THIS PAPER APPEARED IN THE ECCV VISUAL SURVEILLANCE WORKSHOP (ECCV-VS), MARSEILLE, OCTOBER 2008

Fast Human Detection from Videos Using Covariance Features

Jian Yao Jean-Marc Odobez

IDIAP Research Institute
Centre du Parc, Rue Marconi 19, CH-1920 Martigny, Switzerland
{Jian.Yao,Jean-Marc.Odobez}@idiap.ch

Abstract

In this paper, we present a fast method to detect humans from videos captured in surveillance applications. It is based on a cascade of LogitBoost classifiers relying on features mapped from the Riemanian manifold of region covariance matrices computed from input image features. The method was extended in several ways. First, as the mapping process is slow for high dimensional feature space, we propose to select weak classifiers based on subsets of the complete image feature space. In addition, we propose to combine these sub-matrix covariance features with the means of the image features computed within the same subwindow, which are readily available from the covariance extraction process. Finally, in the context of video acquired with stationary cameras, we propose to fuse image features from the spatial and temporal domains in order to jointly learn the correlation between appearance and foreground information based on background subtraction. Our method evaluated on a large set of videos coming from several databases (CAVIAR, PETS, ...), and can process from 5 to 20 frames/sec (for a 384x288 video) while achieving similar or better performance than existing methods.

1 Introduction

Detecting humans in images and videos is one of the important challenges in computer vision. This is due to factors such as the large variation of appearance and pose that human forms can take due to their clothing, the nature of articulations of the body, the changes in camera view point or illumination variations. In this paper, we address the fast detection of humans in videos recorded by a stationary camera. This is an essential step in many applications related to surveillance and smart spaces such as meeting rooms or offices. Indeed, improving human modeling and detection is crucial for tracking algorithms, especially when scenes become more crowded.

In general, there are two main approaches to tackle the detection of humans in images. The first consists of modeling the human by body parts whose locations are constrained by a geometric model [7, 9]. In [9], body parts were represented by combinations of joint orientation and position histograms. Separate Adaboost detectors were trained for several body parts. Localization was then obtained by optimizing the likelihood of part occurrence along with the geometric relation. However, while these techniques usually attempt to provide a general object recognition framework [8], they do not lend themselves to fast implementations. In addition, they take into account the articulated nature of the human body, which might not be so appropriate when dealing with low resolution human images such as those often encountered in surveillance videos.

The second approaches are based on applying a human detector for all possible subwindows in a given image [5, 11, 16]. For instance, in [11], a SVM classifier was learned using Haar wavelets as descriptors. In [16], an efficient detector applicable to videos was built using a cascade of Adaboost classifiers relying also on Haar wavelet descriptors but extracted from spatio-temporal differences. Recently, [1] proposed a very good detector that relied on a linear SVM classifier applied to densely sampled histograms of orientation gradient (HOG). It was extended in [2] to videos using histograms of differential optical flow features in addition to HOG. Finally, very recently, [15] proposed a method that outperformed previous methods [1]. It is based on a cascade of LogitBoost classifiers that uses covariance features as human descriptors. More precisely, subwindows of the detection windows are represented by the covariance matrix of image features, such as spatial location, intensity, gradient magnitude and orientation. The LogitBoost classifier was modified by mapping the covariance matrix features in an appropriate space to account for the fact that covariance matrices do not lie in a vector space but in a Riemannian manifold. This resulted in superior performance.

In the present paper, out goal is to detect humans in videos captured from stationary cameras. We extend the method of Tuzel et al. [15] in several ways to speed up the computation and take into account the temporal information. First, as the covariance mapping step performed for each weak classifier is slow for high dimensional input feature space, we propose to build weak classifiers based on subsets of the complete image feature space. This corresponds to using sub-matrices of the full covariance matrix and allows us to explore the covariance between features in small groups rather than altogether for each weak classifier. As the number of subsets to explore increases exponentially, we propose a tractable method to select with high probability the subset that provides the best performance in the logit-boost training stage.

Secondly, we propose to combine these sub-matrix covariance features with the image feature means computed within the same subwindow, which are available at no additional cost. The use of these means allows faster rejection and leads to similar or better results at a reduced cost.

Thirdly, in the context of videos acquired with stationary cameras, we propose to fuse image features from the spatial and temporal domains in order to take advantage of both appearance and foreground information. While in the past background subtraction results have commonly be used as a region of interest (ROI) selection process, e.g. [6] (an exception is [16]), we propose here to use them directly as features in the classifiers. This has several advantages. First, due to the cascade approach, the temporal features still play implicitly a ROI role allowing for faster processing. This will be achieved in a more informative way, by exploring the correlation between these temporal features and the spatial ones. Second, we propose to use object foreground probabilities rather than background subtraction binary masks. This is interesting as these probabilities can exhibit variations related to the human body pose (to the contrary of cast shadow for instance), as illustrated by some examples in Fig. 2. In addition, this choice alleviates the need for setting the background detection threshold; a sensitive issue in practice. When too low a threshold is used, the resulting over-detection produces less informative masks. When too high a threshold is used, there will be missed detections. Our choice should thus be more robust against variation in the contrast between humans and the background.

