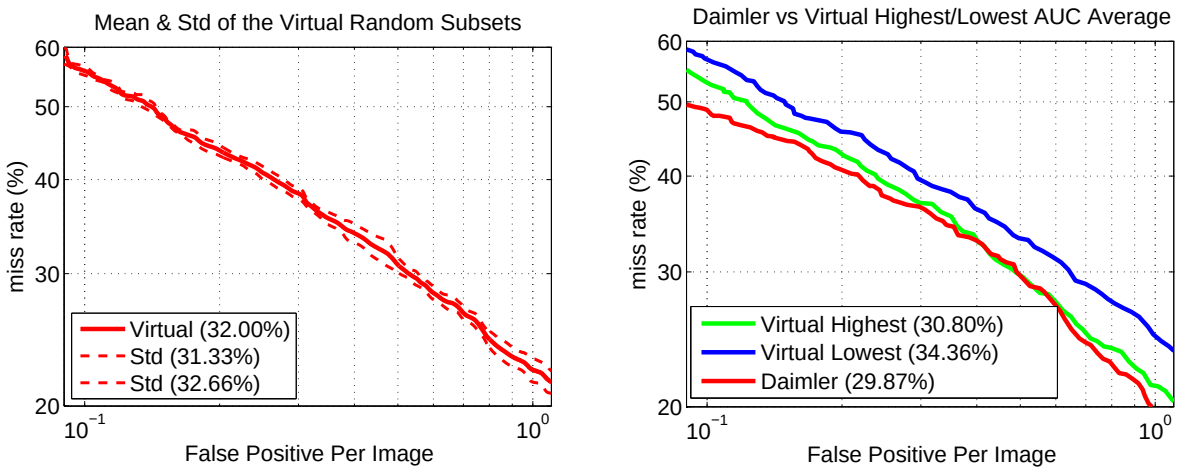


Figure 7: Per-image evaluation of the pedestrian detectors following PASCAL criterion (Everingham & al., 2006). Left: mean performance and standard deviation obtained in the 5 experiments with virtual samples and the sixty models over the mandatory testing set. Right: performances of virtual highest and lowest AUC average versus Daimler performance. The percentage between parenthesis next to each curve description represents the AUC average between range $[10^{-1}, 10^{0}]$.

models is reduced the standard deviation increases: when using a number of 15 the standard deviation value is 1.89 points, while when using 2 the value is 12.36 points. While the average performance when using a reduced number of pedestrian models is worse than the ones with 30, and 60, in some cases detectors learnt with 10 models achieve almost the same performance than the ones trained with more models. This reflects that depending on the models selected to train we can achieve a higher or lower performance, meaning that some virtual pedestrians suit more than the others real ones, always in terms of HOG features.

Figure 9 (a) shows the performance of the approaches when training with different total number of positive examples. In this case the curves show the mean of five random experiments when training with 1440, 4320, and 12960. The results manifest that training with 1440 is not enough to achieve the desired performance, while training with 4320 or 12960 do. Accordingly, the classifier converges with already 4320. In Figure 9 (b) we show the different performance between detectors trained on real, virtual and their combination. The performance when using both data altogether outperform the ones using separate data. In this case, four points better than just using independent sets. Indeed, it seems that real and virtual data can be complementary, outperforming in this case both detectors. This complementarity could be explained by the fact that both detectors detect almost the same pedestrians and fail in different background as already mentioned.

Next experiments evaluate the effect of the pose of training models in the performance. In such experiments we model four different distributions to assess their behaviour when testing on real images. Figure 10 (a) shows the different mean performances versus the original one (with no selection). As it can be seen, the performance that suits more the dataset and has closer performance to the original one is the called Gamma025, which samples are more likely to be in a close legs pose than a lateral walking one. Figure 10 (b) shows the comparison between the original and the Gamma025 distributions, in which it can be seen the similarity.

