Multi-stage Sampling with Boosting Cascades for Pedestrian Detection in Images and Videos

Giovanni Gualdi, Andrea Prati, and Rita Cucchiara

University of Modena and Reggio Emilia*, Italy
{giovanni.gualdi,andrea.prati,rita.cucchiara}@unimore.it

Abstract. Many works address the problem of object detection by means of machine learning with boosted classifiers. They exploit sliding window search, spanning the whole image: the patches, at all possible positions and sizes, are sent to the classifier. Several methods have been proposed to speed up the search (adding complementary features or using specialized hardware). In this paper we propose a statistical-based search approach for object detection which uses a Monte Carlo sampling approach for estimating the likelihood density function with Gaussian kernels. The estimation relies on a multi-stage strategy where the proposal distribution is progressively refined by taking into account the feedback of the classifier (i.e. its response). For videos, this approach is plugged in a Bayesian-recursive framework which exploits the temporal coherency of the pedestrians. Several tests on both still images and videos on common datasets are provided in order to demonstrate the relevant speedup and the increased localization accuracy with respect to sliding window strategy using a pedestrian classifier based on covariance descriptors and a cascade of Logitboost classifiers.

Keywords: fast pedestrian detection, fast object detection, boosting classifiers, stochastic object detection, statistical object detection, Monte Carlo sampling, multi stage object detection.

1 Introduction and Related Works

Object detection and recognition in images and videos are problems that have been strongly addressed in computer vision in the past years. Some object classes (such as faces, pedestrians, vehicles, characters) received special attention by the research community, since their peculiarities can be "easily" modeled with machine learning techniques and classifiers can be efficiently exploited.

These classifiers are applied on image patches (or "windows") of a given size and in case the object is searched on a whole image, a sliding window search (e.g. [1,2,3]) is normally proposed. The algorithm passes to the window-based classifier all possible windows of an image; the approach has the drawback of brute force methods, that is the high computational load due to the number of windows to check, that grows quadratically in each dimension to span over (typically

* Giovanni Gualdi and Rita Cucchiara are with DII; Andrea Prati is with DISMI.

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part VI, LNCS 6316, pp. 196–209, 2010.

three, i.e. image coordinates and scale) [4]. Obviously, this computational load grows up if the search is performed on all the frames of a video. Consequently, several works focus on the reduction of the computational burden, following three main streams: (a) pruning the set of sliding windows by exploiting other cues (e.g. motion [5], depth [6], geometry and perspective [7], or whatever cue that is different from the appearance cue used by the detector itself); (b) speeding up with hardware-optimized implementations (such as GPUs [8]); (c) efficiently exploring the sub-window space through optimal solution algorithms [4,9].

In this paper, we address a new search paradigm to overcome the problem of the sliding window search in a general-purpose manner that does not conflict with all the other aforementioned optimizations (either hardware or software). The proposed method exploits a Monte Carlo sampling to provide an incremental estimation of a likelihood function and our innovative contribution is the use of the response/confidence of the classifier to build such likelihood function. In practice, this response is employed to increasingly draw samples on the areas where the objects are potentially present and avoiding to waste search time over other regions. Although we focus on cascade of boosting classifiers, where the classification is achieved by passing through the stages of the cascade, the proposal could be extended to any classifier that provides a classification confidence.

Mimicking the search of human vision, also [10,11] tackle the problem of optimized object detection. [10] explores the maximization of information gain: although it obtains speed-ups that are comparable to ours, two limitations are suffered: a slight degradation of performances w.r.t. sliding window detection (instead we obtain higher accuracy) and single-target detection (conversely our method is intrinsically multi-target). [11] proposes a deterministic (grid-distributed), multi-stage (coarse-to-fine) detection: successful detections at coarse resolutions yield to refined searches at finer resolutions. We also propose a multi-stage approach; however [11] binarizes the response of the classifier at each stage, while we propose to exploit its continuity, in order to be able to find true detections even when at earlier stages no successful detections are found.

When dealing with videos, the retrieved likelihood function is then plugged into a Bayesian recursive context, through a particle filter. Although this technique is often exploited for object tracking [12,13,14], our proposal does not aim to that achievement, rather it exploits the recursive framework to exploit the temporal coherency of the objects in order to further increase efficiency and accuracy of object detection. When the target distribution is multi-modal (due to ambiguity, clutter or presence of multiple targets), the particle filters suffer of the problem of *sample depletion*, and there are several extensions to handle multi-target tracking [15,16,17] or multi-modal posteriors, such as the *mixture particle filter* [18], where the different targets correspond to the modes of the mixture pdf. This approach has been refined in the *boosted particle filter* [17], where a cascaded Adaboost is used to guide the particle filter. The proposal distribution is a mixture model that incorporates information from both the Adaboost and the dynamical model of the tracked objects. Differently from other methods, we do not generate new particle filters together with the entrance of new objects in

the scene: indeed this approach can quickly degrade the performance due to the increase of the number of targets. On the opposite, our proposal is capable to handle a variable number of objects thanks to a quasi-random sampling procedure and a measurement model that is shared among the objects of interest.