All together, the result is a near real-time human detector that performs accurately on challenging datasets. The rest of the paper is organized as follows. Section 2 introduces the covariance features. In Section 3 we presents a brief description of the LogitBoost classification algorithm for Riemanian manifolds. Section 4 presents our approach. Experimental results are presented in Section 5.

2 Region Covariance Descriptors

Let $\mathbf{I}$ be an input image of dimension $W \times H$. From this image we can extract at each pixel location $\mathbf{x} = (x, y)^\top$ a set of d features such as intensity, gradient, and filter responses. Accordingly, we can define a $W \times H \times d$ feature image $\mathbf{H}$.

Selected Features: We propose to use the following 8-dimensional set $\mathbf{H}(\mathbf{x})$:

$$\mathbf{H}=\left[\mathbf{x}\ |\mathbf{I}_x|\ |\mathbf{I}_y|\ \sqrt{\mathbf{I}_x^2+\mathbf{I}_y^2}\ \arctan\frac{|\mathbf{I}_y|}{|\mathbf{I}_x|}\ \mathbf{G}\ \sqrt{\mathbf{G}_x^2+\mathbf{G}_y^2}\right]^\top \quad (1)$$

where $\mathbf{I}_x$ and $\mathbf{I}_y$ are the first-order intensity derivatives, and $\arctan\frac{|\mathbf{I}_y|}{|\mathbf{I}_x|}$ represents the edge orientation. $\mathbf{G}$ denotes a foreground probability value (a real number between 0 and 1 indicating the probability that the pixel $\mathbf{x}$ belongs to the foreground), and $\mathbf{G}_x$ and $\mathbf{G}_y$ are the corresponding first-order derivatives. With respect to [15], the main difference is in using the two foreground related measures instead of second-order intensity derivatives $\mathbf{I}_{xx}$ and $\mathbf{I}_{yy}$ of the original images. In the context of human detection in videos, foreground measures should be much more informative. To extract the foreground features, we rely on the robust background subtraction technique described in [17]. Examples are shown in Fig. 2.

Covariance computation: Given a rectangular window R, we can compute the covariance matrix $\mathbf{C}_R$ of the features inside that window according to:

$$\mathbf{C}_R = \frac{1}{|R|-1}\sum_{\mathbf{x}\in R}(\mathbf{H}(\mathbf{x}) - \mathbf{m}_R)(\mathbf{H}(\mathbf{x}) - \mathbf{m}_R)^\top \quad (2)$$

where $\mathbf{m}_R$ is the mean vector in the region R, i.e. $\mathbf{m}_R = \frac{1}{|R|}\sum_{\mathbf{x}\in R}\mathbf{H}(\mathbf{x})$, and $|\cdot|$ denotes the set size operator. The covariance matrix is a very informative descriptor which encodes information about the variance of the features, their correlations with each other, and spatial layout. It can be computed efficiently using integral images [14].

Covariance normalization: The covariance features are robust towards constant illumination changes. To allow robustness against local linear variations of the illumination, we apply the following normalization. Let r be a possible subwindow inside the test window R. We first compute the covariance of the subwindow $\mathbf{C}_r$ using the integral representation. Then, all entries of the covariance $\mathbf{C}_r$ are normalized w.r.t. the standard deviations of their corresponding features inside the detection window R, which can be obtained from the diagonal terms of the covariance $\mathbf{C}_R$ [15]. The resulting covariance is denoted $\mathbf{C}'_r$.

3 LogitBoost Learning on Riemannian Space

LogitBoost algorithm: We first briefly introduce the standard LogitBoost algorithm on vector spaces [4], which is a variant of the popular Adaboost algorithm. In this section, let $\{\mathbf{x}_i, y_i)\}_{i=1...N}$ be the set of training examples, with $y_i \in \{0, 1\}$ and $\mathbf{x}_i \in \mathbb{R}^n$. The goal is to find a decision function F which divides the input space into the 2 classes. In LogitBoost, this function is defined as a sum of weak classifiers, and the probability of an example $\mathbf{x}$ being in class 1 (positive) is represented by

$$p(\mathbf{x}) = \frac{e^{F(\mathbf{x})}}{e^{F(\mathbf{x})} + e^{-F(\mathbf{x})}}, \;\; F(\mathbf{x}) = \frac{1}{2}\sum\nolimits_{l=1}^{N_L} f_l(\mathbf{x}). \qquad (3)$$

The LogitBoost algorithm iteratively learns the set of weak classifiers $\{f_l\}_{l=1\dots N_L}$ by minimizing the negative binomial log-likelihood of the training data:

$$-\sum\nolimits_{i}^{N} \left[y_i \log(p(\mathbf{x}_i)) + (1-y_i)\log(1-p(\mathbf{x}_i))\right], \qquad (4)$$

through Newton iterations. At each iteration l, this is achieved by solving a weighted least-square regression problem: $\sum_{i=1}^{N} w_i \| f_l(\mathbf{x}_i) - z_i \|^2$, where $z_i = \frac{y_i - p(\mathbf{x}_i)}{p(\mathbf{x}_i)(1-p(\mathbf{x}_i))}$ denotes the response values, and the weights are given by $w_i = p(\mathbf{x}_i)(1-p(\mathbf{x}_i))$.

