This paper appeared in CVIU, Volume 115, Number 10, pp 1414-1426, October 2011

The research report [22] provides more detailed explanations and results analysis.

Fast Human Detection from Joint Appearance and Foreground Feature Subset Covariances

Jian Yao[a], Jean-Marc Odobez[a]

[a]*Idiap Research Institute Centre du Parc, Rue Marconi 19, CH-1920 Martigny, Switzerland*

Abstract

We present a fast method to detect humans from stationary surveillance videos. It is based on a cascade of LogitBoost classifiers which use covariance matrices as object descriptors. We have made several contributions. First, our method learns the correlation between appearance and foreground features and show that the human shape information contained in foreground observations can dramatically improve performance when used jointly with appearance cues. This contrasts with traditional approaches that exploit background subtraction as an attentive filter, by applying still image detectors only on foreground regions. As a second contribution, we show that using the covariance matrices of feature subsets rather than of the full set in boosting provides similar or better performance while significantly reducing the computation load. The last contribution is a simple image rectification scheme that removes the slant of people in images when dealing with wide angle cameras, allowing for the appropriate use of integral images. Extensive experiments on a large video set show that our approach performs much better than the attentive filter paradigm while processing 5 to 20 frames/sec. The efficiency of our subset approach with state-of-the-art results is also demonstrated on the INRIA human (static image) database.

Key words: Human detection, surveillance, learning, covariance matrices, information fusion, image rectification, real-time.

1. Introduction

Detecting and localizing humans in videos is an important task in computer vision. On the one hand, it is often a first step upon which more complex activity or behavior analysis is performed, threfore it has many applications in surveillance or monitoring of smart spaces such as offices. Indeed, improving human modeling and detection is crucial for tracking algorithms, especially when scenes become crowded. On the other hand, human detection can also be used on its own, without being involved in a more evolved framework like tracking. For instance, in [2], a human detector was continuously applied to count the number of people in different places of a metro station in order to provide usage statistics to metro operators or to detect abnormal situations (i.e. counts which differ from standard values observed at a given time and a given day of the week).

In this paper, we address the detection of humans in videos recorded by stationary cameras. This task faces important challenges given the large variability of appearance and pose generated by variations in clothing, illumination, body articulations or camera view points. In addition, as illustrated in Fig. 4, the image resolution of humans is usually small and humans can appear slanted due to the use of wide field-of-view (FOV) cameras. Also, humans often occlude each other, and the colors of their clothes are often similar (and similar to the background as well).

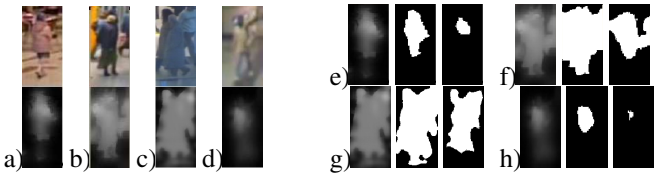

Fig. 1: (a)-(d) Appearance and foreground information of four samples. In (a) the person is camouflaged by the background, while the foreground is more informative; in (b)-(d), the foreground is noisier but appearance remains discriminative (see Fig. 2 for more examples). (e)-(h) Three foreground information representations for each of the (a)-(d) case. From left to right: the foreground probability map, this map thresholded with T_1, or with $T_2 > T_1$. Notice how the probability maps contains more accurate and detailed information than their thresholded counterparts. Thresholding removes foreground information, even with a low threshold (h), adds artificial edges when there is none, or conversely, removes informative gradients between body parts present in the probability map (f).

To take advantage of the temporal dimension of videos, most approaches use background subtraction as a way of rejecting detections containing not enough foreground pixels, implicitly assuming that such detections correspond to false alarms. However, foreground images carry much more precise and discriminative information than that. This is illustrated in Figs. 1 and 2, and demonstrated by the research on human pose estimation from clean silhouettes extracted through background sub-

Email addresses: jianyao75@gmail.com (Jian Yao), odobez@idiap.ch (Jean-Marc Odobez)

traction. Thus, foreground images could be used not only to discard false detections but also to help the detection process itself. Even when people occlude each other or when there is a lack of contrast between a person and the background, some foreground regions remain characteristic of the human shape (head, legs), allowing the correct detection of people when combined with shape and texture cues from the input image. Conversely, in cluttered regions (e.g. having a highly textured background), appearance cues might not be sufficient to assess the presence of a person. In this case, foreground information, which is more robust to camouflage, is helpful to complement the appearance cues and make a decision.