Although our proposal is independent on the adopted classifier, on the employed features and on the target class, in this paper we focus on pedestrian detection where many accurate classifiers have been proposed and benchmarked. Dealing with pedestrian classification, a wide range of features has been proposed; among them, Haar wavelets [19], Histogram of Gradients (HoG) [2], a combination of the two [20], Shapelet [21], Covariance descriptors [3], etc.. Over these features, the most typical classifiers are SVMs (typically linear or histogram intersection kernels SVMs [22]) or the boosting algorithms (e.g. AdaBoost [1], LogitBoost [23], MPL Boost [24]) assembled in rejection cascades: this architecture benefits of the property to use a very reduced portion of the rejection cascade when classifying those patches whose appearance strongly differs from the trained model, reducing therefore the computational load. Conversely, the number of stages to pass through increases in a way that is proportional to the appearance similarity of the patch with the target model. An example of such classifier, that we adopt in the present work, is the covariance-descriptor LogitBoost pedestrian classifier proposed by Tuzel *et al.*[3].

Summarizing, the contribution of our work is two-fold. Firstly, by exploiting a known implementation of a boosting cascade classifier, we propose a new object detection approach (in particular, pedestrians) that challenges the typical sliding windows approach: we claim that, by exploiting the only features used by the classifier itself, it is possible to drive a more efficient exploration of the state space of an image. Secondly, we demonstrate that the data obtained by such method can be easily plugged into a Bayesian-recursive filter, in order to exploit the temporal coherency of the moving objects (pedestrians) in videos to improve detection in very cluttered environments. Results demonstrate a significant speedup together with a higher precision in the object localization.

Moreover, with regards to the literature (especially work in [13] that proposes a multi-stage sampling for object tracking) we proposed the following innovations: (i) we perform detection of multiple objects through a single likelihood model; (ii) we handle object entrances and exits; (iii) a new measurement for likelihood is proposed; (iv) a variable number of particles and variable covariances avoid "over-focusing" on true detections; (v) the likelihood is computed exploiting a portion of the samples instead of the whole set.

2 Pedestrian Detection Using a Cascade of LogitBoosts

For pedestrian detection, in [3] the authors proposed the use of a cascade classifier with a cue given by the covariance matrix of a 8-dimensional set of features F (defined over each pixel of I):

$$F = \left[x, y, |I_x|, |I_y|, \sqrt{I_x^2 + I_y^2}, |I_{xx}|, |I_{yy}|, \arctan \frac{|I_y|}{|I_x|} \right]^T \tag{1}$$

where x and y are the pixel coordinates, I_x,I_y and I_{xx},I_{yy} are respectively the first and the second-order derivatives of the image. Then, for any (rectangular) patch of I, the covariance matrix of the set of features F can be computed and used as "covariance descriptor" in the classifier. This descriptor lies on a Riemannian manifold, and in order to apply any classifier in a successful manner, it should be mapped over an Euclidean space. Without digging into details, a detection over a single patch involves the mapping of several covariance matrices (approx. 350) onto the Euclidean space via the inverse of the exponential map [3]: $\log_\mu(Y) = \mu^{\frac{1}{2}} \log\left(\mu^{-\frac{1}{2}} Y \mu^{\frac{1}{2}}\right) \mu^{\frac{1}{2}}$. This operator maps a covariance matrix from the Riemannian manifold to the Euclidean space of symmetric matrices, defined as the space tangent to the Riemannian manifold in μ, that is the weighted mean of the covariance matrix of the positive training samples. The mapping computation requires at least one SVD of an 8x8 matrix, and since such operation is computationally demanding, it is necessary to optimize the detection process.

To this aim, Tuzel *et al.*adopt a cascade of boosting classifiers and specifically, a set of LogitBoost classifiers based on logistic regressors in a rejection cascade manner. One of the advantages of such structure is computational: given the task of pedestrian detection on real world images and defined the set of windows (or bounding boxes) to test, only a small portion of them will run through the whole set of the LogitBoost classifiers; in fact, most of the patches are typically very dissimilar to the trained pedestrian model and will be rejected at the earlier stages of the cascade, reducing therefore the overall load of detection process.

Given a window w, defined by the 3-dimensional vector (w_x, w_y, w_s) (being respectively coordinates of the window center and window scale w.r.t. a given size; we assume constant aspect ratio), we introduce the *detection response* R as

$$R(w) = \frac{P(w)}{M} \tag{2}$$

where P is the index of the last cascade which provides a positive classification for w and M is the total number of cascades. Given the structure of rejection cascades, the higher the degree of response $R(w)$ is, the further w reached the end of the cascade, the more similar it is to the pedestrian model (up to the extreme of $R = 1$, that means successful classification). Tests over large sets of images in standard benchmarks show that the cascade of LogitBoost classifiers with covariance descriptors rejects most of the negative samples (80% of negative patches) within the first $\frac{1}{5}$ of the cascade (i.e. $R(w) < 0.2$ for 80% of generic negative patches).