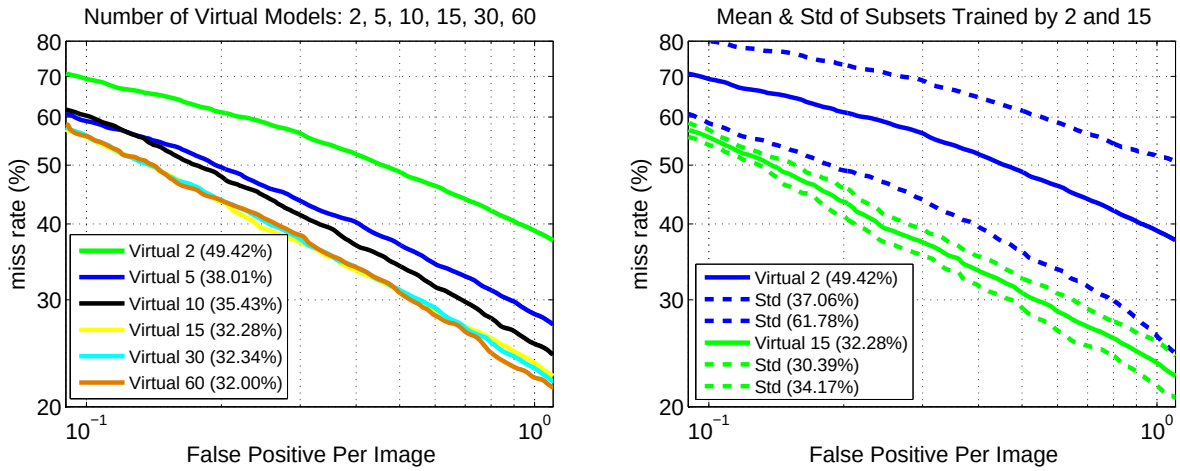
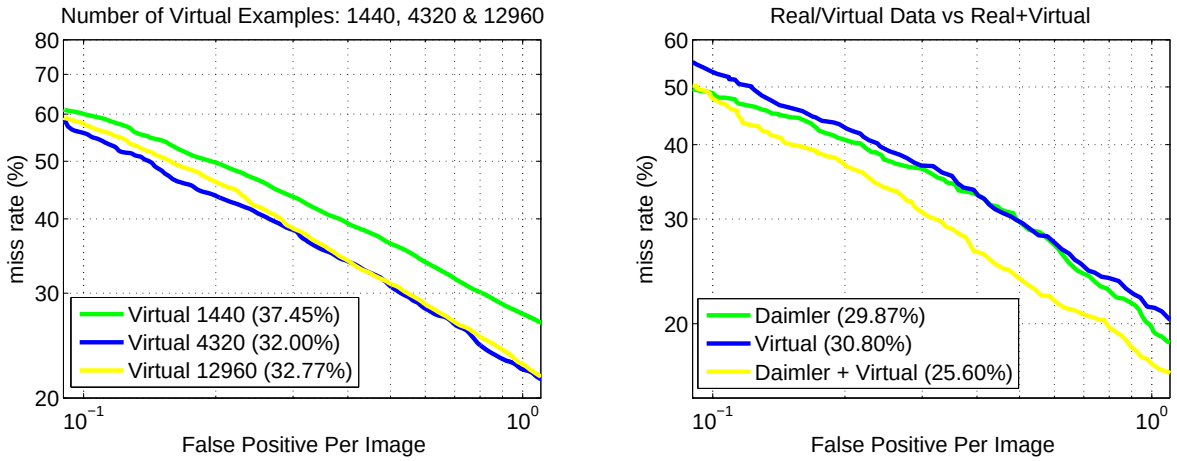


Figure 8: Per-image evaluation of the pedestrian detectors using different number of different models. (a) mean performances obtained in the 5 experiments with virtual samples for each fixed number of pedestrian models: 2, 5, 10, 15, 30 and 60. (b) mean and standard deviation of the subsets generated by 2 and 15 models. Both plots have been obtained over the mandatory test.