The first and main contribution of the paper is a novel human detection algorithm that jointly learns from appearance and foreground features, which we claim is a major way to obtain substantial detection improvements compared to the sole use of appearance features. Noticing that a high degree of correlation between the appearance and foreground shapes at different places of a human template (head, body sides and legs) is a reliable indicator for human detection, we decided to exploit Tuzel *et al.*'s [18] approach which uses covariance matrices as object descriptors within a cascade of LogitBoost classifiers.

The paper makes also several contributions by extending the work of Tuzel *et al.* in several important ways. The first one, related to the main contribution above, is the actual fusion between appearance and foreground cues, which is performed by using features from the still and foreground images. This has several advantages in addition to performance improvement. First, due to the cascade approach, the foreground features play a Region Of Interest (ROI) focusing role allowing for faster processing. As the decision is based on correlation analysis between both still image and foreground features, this ROI role is achieved in a more informative way than in traditional approaches which only look for the presence of enough foreground pixels. The second advantage is the use of *continuous* foreground probabilities rather than background subtraction *binary* masks. This choice alleviates the need for setting a foreground detection threshold, which is a sensitive issue in practice, as illustrated in Fig. 1. The algorithm is thus more robust against variations in contrast between humans and background.

The use of feature subsets constitutes the second paper contribution as explained below. The importance of mapping the covariance matrices in an appropriate space to account for the fact that they lie in a Riemannian manifold was demonstrated in [18]. However, this mapping, which is performed for each weak classifier at run time, is slow when the covariance matrix dimension is high. Also, as there might not always exist consistent relations between the covariance coefficients of *all features* in the training data, the learned weak classifiers may have poor generalization performance at test time. To address these two issues we propose to exploit the covariance between feature subsets (hence resulting in lower dimensional matrices) to build the weak classifiers, rather than between all features, as was systematically done in [18]. Embedded in the LogitBoost framework, subsets with the most consistent and discriminant covariances are selected. This results in a more robust and much faster detection algorithm.

In this framework, we also investigated the use of image features' means as additional input for the weak classifiers. Intuitively, it is useful at describing the presence of strong edges or foreground information at different positions of the template. Experiments showed that such features provide similar performance at a reduced processing cost.

The third and final contribution of the paper is not related to Tuzel *et al.*'s work. It is an image rectification step allowing to reduce people's geometric appearance variability in images due to the use of large FOV cameras. More precisely, in these cases, people often appear slanted in the border of an image even after the removal of radial distortions, as illustrated in Fig. 4. This is a problem for human detectors which often consist in applying a classifier on rectangular regions, or in other tasks (e.g. tracking) when integral images are used to efficiently extract features over boxes. To handle this slant issue, we propose a method that maps the 3D vertical lines into 2D vertical image lines, as illustrated in Fig. 4.

Experiments were conducted on large publicly available video databases to assess the different algorithm components and demonstrate the validity of our approach. Additional experiments on the INRIA still image database showed that the use of feature subsets greatly reduced the computational speed while providing better detection results.

The rest of the paper is organized as follows. Section 2 introduces related work. Section 3 introduces the covariance features. In Section 4 we present a brief description of the LogitBoost classification algorithm for Riemannian manifolds. In Section 5 we introduce the main novelties of our approach. Technical details about the training and detection are given in Section 6. Experimental results are presented in Section 7, while Section 8 concludes the paper.

2. Related Work

In the following, we briefly survey human detection techniques that apply to still images, and then review more specifically the works on human detection in videos.