Pedestrian detection at frame level is usually performed with a sliding window approach, i.e. a complete scanning of the "Sliding Windows Set" (SWS), that contains the windows at all possible window states (w_x, w_y, w_s). The cardinality of the SWS depends on the size of the image, on the range of scales to check and on the degree of coarseness for the scattering of the windows: regarding this latter parameter, to obtain a successful detection process, the SWS must be rich enough so that at least one window targets each pedestrian in the image. To be more precise, every classifier has a degree of sensitivity to small translations and

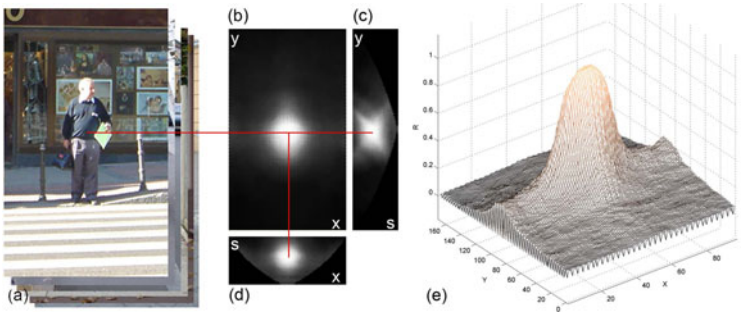


Fig. 1. Region of support for the cascade of LogitBoost classifiers trained on INRIA pedestrian dataset, averaged over a total 62 pedestrian patches; (a) a positive patch (pedestrian is 48x144); (b-d) response of the classifier: (b) fixed w_s (equal to 48x144), sliding w_x, w_y; (c) fixed w_x (equal to x of patch center), sliding w_s, w_y; (d) fixed w_y (equal to y of patch center), sliding w_x, w_s; (e) 3D plot of the response in (b).

scale variations, i.e. the response of the classifier in the close neighborhood (both in position and scale) of the window encompassing a pedestrian, remains positive ("region of support" of a positive detection). Having a sufficiently wide region of support allows to uniformly prune the SWS, up to the point of having at least one window targeting the region of support of each pedestrian in the frame. Vice versa, a too wide region of support could generate de-localized detections [4].

On this regard, an important advantage of the covariance descriptors is its relatively low degree of sensitivity to small translations and scale variations, i.e. its region of support over the positive detections was demonstrated to be higher with respect to many other descriptors (especially w.r.t. HoG). Its size depends on the training data, and the cascade of LogitBoost classifiers trained on the INRIA pedestrian dataset [3] shows a radius of such region of approximately 15% of the window size and 20% in the window scale (Fig. 1).

3 Multi-stage Sampling-Based Detection

The covariance-based pedestrian detector of [3] quantizes uniformly the state space with sliding windows, incurring in the two-fold problem of large waste in computational time searching over areas where pedestrians are not present and need of a redundant SWS to find every pedestrian in the scene.

Our objective is to provide a non-uniform quantization and to model the detection as an estimation of the states given the observations; we aim at estimating the modes of the continuous density function $p(\boldsymbol{X}|\boldsymbol{Z})$, where $\boldsymbol{X} = (w_x, w_y, w_s)$ is the state and $\boldsymbol{Z}$ corresponds to the image. In section 3.1 we introduce an approximation of the likelihood function, progressively improved through a multi-stage sampling-based process. Such procedure has the advantage to provide a global view of the landscape of the likelihood function and, at the same time, to support efficient sample placement. The likelihood allows pedestrian detection within the

single image. In section 3.2 we deal with pedestrian detection in videos, plugging the likelihood approximation method into a Bayesian-recursive filter.

3.1 Multi-stage Kernel-Based Density Estimation on a Single Image

Let's not consider any a prior information in the image (such as motion, geometry, depth, etc.) in order to provide a general solution. Consequently, the state pdf can be assumed proportional to the measurement likelihood function, i.e. $p(\boldsymbol{X}|\boldsymbol{Z}) \propto p(\boldsymbol{Z}|\boldsymbol{X})$.

The measurement likelihood function is estimated by iteratively refining it through m stages based on the observations. Algorithm 1 shows the complete procedure. The initial distribution $q_0(\boldsymbol{X})$ is set to a uniform distribution on the state space and it is sampled, extracting the first S_1 set of N_1 samples (see line 1 of Algorithm 1 and yellow points in the exemplar image of Fig. 2). Each sample s represents a state (w_x, w_y, w_s) in the domain of the windows. Scattering samples according to a uniform distribution is somehow similar to the sliding window strategy, though the samples are not equally distributed and their locations are not deterministically defined: indeed, the N_1 samples could also be grid-distributed without affecting the bottom line of the proposed method; instead, the key point here is N_1 be significantly lower than the cardinality of a typical SWS (see experiments in Section 4). The rationale is that part of these samples will fall in the basin of attraction of each region of support of the pedestrians in the image and will provide an initial rough estimation of the measurement function. Being driven by the previous measurements, at any stage i, the distribution q_i is progressively refined, to perform new sampling. This growing confidence over the proposal makes it possible to decrease, from stage to stage, the number of N_i to sample (see Fig. 2), differently from [13], where N_i is constant over stages.