5 Conclusions

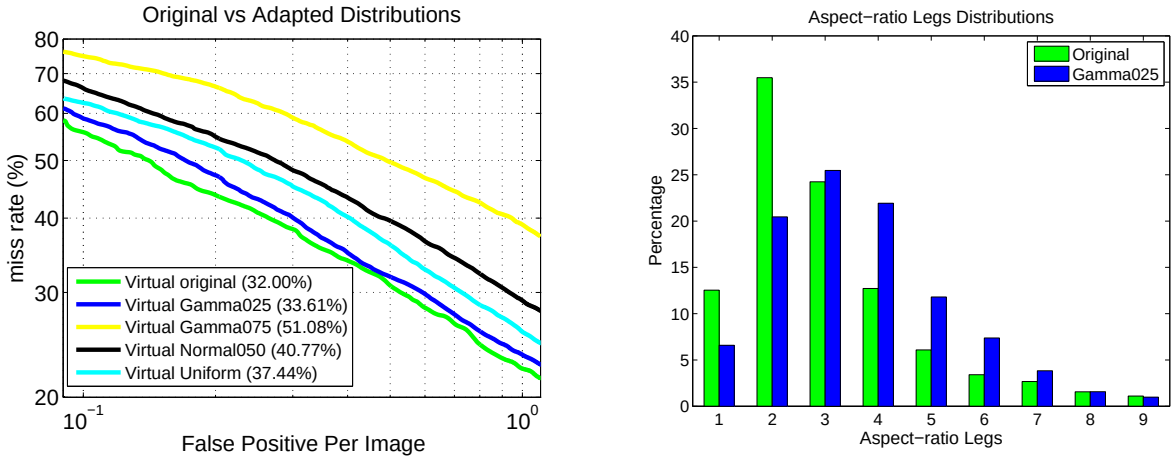
In this chapter we have explored the potential of virtual worlds in order to train appearance-based models for pedestrian detections in ADAS. The machine learning algorithm used to classify in our experiments is the linear SVM based on HOG features, a *de facto* standard in pedestrian detection. The whole detection pipeline consists of a sliding-window, the classification and finally, a non-maximum-suppression procedure. We first compare the virtual detector versus the real one in real images, and conclude that both performances are fairly the same. Several experiments with different virtual datasets to explore the possibilities in pedestrian detection, and concretely in appearance, that are carried to assess what virtual data can offer. In particular, we demonstrate that just few virtual pedestrians can achieve almost the same performance of the detectors does not improve, so adding new models does not have an effect. This same conclusion can be seen in (Pishchulin et al., 2011a). Besides, we investigate whether increasing the total number of examples used can benefit the detection or not. In this case the obtained results show that the performance converges. However, this fact can come from the machine learning and features used, so in future experiments we will test other widely used features such as Haar-like (Viola & Jones, 2001) features or LBP (Ojala et al., 1994). Finally, we study how the pose in pedestrians can influence in detection. The different pose distributions used proves that, in this specific case, the original distribution, in which the number of pedestrians walking across the camera is small compared to the others (*i.e.*standing/front-rear), achieves the best performance. These last results expose once again the similarity between virtual and real data when performing urban scenarios. Besides, combination results show the complementarity of real and virtual, outperforming the baseline. Therefore, the work done so far indicates that the virtual scenario generation stage is an important key to achieve state-of-the-art performance.

As future work we plan to test the proposed approach with more databases, features and learning machines. Specifically, we plan to do a more custom design of the classifier to obtain even better performance, *e.g.*, by following active learning approaches. In addition, the detection of other targets such as vehicles can also be evaluated under the proposed framework.



(a) (b)

Figure 9: Per-image evaluation of the different experiments. (a) Comparison of virtual detectors using different number of training examples: 1440, 4320, and 12960 pedestrians. (b) Comparison of different detectors trained on real and virtual data separately, and altogether.



(a) (b)

Figure 10: Experiments on the pose variation. (a) shows the different performances between the adapted distributions and the original one. (b) shows the comparison between pedestrian poses found in the original distribution and the gamma025 distribution. Each bar k, with $k \in \{1,\ldots,9\}$, shows the percentage of pedestrians that belong to the aspect-ratio range $[0.1k, 0.1(k+1)[$.

Classifier trained with real-world samples.

Classifier trained with virtual-world samples.

Figure 11: Qualitative results at 10^0 FPPI taken when following PASCAL VOC criterion. Top row: using the pedestrian detector based on Daimler's training set. Bottom row: using the pedestrian detector corresponding to the training with virtual samples, in particular with the classifier of highest detection rate at 10^0 FPPI. Green bounding boxes are right detections, yellow ones are false positives and red ones misdetections.

Acknowledgments

This work is supported by Spanish MICINN projects TRA2011-29454-C03-01, TIN2011-29494-C03-02, Consolider Ingenio 2010: MIPRCV (CSD200700018), and Javier Marín FPI Grant BES-2008-007582.