To detect human in still images, an approach consists in modeling the human by body parts whose locations are constrained by a geometric model [12, 10]. As an example, Leibe *et al.* [10], proposed a probabilistic human detector that combines bottom-up evidence from local features with a top-down segmentation and verification step. However, these methods usually do not lend themselves to fast implementations, and their modeling of the articulated nature of the human body might be too detailed when dealing with low resolution images. A more appropriate approach in these conditions consists in applying a fixed-template detector at all possible subwindows of an image. Methods differ by the types of input features and the training approaches [8, 3, 18]. For instance, Dalal and Triggs [3] proposed a detector relying on a linear SVM classifier applied to densely sampled histograms of orientation gradient (HOG), an approach that was sped up using the cascade and boosting framework [1]. Very recently, Tuzel *et al.* [18] proposed a cascade of LogitBoost classifiers using covariance as object descriptors which outperformed previous approaches. These techniques proved to

be robust but were mainly applied to images with enough resolution. Their performance on surveillance data or the use of foreground information was not investigated.

Few works have actually investigated human detection from videos. Optical flow has been the main cue, for instance to first extract window candidates before applying a still image human detector [6]. In [16], an SVM was trained on optical flow patterns to create a human classifier. Dalal and Triggs [4] presented a more robust approach by extending their still image HOG detector to videos using histograms of differential optical flow features. While these techniques do not assume a static camera, they require good quality optical flow which is usually expensive to compute, and partial occlusion is often an uninvestigated issue.

In the surveillance domain, most previous methods for human detection rely on motion. Temporal changes between successive images or between an image and a model of the learned background are first detected. Then moving pixels are grouped to form blobs [20, 6], which are further classified into human or non-human entities using blob shape features if necessary [24]. This type of approaches works fine for isolated people, in low density situations. However, in many cases, such an assumption does not hold. To address this issue, techniques have been proposed to segment blobs into different persons. In [23], a Bayesian segmentation of foreground blob images by optimizing the layered configuration of people using a data driven MCMC approach is conducted. Other authors applied a static human detector [6, 24, 9] on extracted foreground regions, thus following a common trend of using background subtraction results as a ROI selection process. However, this results in a suboptimal exploitation of the dynamic information and of its correlation with the appearance component of the data, as we show in this paper.

Finally, the work of Viola *et al.* [19] is the only one we found that exploits spatio-temporal features for human detection. It is based on Adaboost classifiers relying on Haar wavelet descriptors extracted from spatio-temporal differences. While scale invariance is obtained by using pyramids, the method is not invariant to the temporal scale (e.g. resulting from processing one frame out of two). In addition, the Haar features are somewhat crude and recent works have shown that better shape features can be exploited (HOG features [3] or covariances [18]).

3. Region Covariance Descriptors

Let $\mathbf{I}$ be an input image of dimension $W \times H$, from which we can define a $W \times H \times d$ feature image by extracting at each location $\mathbf{p} = (x, y)$ a set of d features expected to characterize well a person appearance. To detect persons in still images, Tuzel et al. [18] proposed to use the following $d = 8$ feature set $\mathbf{H}(\mathbf{p}) =$

$$\left[\mathbf{p}\ |\mathbf{I}_x(\mathbf{p})|\ \left|\mathbf{I}_y(\mathbf{p})\right|\ \sqrt{\mathbf{I}_x^2(\mathbf{p}) + \mathbf{I}_y^2(\mathbf{p})}\ \text{atan}\frac{\left|\mathbf{I}_y(\mathbf{p})\right|}{|\mathbf{I}_x(\mathbf{p})|}\ |\mathbf{I}_{xx}(\mathbf{p})|\ \left|\mathbf{I}_{yy}(\mathbf{p})\right|\right]^\top \tag{1}$$

where $\mathbf{I}_x$, $\mathbf{I}_y$, $\mathbf{I}_{xx}$ and $\mathbf{I}_{yy}$ denote the first and second-order intensity derivatives, and $\text{atan}\frac{|\mathbf{I}_y(\mathbf{p})|}{|\mathbf{I}_x(\mathbf{p})|}$ represents the orientation of the gradient at the pixel position $\mathbf{p}$.

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

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

where $\mathbf{m}_R$ is the mean vector in the region R, i.e. $\mathbf{m}_R = \frac{1}{|R|} \sum_{\mathbf{p} \in R} \mathbf{H}(\mathbf{p})$, and $|\cdot|$ denotes the set size operator. The covariance matrix is a very informative descriptor. In addition to the features' variance and correlation, it encodes the spatial layout of the features inside the window since the correlation with the position $\mathbf{p}$ is also computed. In practice, exploiting the fact that covariance coefficients can be expressed in terms of first and second order moments, integral images are used to gain computation efficiency [17]. More precisely, for a feature vector of dimension d, the number of integral images to be computed is $\frac{d \times (d+1)}{2}$ for the correlation coefficient (taking into account the symmetry) and d for the first order moments.