The N_1 samples drawn from $q_0(\boldsymbol{X})$ (line 6) will be used to provide a first approximation of the measurement density function p_1, through a Kernel Density Estimation (KDE) approach with Gaussian kernel, generating a mixture of N_1 Gaussians: for each j-th component, mean, covariance and weight are defined as follows: the mean $\mu_i^{(j)}$ is set to the sample value $s_i^{(j)} = \left(w_{x,i}^{(j)}, w_{y,i}^{(j)}, w_{s,i}^{(j)}\right)$; the covariance matrix $\Sigma_i^{(j)}$ is set to a covariance Σ_i (line 8), which, at any given stage i, is constant for all samples. The work in [13] proposed to determine the Σ for each sample as a function of its k-nearest neighbors; this strategy yielded fairly unstable covariance estimations when applied to our context: indeed, given the low number of samples used in our method, k is to be kept pretty low (to maintain a significance over the covariance estimation), and this makes the estimation quite dependent on the specific randomized sample extraction. We preferred to assign an initial Σ_1 proportional to the size of the region of support of the classifier, and decrease the Σ_i of the following stages: this has the effect of incrementally narrowing the samples scattering, obtaining a more and more focused search over the state space.

Algorithm 1. Measurement Step

1: Set $q_0(\boldsymbol{X}) = U(\boldsymbol{X})$
2: Set $S = \emptyset$
3: **for** $i = 1$ **to** m **do**
4: **begin**
5: Draw N_i samples from $q_{i-1}(\boldsymbol{X})$:
6: $S_i = \left\{ s_i^{(j)} | s_i^{(j)} \sim q_{i-1}(\boldsymbol{X}),\ \ j = 1, \ldots, N_i \right\}$
7: Assign a Gaussian kernel to each sample:
8: $\mu_i^{(j)} = s_i^{(j)} \quad ; \quad \Sigma_i^{(j)} = \Sigma_i$
9: Compute the measurement on each sample $s_i^{(j)}$:
10: $l_i^{(j)} = R^{\lambda_i}\left(\mu_i^{(j)}\right)$ with $R^{\lambda_i} \in [0, 1]$
11: Obtain the measurement density function at step i:
12: $p_i(\boldsymbol{Z}|\boldsymbol{X}) = \sum_{\pi_i^{(j)} \neq 0} \pi_i^{(j)} \cdot \mathcal{N}\left(\mu_i^{(j)}, \Sigma_i^{(j)}\right)$
13: where: $\pi_i^{(j)} = \frac{l_i^{(j)}}{\sum_{k=1}^{N_i} l_i^{(k)}}$
14: Compute the new proposal distribution:
15: $q_i(\boldsymbol{X}) = (1 - \alpha_i)\, q_{i-1}(\boldsymbol{X}) + \alpha_i \frac{p_i(\boldsymbol{Z}|\boldsymbol{X})}{\int p_i(\boldsymbol{Z}|\boldsymbol{X}) d\boldsymbol{X}}$
16: Retain only the samples with measurement value 1:
17: $\widetilde{S}_i = \left\{ s_i^{(j)} \in S_i | R\left(\mu_i^{(j)}\right) = 1,\ \ j = 1, \ldots, N_i \right\}$
18: $S = S \bigcup \widetilde{S}_i$
19: **end**
20: Run variable-bandwidth meanshift (Non-Maximal-Suppression) over S. Obtain the set of modes $\mathcal{M}_1$
21: Prune the modes in $\mathcal{M}_1$ that do not represent reliable detection (see text). Obtain the new set of modes $\mathcal{M}_2$
22: Assign a Gaussian Kernel to each modes $\omega^{(j)} \in \mathcal{M}_2$ and compute the final likelihood function:
23: $p(\boldsymbol{Z}|\boldsymbol{X}) \propto \sum_{\forall \omega^{(j)} \in \mathcal{M}_2} \mathcal{N}\left(\omega^{(j)}, \overline{\Sigma}\right)$

Finally, the response R of the classifier (eq. 2) is exploited, in a novel way, to determine the weight $\pi_i^{(j)}$ of the j-th component. The intention is that those samples falling close to the center of any region of support (i.e., close to the mode/peak of the distribution) might receive higher weight with respect to the others, so that the proposal distribution q_i, that is partly determined by p_i, will drive the sampling of the next stage more toward portion of the state space where the classifier yielded high responses. Conversely, sampling must not be wasted over areas with low response of the classifier. In other words, these weights must act as attractors which guide the samples toward the peaks. This is accomplished by connecting the weights $\pi_i^{(j)}$ to the response R of the pedestrian detector in the sample location $\mu_i^{(j)}$ (line 10).