LogitBoost for Riemannian manifolds: However, since covariance matrices do not lie in a vector space but in the Riemannian manifold of symmetric positive definite matrices $\mathcal{M}$, Tuzel et al. [15] proposed modifications to the original LogitBoost algorithm to specifically account for the Riemannian geometry. This was done by introducing a mapping $h : \mathcal{M} \rightarrow \mathbb{R}^n$ projecting the input covariance features into the Euclidian tangent space at a point $\boldsymbol{\mu}_l$ of the manifold $\mathcal{M}$:

$$h : \mathbf{X} \mapsto \mathbf{x} = h(\mathbf{X}) = \mathrm{vec}_{\boldsymbol{\mu}_l}(\log_{\boldsymbol{\mu}_l}(\mathbf{X})) \qquad (5)$$

where the vec and $\log$ operators are defined by $\mathrm{vec}_{\mathbf{Z}}(\mathbf{y}) = \mathit{upper}(\mathbf{Z}^{-\frac{1}{2}}\mathbf{y}\mathbf{Z}^{-\frac{1}{2}})$ with *upper* denoting the vector form of the upper triangular martix part, and $\log_{\mathbf{Z}}(\mathbf{Y}) = \mathbf{Z}^{\frac{1}{2}}\log(\mathbf{Z}^{-\frac{1}{2}}\mathbf{Y}\mathbf{Z}^{-\frac{1}{2}})\mathbf{Z}^{\frac{1}{2}}$ and $\log(\Sigma) = \mathbf{U}\log(\mathbf{D})\mathbf{U}^{\top}$ where $\Sigma = \mathbf{U}\mathbf{D}\mathbf{U}^{\top}$ is the eigenvalue decomposition of the symmetric matrix Σ, and $\log(\mathbf{D})$ is a diagonal matrix whose entries are the logarithm of the diagonal terms of $\mathbf{D}$ [12, 15]. One question that arises is: for a weak classifier f_l, how can we select the projection point $\boldsymbol{\mu}_l$? Tuzel et al. [15] proposed to use the weighted mean of all training examples, which is defined by: $\boldsymbol{\mu}_l = \arg\min_{\mathbf{Y}\in\mathcal{M}} \sum_{i=1}^{N} w_i d^2(\mathbf{X}_i, \mathbf{Y})$ where the function $d^2(\mathbf{X}, \mathbf{Y})$ measures the distance between two points $\mathbf{X}$ and $\mathbf{Y}$ in the Riemannian space $\mathcal{M}$. This minimization is achieved using a gradient descent procedure described in [12]. Since the weights are adjusted through boosting, at a given iteration l, the mean will move towards the examples which have not been well classified during previous iterations, allowing to build more accurate classifiers for these points. Ultimately, a weak classifier is defined as: $f_l(\mathbf{X}) = g_l(h(\mathbf{X}))$ where g_l can be any function from $\mathbb{R}^n \rightarrow \mathbb{R}$. In this paper, we used linear functions.

Learning with a cascade: The above method was implemented within a cascade of LogitBoost rejection classifiers [15] (details are given in the next Section). At each iteration l, a collection of weak classifiers are learned and the one that minimizes the negative binomial log-likelihood (4) is actually added as f_l to form the decision function F. The collection of classifiers is made out of all the covariance features that can be extracted from the subwindows r of the detection window R. However, to keep the computation tractable, only a subset is tested. At each boosting iteration l, we randomly select $N_w = 200$ subwindows whose size is at least 1/10 of the width and height of the detection window [15].

4 Proposed Algorithm

In this section, we describe the improvements we made to the approach as well as more technical details about the cascade training.

4.1 Using Feature Subsets

The cascade of LogitBoost classifiers is quite fast. However, at runtime, most of the computation is spent on the eigenvalue decomposition requested to compute the logarithm of a matrix in the mapping step (cf (5) and formulas that follow). The load depends on the feature dimension, as illustrated in Fig. 1(a), which shows the relative computation time of a LogitBoost classifier composed of 10 weak classifiers. One option to speed-up the process could be to decrease the overall feature size. However, this could be at the cost of performance. We propose instead to use weak classifiers relying on subsets of the full feature set. In this way, all the features are kept and the more consistent correlation between them can be exploited.

Selecting the feature subsets: Assume that we have a d-dimensional feature vector, and that we are interested in selecting subsets of size $m(< d)$. Let $\mathcal{S}_d^m = \{\mathrm{S}_{m,i}\}_{i=1\dots C_m^d}$ denote the set of all subsets of size m, where $\mathrm{S}_{m,i}$ is the i-th such m-subset, and $C_m^d = \frac{d!}{(d-m)! \times m!}$ denotes the number of un-ordered $m-$subsets. At each LogitBoost iteration, we would like to find the best subwindow-subset couple $(r^{\star}, i^{\star})$ that provides the minimum negative binomial log-likelihood, i.e.: $(r^{\star}, i^{\star}) = \arg\min_{r,i} L_r(\mathrm{S}_{m,i})$, where $L_r(\mathrm{S}_{m,i})$ denotes the negative binomial log-likelihood defined in (4) after the training of the weak classifier on subwindow r with the feature subset $\mathrm{S}_{m,i}$. Such an exhaustive search involve the training of $N_w \times C_m^d$ weak classifiers, which becomes quickly intractable when m is large (the classifier is more costly to train), and C_m^d is large.