Covariance normalization: Since we rely on image derivative features and since the covariance operator is invariant to mean variations of the features, the covariance entries are quite robust to constant illumination changes. To allow further robustness against linear feature variations within a detection window, we normalize the features as follows [18]. Let R represent the detection window in which we test the presence of a person, and r a subwindow inside R where we want to extract covariance features as input to a weak classifier (see Fig.6 for illustration). We first compute the covariance of the detection window $\mathbf{C}_R$ and the subwindow $\mathbf{C}_r$. Then, all entries of $\mathbf{C}_r$ are normalized w.r.t. the standard deviations of their corresponding features inside the detection window R:

$$\mathbf{C}'_r(i, j) = \frac{\mathbf{C}_r(i, j)}{\sqrt{\mathbf{C}_R(i, i)\mathbf{C}_R(j, j)}} \tag{3}$$

where $\mathbf{C}'_r$ denotes the resulting normalized covariance. This is equivalent to (but faster than) first performing a z-normalization of the features in the detection window R (i.e. transforming the features to have zero mean and unit variance) and computing the covariance of the resulting features in the subwindow r.

4. LogitBoost Learning on Riemannian Space

We use LogitBoost classifiers based on covariance features for human detection. In the following, we first briefly introduce the LogitBoost algorithm on vector spaces [7]. Then we describe the modifications proposed in [18] to account for the fact that covariance matrices do not lie in the Euclidian space.

4.1. The LogitBoost Algorithm

In this section, let $\{\mathbf{x}_i, b_i)\}_{i=1...N}$ be the set of training examples, with $b_i \in \{0, 1\}$ and $\mathbf{x}_i \in \mathbb{R}^n$. The goal is to find a decision

function F dividing the input space into two classes. In LogitBoost, F is defined as a sum of weak classifiers $\{f_l\}_{l=1...N_L}$, and the probability of an example $\mathbf{x}$ being in class 1 (positive) is represented by

$$p(\mathbf{x}) = \frac{e^{F(\mathbf{x})}}{e^{F(\mathbf{x})} + e^{-F(\mathbf{x})}}, \quad F(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{N_L} f_l(\mathbf{x}). \tag{4}$$

The LogitBoost algorithm iteratively learns the set of weak classifiers by minimizing the negative binomial log-likelihood of the training data:

$$-\sum_i^N \left[b_i \log(p(\mathbf{x}_i)) + (1-b_i)\log\left(1-p(\mathbf{x}_i)\right)\right], \tag{5}$$

through Newton iterations [7]. At each iteration l, this is achieved by solving a weighted least-square regression problem:

$$\sum_{i=1}^{N} w_i \|f_l(\mathbf{x}_i) - z_i\|^2, \tag{6}$$

where $z_i = \frac{b_i - p(\mathbf{x}_i)}{p(\mathbf{x}_i)(1-p(\mathbf{x}_i))}$ are the response values, and the weights are given by:

$$w_i = p(\mathbf{x}_i)(1-p(\mathbf{x}_i)). \tag{7}$$

These weights and responses are computed using the weak classifiers learned up to iteration $l-1$. As can be noticed, the weights are close to 0 for data points whose probabilities are close to 0 or 1, and maximum for points with $p(\mathbf{x}) = 0.5$ i.e. for points which are not yet well classified into one category.

4.2. LogitBoost for Riemannian Manifolds

One could use the covariance features directly as input to the LogitBoost algorithm. This implicitly assumes covariance matrices to be elements of the Euclidian space $\mathbb{R}^n$. However, covariance matrices are more specific and lie in the Riemannian manifold $\mathcal{M}$ of symmetric positive definite matrices. Since the canonical Euclidian distance of $\mathbb{R}^n$ may not reflect well the actual distance between matrices in this manifold, Tuzel et al [18] introduced a mapping h from $\mathcal{M}$ into a vector space where the canonical Euclidian distance reflects the manifold geodesic distance.