The exponent λ_i used to compute the measurement is positive and increases at every stage: at early stages, $\lambda_i \in (0; 1)$, therefore the response of the samples

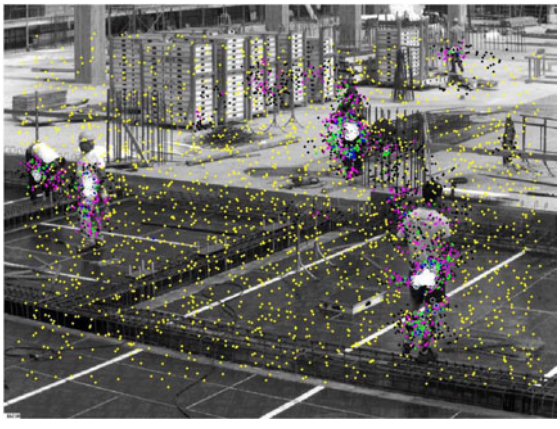


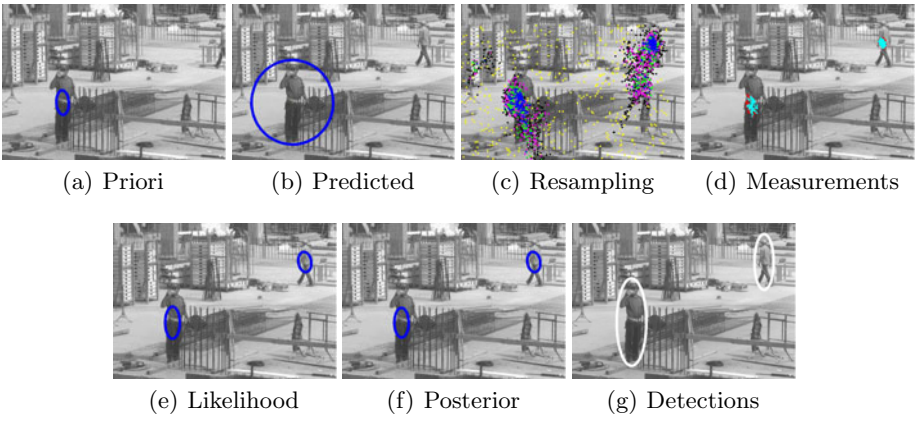
Fig. 2. Distribution of samples across the stages: $m = 5$ and $(2000, 1288, 829, 534, 349) = 5000$ samples. Stage order is yellow, black, magenta, green and blue. White circles represent the samples triggering a successful pedestrian classification.

is quite flattened, in order to treat fairly equally all range of not null responses; at later stages λ_i grows beyond 1, so that only the best responses will be held in account, while the others will be nullified. This behavior is clearly shown in Fig. 2 where the samples at subsequent stages (even if less numerous) are concentrated in the peaks of the distribution (i.e. where the response of the pedestrian detector is higher).

The Gaussian mixture of line 12 in Algorithm 1 is used as a partial estimation $p_i(\boldsymbol{Z}|\boldsymbol{X})$ of the likelihood function. This estimation is linearly combined with the previous proposal distribution $q_{i-1}(\boldsymbol{X})$ to obtain the new proposal distribution (line 15), where α_i is called *adaptation rate*.

The process is iterated for m stages and at the end of each stage only the samples of S^i that triggered a successful human detection (i.e. $R = 1$) are retained (line 17) and added to the final set of samples S (line 18). The samples retained in S are shown with white circles in Fig. 2. The number m of iterations can be fixed or adjusted according with a suitable convergence measure.

The non-maximal suppression is accomplished using a *variable-bandwidth mean-shift* suited to work on Gaussian mixtures [13], that provides a mixture of Gaussians representing in a compact way the final modes of the distribution. All those modes that contain less than τ_1 detections, or that contain less than $\tau_1/2$ strong detections are suppressed. Given the classification confidences provided by each LogitBoost classifiers of the cascade, a detection is considered strong if the minimum confidence is higher than a threshold τ_2. Numerical values are given in Sec. 4. The survived modes are considered successful detections and the derived mixture corresponds to the final likelihood function $p(\boldsymbol{Z}|\boldsymbol{X})$ (line 23).



(a) Priori (b) Predicted (c) Resampling (d) Measurements

(e) Likelihood (f) Posterior (g) Detections

Fig. 3. Multi-stage sampling in the context of Bayesian recursive filtering. In (c) the yellow dots represents the quasi-random sampling. The coloring is consistent with Fig. 2. The man on the upper-right corner is out of the influence of the predicted pdf, but the uniform component of eq. 5 allows some samples to fall within the region of support of that person and to act as attractors for the samples in the next stages. In (d), red dots represents successful detections, cyan dots are successful detections with high detection confidence.

3.2 Kernel-Based Bayesian Filtering on Videos

We extend here the previous method to the context of videos, by propagating the modes in a Bayesian-recursive filter. Differently from tracking approaches, the conditional density among frames (observations in time) is not used here to solve *data association*. Instead, the recursive nature of particle filtering exploits temporal coherence of pedestrians only to further improve detection. In the sequential Bayesian filtering framework, the conditional density of the state variable given the measurements is propagated through prediction and update stages as:

$$p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}\right) = \int p\left(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}\right) p\left(\boldsymbol{X}_{t-1}|\boldsymbol{Z}_{1:t-1}\right) d\boldsymbol{X}_{t-1} \tag{3}$$

$$p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t}\right) = \frac{p\left(\boldsymbol{Z}_t|\boldsymbol{X}_t\right) p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}\right)}{\int p\left(\boldsymbol{Z}_t|\boldsymbol{X}_t\right) p\left(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}\right) d\boldsymbol{X}_t} \tag{4}$$

The priori $p\left(\boldsymbol{X}_{t-1}|\boldsymbol{Z}_{1:t-1}\right)$ is propagated from the posteriori at the previous frame and for the first frame $p\left(\boldsymbol{X}_0|\boldsymbol{Z}_0\right)$ no prior assumptions are made and uniform distribution is employed. The predicted pdf is obtained (eq. 3) as the product of the priori with the motion model and then marginalizing on $\boldsymbol{X}_{t-1}$. Since in complex scenes correct motion model is unknow [25], we applied a zero-order function with Gaussian noise of fixed covariance.