Rather than using random selection of feature subsets, we adopted the following approach. First we fully test all the 2-subsets, whose weak classifiers can be trained very fast, and obtain the set $\{L_r(\mathrm{S}_{2,i})\}_{i=1\dots C_2^n}$ where smaller value means that the pair of features is a better choice for classification. Then, for each subset $\mathrm{S}_{m,i}$ of size $m > 2$,

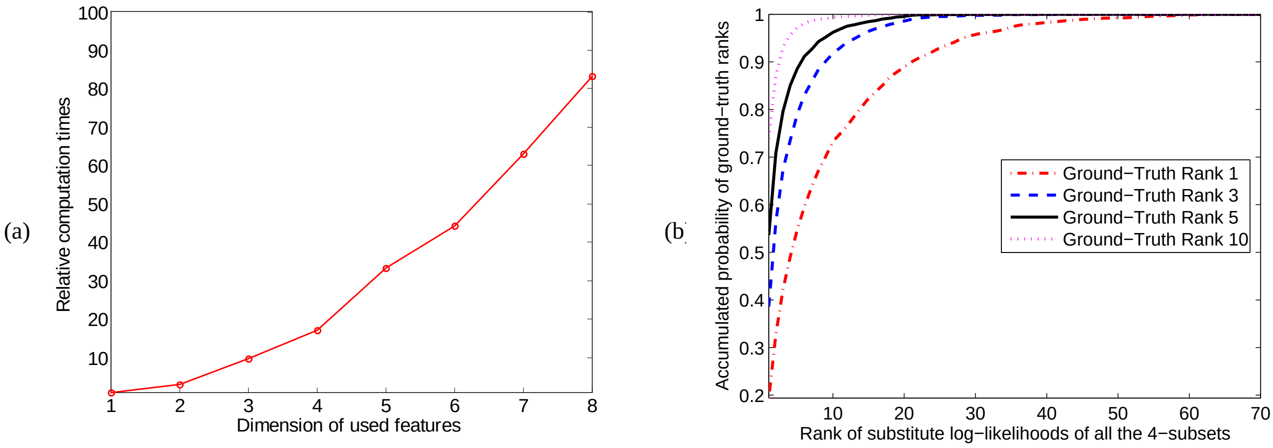

Fig. 1: (a) Relative computation time of LogitBoost classifiers, for different feature sizes. Size one is taken as reference; (b) Grouth-truth ranks of $\{L_r(\mathrm{S}_{m,i})\}$ vs. approximated ranks of $\{\tilde{L}_r(\mathrm{S}_{m,i})\}$, for $m = 3$.

we compute a *substitute value* of negative binomial log-likelihood $\tilde{L}_r(\mathrm{S}_{m,i}) = \sum_{\mathrm{S}_{2,s}\in\mathrm{S}_{m,i}} L_r(\mathrm{S}_{2,s})$ and then select the q best subsets according to these values to be actually tested. The principle that we use is that good pairs of features, which exhibit high correlation feature discrimination, should produce good feature subsets of higher dimension.

Illustration of this principle is shown in Fig. 1(b), which compares weak classifier ranks using $\tilde{L}$ vs using L when building a human detector. The curves plot the probability that within the first q values of $\tilde{L}$ (horizontal axis), we find at least one of the k best subset (k=1,3,5, or 10) according to the ground truth L. As can be seen, by selecting $q = 8$ subsets (out of 56) using $\tilde{L}$, the chances that one of them is actually one of the top 3 best are higher than 94%. Thus our approach provides a better way of selecting good m-subset features than uniform random selection, and saves a significant amount of time in training.

4.2 Using Mean Features

The covariance matrix $\mathbf{C}_r$ of a subwindow r can be efficiently computed using integral images [14]. During computation, the mean features $\mathbf{m}_r$ are also computed, and we propose to use them as additional features for training and detection. Since these features directly lie in a d-dimensional Euclidean space (i.e. $\mathbf{m}_r \in \mathbb{R}^d$), we don't need any form of mapping like in the covariance case. However, in order to be robust against illumination changes, the subwindow mean vector entries of $\mathbf{m}_r$ are normalized w.r.t. the corresponding entries of the mean vector $\mathbf{m}_R$ in the detection window R, which results in $\mathbf{m}'_r$. The weak classifiers that we propose are thus defined as: $f_l(\mathbf{X}_r) = g_l(h(\mathbf{C}'_r), \mathbf{m}'_r)$ where h is the mapping function defined in (5) that projects the normalized covariance $\mathbf{C}'_r$ features into the tangent space at the weighted-mean matrix, as explained in Section 3. In other words, we use the concatenation of the mapped covariance features with the normalized mean features in the linear function g_l to be used in the LogitBoost classifier.[1]