More specifically, the mapping $h : \mathcal{M} \to \mathbb{R}^n$ is defined as the transformation that maps a covariance matrix into the Euclidian tangent space (denoted by $\mathcal{T}_{\boldsymbol{\mu}_l}$) at a point $\boldsymbol{\mu}_l$ of the manifold $\mathcal{M}$. More formally [14]:

$$h : \mathbf{X} \mapsto \mathbf{x} = h(\mathbf{X}) = \mathrm{vec}_{\boldsymbol{\mu}_l}\left(\log_{\boldsymbol{\mu}_l}(\mathbf{X})\right). \tag{8}$$

where the vec and log operators are defined matrix-wise 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 matrix part, and

$$\log_{\mathbf{Z}}(\mathbf{Y}) = \mathbf{Z}^{\frac{1}{2}}\log(\mathbf{Z}^{-\frac{1}{2}}\mathbf{Y}\mathbf{Z}^{-\frac{1}{2}})\mathbf{Z}^{\frac{1}{2}}. \tag{9}$$

The logarithm of a matrix Σ, $\log(\Sigma)$, is defined as

$$\log(\Sigma) = \mathbf{U}\log(\mathbf{D})\mathbf{U}^\top \text{ where } \Sigma = \mathbf{U}\mathbf{D}\mathbf{U}^\top \tag{10}$$

Fig. 3: Human detection issues. a) Foreground regions may contain other objects (e.g. parts of a metro car). b) Specular reflection and cast shadow generate background false alarms. People might be partially visible or split into mutiple blobs, and a given blob may contain several people. Also, there might be ghosts when people are integrated in the background model and then leave the scene.

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 $\mathbf{D}$ [14, 18].

Qualitatively, the manifold geodesic distance is well represented by the canonical Euclidian distance after the mapping only in the neighborhood of $\boldsymbol{\mu}_l$. The selection of this point is thus important. Intuitively, $\boldsymbol{\mu}_l$ should be as close as possible to the data points to classify, and one natural way is to select it as the weighted mean (in the Riemannian sense) of the training examples $\mathbf{X}_i$:

$$\boldsymbol{\mu} = \arg\min_{\mathbf{Y}\in\mathcal{M}} \sum_{i=1}^{N} w_i d^2(\mathbf{X}_i, \mathbf{Y}) \tag{11}$$

where $d(\mathbf{X}, \mathbf{Y})$ denotes the geodesic distance in $\mathcal{M}$ between $\mathbf{X}$ and $\mathbf{Y}$. Since the weights are adjusted through boosting (see Eq. (7)), at iteration l this choice will make the classification focus on the current decision boundary, by moving the mean towards the examples which have not been well classified yet (in practice, only the positive samples are used in Eq. (11) [18]). The minimization of Eq. (11) can be conducted using an iterative procedure [14].

In summary, a weak classifier is defined as:

$$f_l(\mathbf{X}) = g_l(h(\mathbf{X})) = g_l\left(\mathrm{vec}_{\boldsymbol{\mu}_l}\left(\log_{\boldsymbol{\mu}_l}(\mathbf{X})\right)\right) \tag{12}$$

where g_l can be any function from $\mathbb{R}^n \to \mathbb{R}$. In this paper, we used linear functions. Thus, at a given LogitBoost iteration, both the weighted mean $\boldsymbol{\mu}_l$ and the linear coefficients a_l of the regressor g_l will be learned.

5. Joint Appearance and Foreground Feature Subset Covariance

In this section, we introduce the main novelties of our approach to perform human detection in videos and increase the speed and performance of the detector w.r.t. the method described in the previous Section.

5.1. Slant Removal Preprocessing

In surveillance videos, due to the use of wide angle cameras, standing people may appear with different slants in the

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

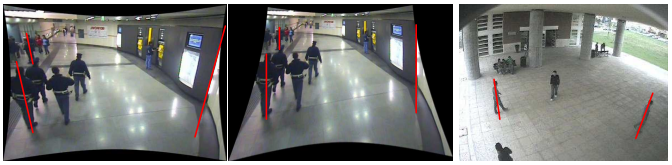

Fig. 4: Vertical vanishing point mapping. *Left:* after distorsion removal and before the mapping. We can observe people slant. *Central:* after the mapping to infinity. *Right:* another example (see mapped image in Fig. 13).