Fig. 3 depicts the different steps of this procedure. The priori is convolved with white noise which has the only effect of increasing its covariance (producing the

Table 1. Benchmark

		# images	img size	# peds	peds size	avg peds/img
Tests on Images	INRIA [2]	288	333x531-1280x960	582	80-800px	2.02
	Graz02 [27]	310	640x480	620	55-410px	2.00
Tests on CWS Videos	Video 1	148	800x600 @ 1 fps	340	55-350px	2.30
	Video 2	114		398		3.49
	Video 3	68		83		1.22

predicted pdf - Fig. 3(b)). Differently from the case of single images, where q_0 is uniform, in videos, at each time t (i.e. frame), $q_0(\boldsymbol{X}_t)$ is obtained by applying a *quasi-random sampling* [26] to the predicted distribution p:

$$q_0(\boldsymbol{X}_t) = \beta \cdot p(\boldsymbol{X}_t|\boldsymbol{Z}_{1:t-1}) + (1-\beta) \cdot U(\boldsymbol{X}_t) \tag{5}$$

where β decides the amount of random sampling. The random sampling is crucial to detect new pedestrians entering the scene (Fig. 3(c)). Given q_0, the procedure described in the previous section is used to iteratively estimate the likelihood $p(\boldsymbol{Z}_t|\boldsymbol{X}_t)$ (Fig. 3(e)). Any newly detected likelihood mode is confirmed as a new-entry pedestrian detection. The quasi-random sampling is applied only to the proposal distribution q_0 (the proposal of the first stage of the multi-stage sampling). The likelihood and the predicted are multiplied to obtain (unless a normalization factor) the posterior pdf (see eq. 4).

4 Experimental Results

We performed extensive experimentation of the proposed multi-stage boosting method (*MSBoost* or MSB hereinafter) both on images and videos with fairly high resolution (rarely less than 640x480, up to 1280x960): in these conditions the sliding window (SW) can be very demanding, and the benefit of MSBoost is highlighted. Additionally, we are also considering a large range of scales since the considered images contain people of quite diverse sizes. Finally, tests on videos were carried out considering no other information than appearance (neither motion nor scene geometry). Experimental results are obtained on the benchmark reported in Table 1. In order to compare with the state of the art we used publicly available datasets which also provide ground-truth annotations. In the case of images, we have used the Graz02 dataset [27] and the well-known INRIA dataset [2]. Regarding the videos, we used 3 video clips taken from construction working sites (CWS), that contain on average 19 entrances/exits per video.

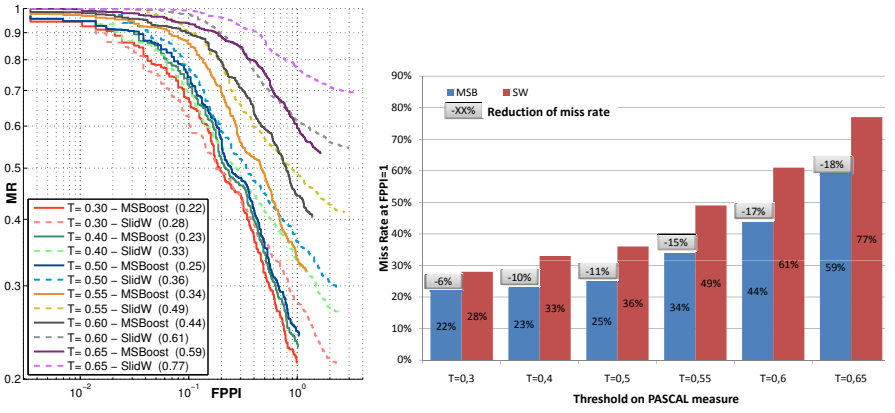
The accuracy of pedestrian detection is measured at object level in terms of the matching of the bounding box found by the detector (BB_{dt}) with the bounding box in the ground truth (BB_{gt}). A matching is found using the measure defined in the PASCAL object detection challenges [28] which states that the ratio between the area of overlap of BB_{dt} with BB_{gt} and the area of merge of the two BBs must be greater than a given threshold T, that is typically set to

50%; however, in some experiments we test the detection at lower and higher values of T, in order to better evaluate the localization accuracy of the detection of MSBoost w.r.t. SW. Throughout all tests, multiple detections of the same ground-truthed person, as well as a single detection matching multiple ground-truthed people, are affecting the performance in terms of recall and precision.