4.3 Training the cascade

The human detector is trained using a rejection cascade of LogitBoost classifiers framework. We used $K = 30$ cascade levels. The number ${N_L}^k$ of weak classifiers composing the k-th cascade level is selected by optimizing the LogitBoost classifier to correctly detect at least 99.8% of the positive examples, while rejecting at least 30% of the negative examples. In addition, we enforce a margin constraint between the positive examples and the decision boundary. Let $p_k(\mathbf{x})$ be the probability of an example $\mathbf{x}$ being positive at the cascade level k, as defined in (3). Let $\mathbf{x}_p$ be the positive example that has the $(0.998N_p)$-th largest probability among all the positive examples and $\mathbf{x}_n$ be the negative example that has the $(0.3N_n)$-th smallest probability among all the negative examples where N_p and N_n are the numbers of positive and negative examples used for training at the cascade level k. Weak classifiers are added to the cascade level k until $p_k(\mathbf{x}_p) - p_k(\mathbf{x}_n) > th_b$ where we set $th_b = 0.2$. At test time, an example $\mathbf{x}$ will be rejected if $p_k(\mathbf{x}) \leq p_k(\mathbf{x}_n)$. In order to train a cascade level k, we used $N_p = 4000$ and $N_n = 8000$ positive and negative examples. These examples were obtained by applying the detector up to the $k-1^{th}$ level to a set of around 10000 positive examples and those with the least probability of being positive are kept for training. In a similar way, the negative

[1] Note that when a feature subset is used for the covariance, only the means of that subset are used in the weak classifier.

examples were selected as the false positive examples of the $k-1^{th}$ detector applied to training data.

5 Experimental Results

5.1 Training and Testing Datasets

We collected a total of 15 video sequences captured from stationary cameras. There are 10 indoor and 5 outdoor video sequences selected from the shopping center CAVIAR data, from the PETS data, and from several metro station cameras. A total of around 10000 positive examples were extracted from these 15 video sequences. Some typical examples are shown in Figure 2. Note that in these examples, there exist the large variations of appearances, pose, camera view-points, the presence of luggage or trolleys, occlusions, and the variability in the foreground extraction. Negative examples were obtained by: (i) collecting 1000 still images without people and coupling them with inconsistent foreground detection results; (ii) cropping about 1000 regions from the collected video data which don't contain humans (iii) bootstrapping, i.e. by collecting more negative 'look like' people by applying the current detector on training data without people.

For testing, we set apart 523 images from video clips belonging to 10 of the above sequences and not used for training, and added data from 2 new video sequences. A total of 1927 humans was annotated, comprising 327 humans with significant partial occlusion and around 200 humans with a resolution of less than 700 pixels.

5.2 Evaluation Methodology

The detectors were evaluated on the testing data by applying them on image subwindows with different locations, scales, and aspect ratios, according to the following: the width ranged from 25 to 100 pixels; the aspect ratio (height divided by width) ranged from 1.8 to 3.0. The positive detections were then filtered out by keeping local maxima of these detection outputs according to the probabilities defined in (3) as the final detected persons. Two types of performance measure curves were used. In both cases, curves were generated by adding cascade levels one by one.

Detection Error Tradeoff (DET) curves : In the recent literature [1, 10, 15], DET curves have been used to quantify the raw binary classifier performance at the window level. DET curves measure the proportion of true detections against the proportion of false positives. They plot the miss rate, $\frac{\texttt{\#FalseNeg}}{\texttt{\#TruePos}+\texttt{\#FalseNeg}}$, versus false positives (here the False Positives Per tested Window or FPPW) on a log-log scale. To produce this curve, the 1927 positive examples of the testing data were used to evaluate the miss-rate, while the FPPW was obtained by testing all searching windows of the testing data which do not overlap or overlap by less than 50% with any positive example. The overlap is measured as the F-measure $F_{area} = \frac{2\rho\pi}{\rho+\pi}$, where $\rho = \frac{|GT \cap C|}{|GT|}$ and $\pi = \frac{|GT \cap C|}{|C|}$ are the area recall and precision, with GT denoting the ground truth region, and C the tested window.

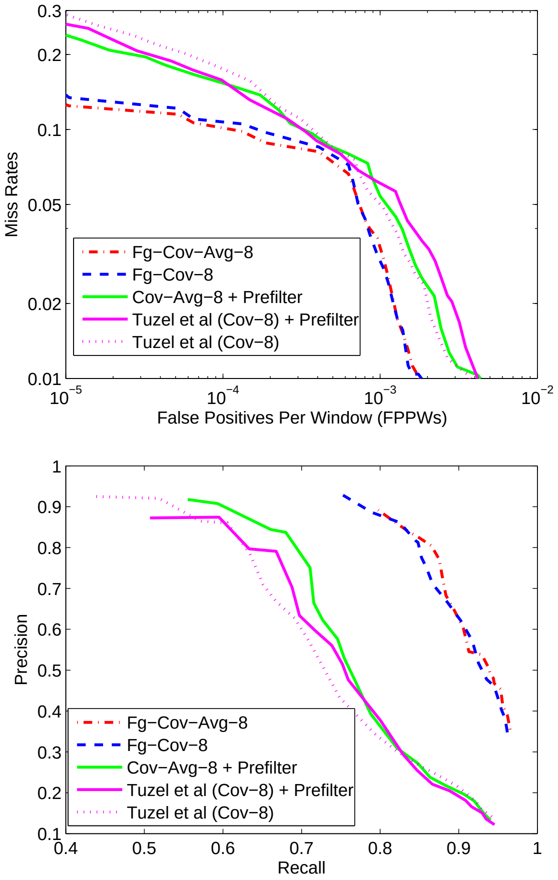

Fig. 3: The performance of different approaches for our method with 8-dimensional features.