image depending on their position in the image, as illustrated in Fig. 4. This introduces variability in the feature extraction process when using rectangular regions. To handle this issue, we propose to use an appropriate projective transformation $\mathbf{K}_\perp$ of the image plane in order to map its vertical finite vanishing point to a point at infinity. As a result, the 3D vertical direction of persons standing on the ground plane will always map to 2D vertical lines in the new image, as shown in Fig. 4. This transformation helps in extracting more accurate observations and obtaining better detection results while keeping the computation efficiency of integral images.[1]

The computation of the homography $\mathbf{K}_\perp$ is constrained by the following points. It has to map the image vertical vanishing point $\mathbf{v}_\perp = (x_\perp, y_\perp, 1)^\top$ to a vanishing point at infinity $(0, y_\infty, 0)^\top$ where y_∞ can be any non-zero value. In addition, to avoid severe projective distortions of the image, we enforce that the transformation $\mathbf{K}_\perp$ acts as much as possible as a rigid transformation in the neighborhood of a given selected point $\mathbf{p}_0$ of the image. That is, the first order approximation of the transform in the neighborhood of $\mathbf{p}_0$ should be a rotation rather than a general affine transform. An appropriate choice of $\mathbf{p}_0$ to enforce such a constraint can be the image center. Technical details for computing $\mathbf{K}_\perp$ are given in [22].

5.2. Integrating Foreground Information

To detect persons in videos captured from stationary cameras, we propose to exploit the results of background substraction as additional foreground features in the detector. This is done by defining the feature vector $\mathbf{H}'(\mathbf{p})$ at a given point $\mathbf{p}$ as:

$$\left[\mathbf{p}\ \left|\mathbf{I}_x(\mathbf{p})\right|\ \left|\mathbf{I}_y(\mathbf{p})\right|\ \sqrt{\mathbf{I}_x^2(\mathbf{p})+\mathbf{I}_y^2(\mathbf{p})}\ \operatorname{atan}\frac{\left|\mathbf{I}_y(\mathbf{p})\right|}{\left|\mathbf{I}_x(\mathbf{p})\right|}\ \mathbf{G}(\mathbf{p})\ \sqrt{\mathbf{G}_x^2(\mathbf{p})+\mathbf{G}_y^2(\mathbf{p})}\right]^\top \tag{13}$$

where $\mathbf{I}_x$, $\mathbf{I}_y$ and $\operatorname{atan}\frac{|\mathbf{I}_y(\mathbf{p})|}{|\mathbf{I}_x(\mathbf{p})|}$ have the same meanings as in Eq. (1). $\mathbf{G}(\mathbf{p})$ denotes a foreground probability value (a real number between 0 and 1 indicating the probability that the pixel $\mathbf{p}$ belongs to the foreground) computed using the robust background subtraction technique described in [21], and $\mathbf{G}_x$ and $\mathbf{G}_y$ are the corresponding first-order derivatives. With respect to the features of Eq. (1) [18], the main difference is the use of the two foreground related measures instead of the second-order intensity derivatives $\mathbf{I}_{xx}$ and $\mathbf{I}_{yy}$. We expect the former to be more informative in the context of video surveillance than the latter ones.[2]

When examining the features in $\mathbf{H}'$, we can qualitatively expect the intensity features to provide shape and texture information. The foreground features will mainly provide shape information, but not only thanks to the use of foreground probability maps as observations (e.g. notice in Fig. 2 how often there is some contrast between different body parts in these maps). Another motivation for the use of foreground probability as features is the following. As people can remain static for a long

[1] Note that all computation -background subtraction, gradient computation, integral images, etc- are done on the transformed image. The resulting black boundary regions do not introduce specific problems. In practice, we just set to zero the image derivatives of the boundaries that are artificially created, and only applied our detector to test windows containing less than 20% of black boundary regions.