Regarding our approach, most of the tests has been performed using a total number of 5000 particles, divided over the $m = 5$ stages as follows: $N_i = NP \cdot e^{\gamma \cdot (i-1)}$, where $NP = 2000$ represents the initial number of particles (i.e., N_1), whereas γ is a constant factor (equal to 0.44 in our tests) which ensures that the number of particles diminishes over the stages in an exponential way. A similar approach is followed also for λ_i and Σ_i, which are the exponent for the measurement and the covariance for the Gaussian kernels, respectively. The starting values are 0.1 and $diag(7, 14, 0.125)$ (obtained considering the region of support, and with normalized scales) and the exponential constant are 1 and -0.66, respectively. Finally, the thresholds for the non-maximal suppression (see end of Section 3.1) have been set to $\tau_1 = 4.0$ and $\tau_2 = 4.0$. The first test on single images (INRIA dataset) aims at showing that MSBoost yields higher accuracy than SW in the detection localization; we measure detection performances varying the threshold T of the PASCAL challenge: the higher is T, the higher is the precision in detection accuracy required on the detector. In these tests MSBoost employs 15000 particles, while the SW uses a fixed position stride (10.9% of window size), employing on average 101400 (6.8 times more) windows per image, with peaks of 364000 (24.2 times more). Results are shown in Fig. 4. Fig. 4(a) highlights the trend of MR vs FPPI at different T: as expected, regardless of the detection paradigm, the higher is T, the higher is the MR. However, at any T, MSBoost shows lower MR than sliding window: moreover, as shown in Fig. 4(b), at increasing T, MSBoost decreases its performance in a lower degree w.r.t. SW; in other words, the detection localization of the former is higher and is achieved through the information gain obtained through the multi-stage sampling.

In the second test on single images (Graz02) we compared MSBoost vs SW at different number of windows. The scale stride is set to 1.2, the number of particles employed in MSBoost is 5000, while the number of windows in SW is 5000, 10000, 15000, 20000, 30000 and 50000 (corresponding respectively to a position stride of 15.6%, 10.9%, 9.8%, 8.2%, 7.1% and 5% of window size). The non-maximal suppression for SW is performed with mean shift and $\tau_2 = 2$. Tab. 2(a) shows the results achieved in terms of *False Positives Per Image* (FPPI) and *Miss Rate* (MR) as suggested in [29]. MSBoost with 5000 particles achieves a FPPI comparable to SW with 15000 windows, yielding an 8% lower MR.

Regarding the experiments on videos, we firstly aim at validating the usefulness of the Bayesian-recursive approach; we compared the FPPI and MR obtained on Video 1, by using the SW with 10000 windows, a non-recursive approach (Section 3.1 on each single frame) with 2500 particles and the Bayesian-recursive (Section 3.2) with varying number of particles (5000, 2500 and 1250); see Tab. 2(b). The Bayesian-recursive approach with 1250 particles yields similar FPPI and better MR w.r.t. non-recursive with 2500 particles. Moreover, it



(a) (b)

Fig. 4. Results on INRIA dataset at different values of T, the threshold on the bounding box matching. Number in brackets represent the MR at FPPI=1.

Table 2. Summary of results

(a) On Graz02 dataset

	FPPI	MR
SW (5000)	0.39	0.76
SW (10000)	0.66	0.57
SW (15000)	**0.73**	**0.51**
SW (20000)	1.08	0.46
SW (30000)	1.30	0.40
SW (50000)	1.66	0.37
MSB (5000)	**0.74**	**0.43**

(b) On Video 1

	FPPI	MR
SW (10000)	0.54	0.54
MSB (2500)	0.29	0.34
MSB rec (1250)	0.30	0.29
MSB rec (2500)	0.45	0.14
MSB rec (5000)	0.56	0.13
MSB rec (5000, no-exp decay)	0.56	0.16

(c) On Videos 1,2,3

	FPPI	MR
Video 1	0.56	0.13
Video 2	0.98	0.55
Video 3	0.42	0.78

obtains overall better performance than SW with 10000 windows. Eventually, we also evaluated the usefulness of the exponential decay of particles, as disabling it slightly reduces the performances. Then, to further validate our approach we tested Bayesian-recursive MSBoost on two other videos from the CWS dataset which contain several heavy occlusions of the pedestrians (see Table 1). Results are summarized in Tab. 2(c).

Regarding the computational load, when dealing with cascades of strong classifiers, the time cannot be considered simply proportional to the number of employed windows or samples. In fact, in traditional classifiers the classification time for each window is constant while in cascaded classifiers the time is reduced if the input is rejected at an intermediate level of the cascade. In this sense, the information gain across the multiple stages of the MSBoost produces samples that are increasingly closer to the positive classification and therefore the average number of strong classifiers that are successfully passed increases, raising the computation time: this is testified by the average R of MSBoost that is higher

than the one of SW (0.26 and 0.08 respectively). Nevertheless, MSBoost ends up being definitely faster because: (a) the time to prepare the integral image and the tensors for each patch (that is a fixed time for each window, regardless of its appearance or of the use of MSBoost or SW), is on average 8.92 times the average classification time of a random negative patch; (b) merging overhead and classification time, the per-particle computational load of MSBoost is 1.9 times higher than the per-window computational load of SW; (c) the experimental results demonstrate that MSBoost achieves higher detection accuracy with a number of particles that is from 3 to 10 times lower than the number of windows employed with SW. Thus, the measured computation time for MSBoost is from 1.8 to 5.4 times lower than for SW. This increase of performance is almost independent on the number of objects to be detected. On average MSBoost takes about 1 second to perform 5000 detections using a C++ implementation on a dual-core off-the-shelf PC, also by exploiting the intrinsic parallelization of the algorithm. The complete approach can process about 0.75 frames per second (fps) with 5000 particles, which can be proportionally increased by reducing the number of particles (e.g., it becomes about 3 fps with 1250 particles which give good results on Video 1 of CWS).