Recall-Precision (RP) curves: RP curves are more appropriate to measure the accuracy of the object detection and localisation from a user point of view [3, 13]. RP curves integrates the post-processing steps (e.g. how to combine several raw detector positive output into one or several detected humans). Recall and precision are defined as $\frac{\texttt{\#TruePos}}{\texttt{\#TruePos}+\texttt{\#FalseNeg}}$ and $\frac{\texttt{\#TruePos}}{\texttt{\#TruePos}+\texttt{\#FalsePos}}$, respectively. A detected output is said to match the ground truth if their F_{area} measure is above 0.5. Only one-to-one matches are allowed between detected and ground truth regions.

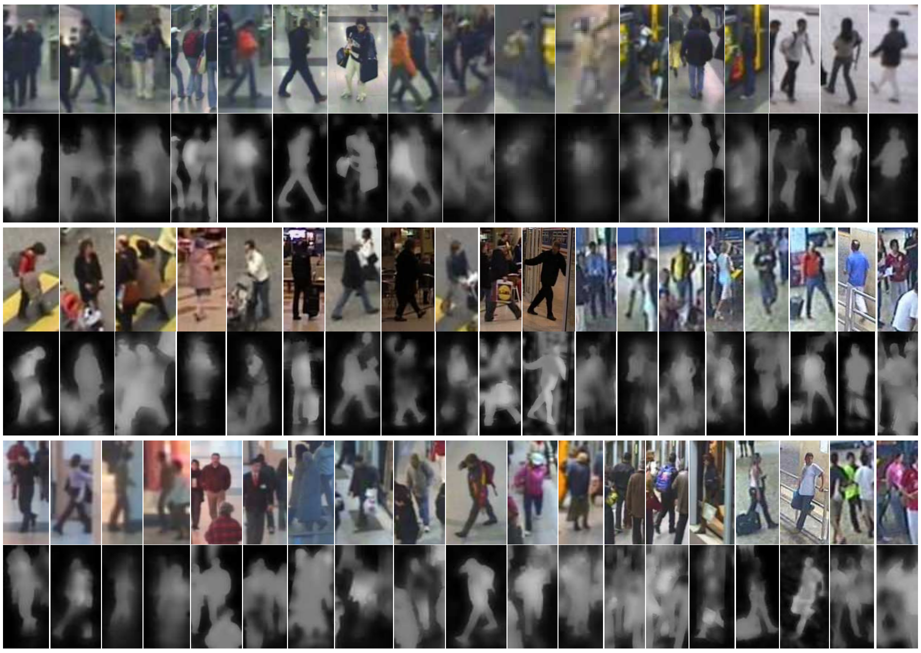

Fig. 2: Positive examples with corresponding foreground probability maps (light - high probability, dark - low probability).

5.3 Results

We consider the method of Tuzel et al [15] as our baseline. Three main improvements to this method were made to handle video data: integration of foreground probability features, selection of feature subsets, and use of mean (average) features in addition to covariance. We trained several detectors with or without the proposed improvements to evaluate their impact on the detection performance. These detectors are named accordingly. For example, the detector *Fg-Cov-Avg-8* uses the 8-dimensional covariance and mean features defined in (1) which integrate intensity and foreground information. When foreground features are not used, we used the 8-dimensional features defined in [15]. In addition, to allow fair comparing, a prefilter is also applied with [15] to only test windows which contain enough (20%) foreground pixels.

In the first experiment, whose results are shown in Fig. 3, we trained four detectors with/without the use of foreground information and mean features. We can observe that the integration of the foreground information in the learning process rather than as a preprocessing step provides much better detection performance. For instance, the RP curve shows that for a recall of 0.9, only around 1 out of 5 detections is correct with [15] even with the prefilter, while with the foreground features, around 3 out of 5 detections are correct. Besides, we can see that the use of the mean features improves the results almost systematically, but usually not significantly.

In the second experiment, we trained three new detectors relying on 2, 3 and 4-subset features (*Fg-Cov-Avg-2* to *Fg-Cov-Avg-4*, respectively). In addition, we trained a combined detector based on 2-subset features in the first 15 cascade levels, 3-subset features in the subsequent 10 levels, and 4-subset features in the final 5 levels (*Fg-Cov-Avg-[2,3,4]*). Fig. 4(a) shows the RP curve that we obtain. We observe that the use of subset features results in similar detection performance than the use of the full 8-dimensional feature set, with *Fg-Cov-Avg-[2,3,4]* providing the best results overall. However, the main interest of our approach is the computation time. Fig. 4(b) shows the average numbers of tested windows per second that a detector processes when applied on the test data. The same computer was used in all the cases. The first observation is that while the mean features only slightly improve the performance, they offer a speed gain of nearly 30% (e.g. compare Tuzel et al [15] with *Cov-Avg-8*). Secondly, as could be expected, the use of the foreground features also helps in increasing the speed by rejecting false hypothesis more quickly. Finally, the main computational gain is obtained by using feature subsets. For instance, the detector *Fg-Cov-Avg-2* runs around 13 times faster than *Fg-Cov-Avg-8* (and more than 20 times faster than [15]). The combined detector *Fg-Cov-Avg-[2,3,4]* achieves a similar speed while slightly improving the performance (see Fig. 4). We can apply these two detectors to videos of size 384x288 (e.g. CAVIAR data) and process 5-20 frames/sec when including the adaptive background subtraction process.