[2] Note that we could have added the foreground features to the features defined in Eq. (1) rather than substituting two of them, leading to a feature vector of dimension 10. Although this could have increased the detection accuracy, we did not investigate this choice as it would have required the computation and storage of 65 integral images instead of 44, generating a 50% increase of memory and computation of the preprocessing step that would have impaired our real time objective.

period of time (and thus start being incorporated into the background), or wear clothes with similar color to that of the background, the foreground probability can be low (see the top right example in Fig. 2). If one would extract a binary foreground map (e.g. by thresholding), important information would be lost: regions could be labelled as belonging to the background although they have a small but non zero probability measure still suggesting the presence of a foreground object. In addition, shape artefacts would be introduced, like for instance boundaries inside people's bodies (see Fig. 3b). For these reasons, we prefer to keep the real values of the foreground probability map as input feature.

The use of covariance features fusing intensity and foreground observations is also extremely useful when multiple people occlude each other. In this difficult case, decisions have to be taken based on regional information, and appearance only methods might be quite confused. By requiring consistency between appearance and foreground shape features (as measured through covariance analysis in subwindows), we expect our detector to be more robust in presence of partial occlusion, and more generally against false alarms that could be due to human-like appearance shapes.

Finally, as we use a cascade of classifiers, foreground features will also help to quickly discard window candidates containing no or poor foreground information.

5.3. Weak Classifiers with Feature Subsets

Mapping the covariance features to the Euclidian space has been shown to work better than using the raw covariance coefficients as input to the LogitBoost algorithm [18]. However, one main issue with this mapping is that it involves *costly* matrix operations at *run-time*. More specifically it requires to perform a singular value decomposition (SVD) to compute a matrix logarithm (cf Eqs. (8)-(10)), an operation whose cost increases quadratically with the feature dimension, as illustrated in Fig. 5. One option to speed up the process could be to decrease the overall feature size d, by removing some of the features in $\mathbf{H}'$. However, this would be at the cost of performance, since it is obvious that some information would definitely be lost. We propose instead to build the weak classifiers from *subsets* of the complete image feature set. In this way, all the image features are kept and the most discriminative subset covariances (defined on lower dimensional Riemannian manifolds) can still be mapped into Euclidean space for binary classification according to the scheme presented in Section 4. Note that weak classifiers will be based on different subsets, and thus information about all image features will be exploited.

The lack of consistent relations between the covariance coefficients of *all* image features is another important motivation to use feature subsets. In other words, the manifold spanned by the training data points in the high-dimensional Riemannian space can be quite complex, resulting in noisy mappings. In this sense, using low-dimensional covariance matrices can be interpreted as a dimension reduction technique, and be more effective for classification.

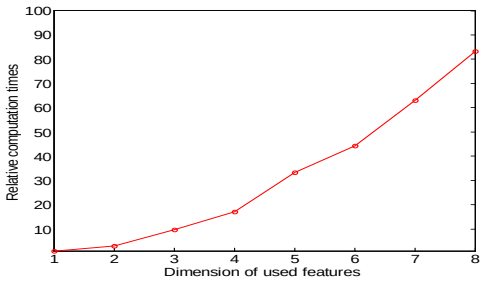

Fig. 5: Relative computation time of LogitBoost classifiers composed of 10 weak classifiers, for different feature sizes. Size one is taken as reference.

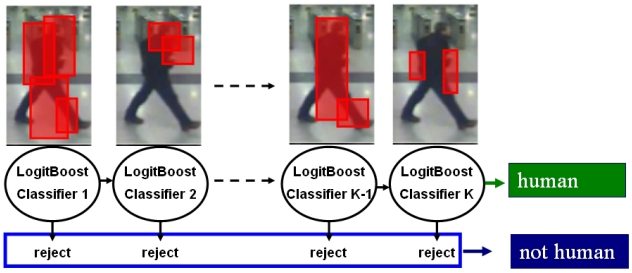

Fig. 6: Cascade of LogitBoost Classifiers. Each LogitBoost classifier is composed of weak classifiers relying on feature subset covariances computed from subwindows.

5.4. Using Mean Features

The covariance provides the second order moments of the image features over subwindows. In some cases, we believe that the first order moments, the means, could be discriminant as well. For instance, a high mean of the intensity gradient along some well placed vertical subwindow should be a good indicator of a human torso, or foreground pixels in the upper central part could denote the presence of a head. We thus propose to use these means as additional features in the LogitBoost algorithm. Since these features lie in a Euclidean space we don't need any form of mapping for them. However, to be robust against illumination changes, we normalize the subwindow mean vector entries of $\mathbf{m}_r$ w.r.t. the corresponding entries of the mean vector $\mathbf{m}_R$ of the detection window R, which results in $\mathbf{m}'_r$. Our weak classifiers are thus defined as: $f_l(\mathbf{X}_r) \triangleq g_l(h(\mathbf{C}'_r), \mathbf{m}'_r)$ where h is the mapping function defined in Eq. (8) (cf Section 4). In other words, we use the concatenation of the mapped covariance features with the normalized mean features as input to the linear function g_l used in the LogitBoost classifier.