5 Conclusions

The work introduces a novel method to avoid the brute force strategy of sliding window for (pedestrian) detection in both images and videos; the proposed method works within the domain of appearance used by the classifier itself, exploiting the response of the boosting cascade to drive an efficient spanning of the state space and using a multi-stage sampling based strategy. The derived measurement function can be plugged in a kernel-based Bayesian filtering to exploit temporal coherence of pedestrian in videos. Experimental results show a gain in computational load maintaining same accuracy of sliding window approach.

References

1. Viola, P.A., Jones, M.J., Snow, D.: Detecting pedestrians using patterns of motion and appearance. IJCV 63, 153–161 (2005)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, vol. 1, pp. 886–893 (2005)
3. Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on riemannian manifolds. IEEE T-PAMI 30, 1713–1727 (2008)
4. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Efficient subwindow search: A branch and bound framework for object localization. IEEE T-PAMI 31 (2009)
5. Tao, J., Odobez, J.M.: Fast human detection from videos using covariance features. In: Workshop on VS at ECCV (2008)
6. Ess, A., Leibe, B., Schindler, K., van Gool, L.: Robust multiperson tracking from a mobile platform. IEEE T-PAMI 31, 1831–1846 (2009)
7. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. IJCV 80, 3–15 (2008)

8. Wojek, C., Dorkó, G., Schulz, A., Schiele, B.: Sliding-windows for rapid object class localization: A parallel technique. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 71–81. Springer, Heidelberg (2008)
9. Lehmann, A., Leibe, B., Van Gool, L.: Feature-centric efficient subwindow search. In: ICCV (2009)
10. Butko, N., Movellan, J.: Optimal scanning for faster object detection. In: IEEE Conference on CVPR 2009, pp. 2751–2758 (2009)
11. Zhang, W., Zelinsky, G., Samaras, D.: Real-time accurate object detection using multiple resolutions. In: IEEE Conference on ICCV 2007, pp. 1–8 (2007)
12. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE T-PAMI 25, 564–575 (2003)
13. Han, B., Zhu, Y., Comaniciu, D., Davis, L.S.: Visual tracking by continuous density propagation in sequential bayesian filtering framework. IEEE T-PAMI 31 (2009)
14. Isard, M., Blake, A.: Condensation - conditional density propagation for visual tracking. IJCV 29, 5–28 (1998)
15. Hue, C., Le Cadre, J.P., Perez, P.: Tracking multiple objects with particle filtering. IEEE Transactions on Aerospace and Electronic Systems 38, 791–812 (2002)
16. Isard, M., MacCormick, J.: Bramble: A bayesian multiple-blob tracker. In: ICCV, pp. 34–41 (2001)
17. Okuma, K., Taleghani, A., de Freitas, N., Little, J.J., Lowe, D.G.: A boosted particle filter: Multitarget detection and tracking. In: Pajdla, T., Matas, J(G.) (eds.) ECCV 2004. LNCS, vol. 3021, pp. 28–39. Springer, Heidelberg (2004)
18. Vermaak, J., Doucet, A., Pérez, P.: Maintaining multi-modality through mixture tracking. In: ICCV, pp. 1110–1116 (2003)
19. Papageorgiou, C., Poggio, T.: A trainable system for object detection. IJCV 38, 15–33 (2000)
20. Wojek, C., Schiele, B.: A performance evaluation of single and multi-feature people detection. In: Rigoll, G. (ed.) DAGM 2008. LNCS, vol. 5096, pp. 82–91. Springer, Heidelberg (2008)
21. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. In: CVPR, pp. 1–8 (2007)
22. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR, pp. 1–8 (2008)
23. Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Annals of Statistics 28, 337–407 (2000)
24. Babenko, B., Dollár, P., Tu, Z., Belongie, S.: Simultaneous learning and alignment: Multi-instance and multi-pose learning. In: Faces in Real-Life Images (2008)
25. Han, B., Comaniciu, D., Zhu, Y., Davis, L.: Incremental density approximation and kernel-based bayesian filtering for object tracking. In: CVPR (2004)
26. Philomin, V., Duraiswami, R., Davis, L.: Quasi-random sampling for condensation. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1843, pp. 134–149. Springer, Heidelberg (2000)
27. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. IEEE T-PAMI 28, 416–431 (2006)
28. Ponce, J., Berg, T., Everingham, M., Forsyth, D., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, C., Torralba, A., Williams, C., Zhang, J., Zisserman, A.: In: Dataset issues in object recognition, pp. 29–48. Springer, Heidelberg (2006)
29. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: CVPR, pp. 304–311 (2009)