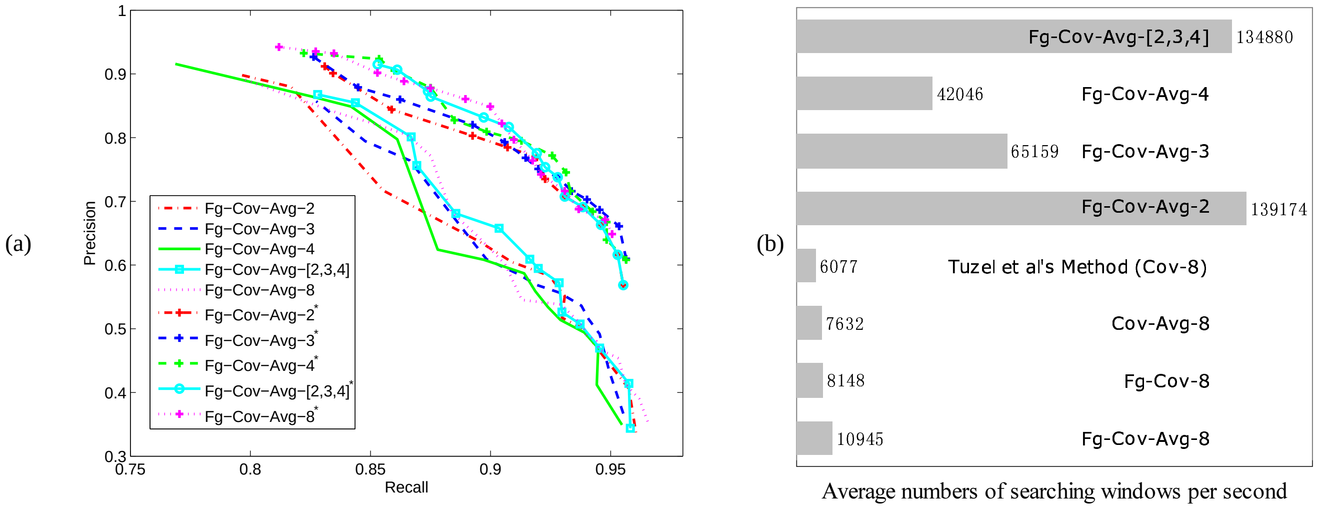

Fig. 4: (a) Performance of different approaches with/without ground-plane geometrical constraint; (b) Average numbers of tested windows (per second)

Finally, to further speed up the process and improve detection performance, we propose to exploit rough ground plane geometrical constraints to limit the human heights from 150cm to 220cm. When applying the detectors again on the test data using this constraint, we obtain a gain due to the removal of some of the false positives (see Fig. 4(a)).

Fig. 5 shows some detection examples obtained with the *Fg-Cov-Avg*-2* detector with geometrical constraints. Green dots show the positive window detection, while red bounding boxes with red center dots are the final detected results after the local maximum post-processing step. Despite the large variability of appearance, pose and view points, the partial occlusions, and the overall small people size, there are only few false positive and negative detections. The main errors come for strong specular reflections and cast shadow (e.g. in CAVIAR, these reflections sometimes almost produce upside-down foreground detection), bad foreground results produced by moving objects (moving escalator in the Metro scene), or occlusions by other persons or objects (e.g. bicycles). In addition, as the proposed method focuses on full human body detection, some humans who are only partially visible are not detected. Video examples are provided as accompanying material.

6 Conclusions

A fast method to detect humans from videos captured by stationary cameras was presented. We made three improvements to the method of Tuzel et al [15] which outperformed existing methods by utilizing covariance matrices as object descriptors. We proposed to use subsets of the complete image feature space to build the weak classifiers of the LogitBoost classifiers. In addition, we exploited the means of the image features with the covariance features, producing similar or better results at a reduced detection computation cost. Finally, we propose to integrate foreground information with appearance information to extract image features for detecting humans in videos. These extensions result in a near real-time human detection that performs accurately on challenging datasets.

7 Acknowledgement

This work was supported by the European Union 6th FP Information Society Technologies CARETAKER project (Content Analysis and Retrieval Technologies Applied to Knowledge Extraction of Massive Recordings, FP6-027231).

References

[1] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893, June 2005.

[2] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *Europe Conf. Comp. Vision (ECCV)*, volume II, pages 428–441, 2006.

[3] Navneet Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble, 2006.

[4] J. Friedman, T. Hastie, and R. Tibshira. Additive logistic regression: a statistical view of boosting. *Ann. Statist.*, 23(2):337C407, 2000.