References

Bishop, R. (2005). *Intelligent Vehicle Technologies and Trends.* Artech House, Inc.

Broggi, A., Fascioli, A., Grisleri, P., Graf, T., & Meinecke, M. (2005). Model–based validation approaches and matching techniques for automotive vision based pedestrian detection. In *Object Tracking and Classification in and Beyond the Visible Spectrum Workshop, Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1–1). San Diego, CA, USA.

Dalal, N. (2006). *Finding People in Images and Videos.* PhD Thesis, Institut National Polytechnique de Grenoble / INRIA Rhône-Alpes.

Dalal, N. & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 886–893). San Diego, CA, USA.

Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2009). Pedestrian detection: A benchmark. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 304–311). Miami Beach, Florida, USA.

Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2011). Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34(4), 743–761.

Enzweiler, M. & Gavrila, D. (2008). A mixed generative-discriminative framework for pedestrian classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1–8). Anchorage, AK, USA.

Enzweiler, M. & Gavrila, D. (2009). Monocular pedestrian detection: Survey and experiments. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(12), 2179–2195.

Everingham, M. & al. (2006). The 2005 PASCAL visual object classes challenge. In *Proceedings of the First PASCAL Challenges Workshop, LNAI, Springer-Verlag*.

Gandhi, T. & Trivedi, M. M. (2007). Pedestrian protection systems: issues, survey, and challenges. *IEEE Trans. on Intelligent Transportation Systems*, 8(3), 413–430.

Gerónimo, D., López, A. M., Sappa, A. D., & Graf, T. (2010). Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32(7), 1239–1258.

Grauman, K., Shakhnarovich, G., & Darrell, T. (2003). Inferring 3d structure with a statistical image-based shape model. In *Proc. IEEE Int. Conf. on Computer Vision* (pp. 641–648). Nice, France.

Laptev, I. (2009). Improving object detection with boosted histograms. *Image and Vision Computing*, 27(5), 535–544.

Lin, Z. & Davis, L. (2008). A pose-invariant descriptor for human detection and segmentation. In *Proc. the European Conf. on Computer Vision* (pp. 423–436). Marseille, France.

Marín, J., Vázquez, D., Gerónimo, D., & López, A. M. (2010). Learning appearance in virtual scenarios for pedestrian detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 137–144). San Francisco, California, USA.

Munder, S. & Gavrila, D. (2006). An experimental study on pedestrian classification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 28(11), 1863–1868.

Ojala, T., Pietik ainen, M., & Harwood, D. (1994). Performance evaluation of texture measures with classification based on Kullback discrimination of distributions. In *Proc. Int. Conf. in Pattern Recognition* (pp. 582–585). Jerusalem, Israel.

Pishchulin, L., Jain, A., Wojek, C., Andriluka, M., Thormaehlen, T., & Schiele, B. (2011a). Learning people detection models from few training samples. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1473–1480). Colorado Springs, USA.

Pishchulin, L., Jain, A., Wojek, C., Thormaehlen, T., & Schiele, B. (2011b). In good shape: Robust people detection based on appearance and shape. In *22nd British Machine Vision Conference (BMVC)* Dundee, UK. Oral.

Taylor, G., Chosak, A., & Brewer, P. (2007). OVVV: Using virtual worlds to design and evaluate surveillance systems. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 1–8). Minneapolis, MN, USA.

Valve Software Corporation (2004).

Tuzel, O., Porikli, F., & Meer, P. (2008). Pedestrian detection via classification on Riemannian Manifold. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(10), 1713 – 1727.

UN-ECE (2007). *Economic Comision for Europe - Statistics of Road Traffic Accidents in Europe and North America*, volume LI. United Nations.

Viola, P. & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 511–518). Kauai, HI, USA.

Wang, X., Han, T., & Yan, S. (2009). An HOG–LBP human detector with partial occlusion handling. In *Proc. IEEE Int. Conf. on Computer Vision* (pp. 32–39). Kyoto, Japan.

Wojek, C., Walk, S., & Schiele, B. (2009). Multi-cue onboard pedestrian detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition* (pp. 794–801). Miami Beach, Florida, USA.