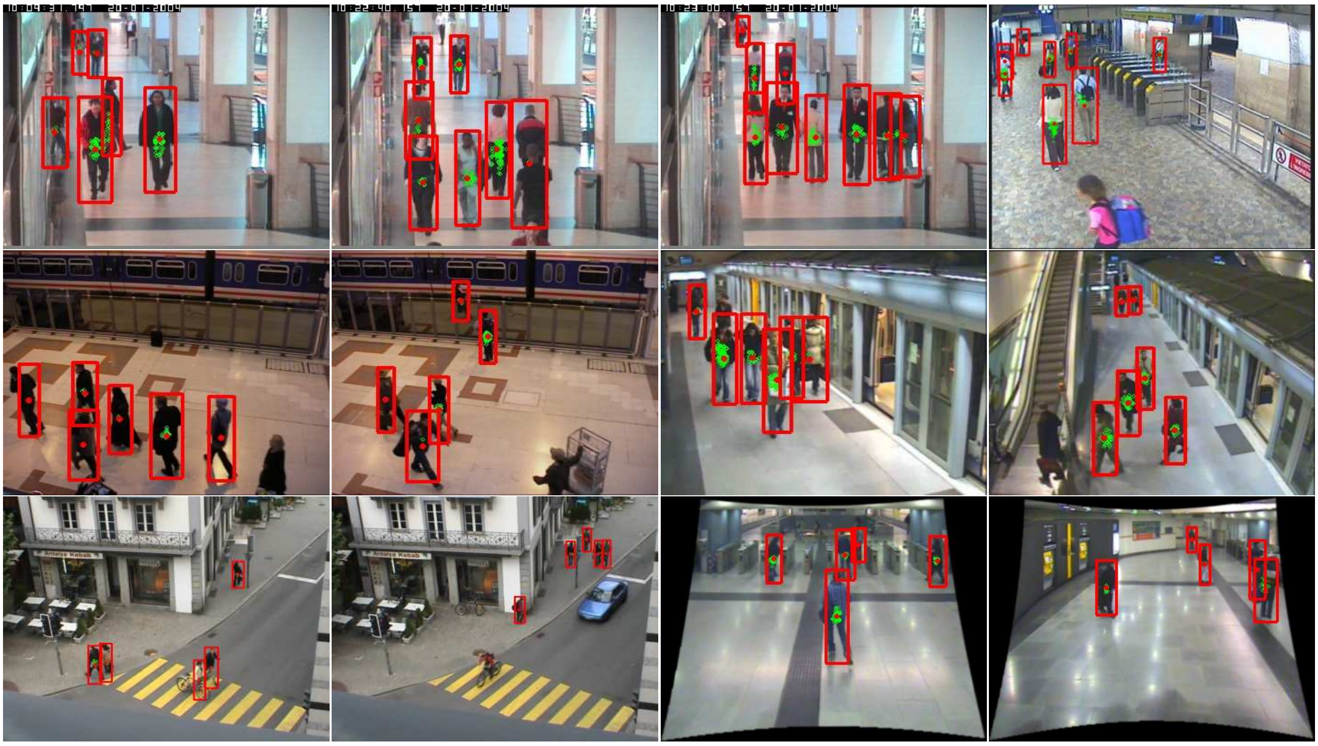

Fig. 5: Detection examples. Green dots show all the detection results. Red dots are final detection results via post-processing and the bounding boxes are average detection window sizes.

[5] D. Gavrila and V. Philomin. Real-time object detection for “smart“ vehicles. In *IEEE CVPR*, pages 87–93, 1999.

[6] Mohamed Hussein, Wael Abd-Almageed, Yang Ran, and Larry Davis. A real-time system for human detection, tracking and verification in uncontrolled camera motion environment. In *IEEE International Conference on Computer Vision Systems*, 2006.

[7] S. Ioffe and D.A. Forsyth. Probabilistic methods for finding people. *Int. Journal of Comp. Vision*, 43(1):46–68, 2001.

[8] C. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *IEEE CVPR, New York*, volume 1, pages 26–36, 2006.

[9] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Europe Conf. Comp. Vision (ECCV)*, volume I, pages 69–81, 2004.

[10] S. Munder and D. M. Gavrila. An experimental study on pedestrian classification. *IEEE Trans. Pattern Anal. Machine Intell.*, 28(11):1863–1868, 2006.

[11] P. Papageorgiou and T. Poggio. A trainable system for object detection. *Int. J. of Computer Vision*, 38(1):15–33, 2000.

[12] Xavier Pennec, Pierre Fillard, and Nicholas Ayache. A riemannian framework for tensor computing. *Int. Journal of Comp. Vision*, 66(1):41–66, 2006.

[13] Edgar Seemann, Mario Fritz, and Bernt Schiele. Towards robust pedestrian detection in crowded image sequences. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 2007.

[14] O. Tuzel, F.M. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Europe Conf. Comp. Vision (ECCV)*, volume II, pages 589–600, 2006.

[15] Oncel Tuzel, Fatih Porikli, and Peter Meer. Human detection via classification on riemannian manifolds. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, 2007.

[16] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *Int. Journal of Comp. Vision*, 63(2):153–161, 2005.

[17] Jian Yao and Jean-Marc Odobez. Multi-layer background subtraction based on color and texture. In *Computer Vision and Pattern Recognition, 2007. CVPR ’07. IEEE Conference on*, pages 1–8, 17-22 June 2007.