6. Algorithm Description

In this section, we describe technical aspects of the cascade training and of our post-processing detection stage. Details can be found in [22].

6.1. Training the Cascade

The detector was implemented within a standard cascade of LogitBoost rejection classifiers, as illustrated in Fig. 6. The

procedure to train a LogitBoost classifier is the same for each level of the cascade (only the training set differs), and follows the approach described in Section 4.1. Below, we provide details on the different algorithmic steps.

Algorithm 1 : Training and selection of weak classifiers based on covariances of m-dimensional feature subsets.

$f^* = TrainAndSelect(\{\mathbf{Q}_i, b_i, z_i, w_i, \}_{i...N}, F)$

Input: $\{\mathbf{Q}_i, b_i, z_i, w_i\}_{i...N}$ and F. $\mathbf{Q}_i$ is an image example, $y_i \in [0, 1]$ is the class label, z_i and w_i are the response value and weight of the example $\mathbf{Q}_i$. F is the current version of the strong classifier.

Output: a weak classifier, defined by a subwindow, a feature subset, the point $\boldsymbol{\mu}$ at which the tangent space is defined and the coefficients of the regressor g.

- Select N_w subwindows
- For each selected subwindow r, select N_s feature subsets of size m
  - ∗ For each selected subset s, learn a $f_{r,s}$ weak classifier:
    - Extract the normalized covariance matrices $\mathbf{X}_i$ and the normalized mean vectors $\mathbf{m}'_i$ of the subwindow r from the examples $\mathbf{Q}_i$
    - Compute the weighted mean $\boldsymbol{\mu}_{r,s}$ of all the data points $\{\mathbf{X}_i\}_{i=1...N}$.
    - Map the data points to the tangent space at $\boldsymbol{\mu}_{r,s}$, $\mathbf{x}_i = \mathrm{vec}_{\boldsymbol{\mu}_{r,s}}(\log_{\boldsymbol{\mu}_{r,s}}(\mathbf{X}_i))$
    - Fit the linear function $g_{r,s}(\mathbf{x}, \mathbf{m}')$ by weighted least-square regression (Eq. (6)).
    - Define $F_{r,s}(\mathbf{Q}_i) = F(\mathbf{Q}_i) + \frac{1}{2} f_{r,s}(\mathbf{Q}_i)$, $p(\mathbf{Q}_i) = e^{F_{r,s}(\mathbf{Q}_i)} / (e^{F_{r,s}(\mathbf{Q}_i)} + e^{-F_{r,s}(\mathbf{Q}_i)})$
    - Compute $\mathsf{L}_{r,s}$, the negative binomial log-likelihood of the data using Eq. (5).
- Return f^*, the weak classifier for which $\mathsf{L}_{r,s}$ is the minimum.

Training a cascade level: In the experiments, we used $K = 30$ cascade levels. At each cascade level k, the number ${N_L}^k$ of weak classifiers is selected by optimizing the LogitBoost classifier to correctly detect at least $d_{\min}$=99.8% of the positive examples, while rejecting at least $f_{\max}$=35% of the negative examples. In addition, we enforce a margin constraint between the probability of the positive examples during training and the classifier decision boundary. This is achieved in the following way. Let $p_k(\mathbf{Q})$ be the probability of an example $\mathbf{Q}$ being positive at the cascade level k, defined according to Eq. (4). Let $\mathbf{Q}_p$ be the positive example that has the $(d_{\min} \times N_p)$-th largest probability among all the positive examples and $\mathbf{Q}_n$ be the negative example that has the $f_{\max} N_n$-th smallest probability among all the negative examples, where N_p and N_n are the number of positive and negative examples used for training, respectively. Weak classifiers are added until $p_k(\mathbf{Q}_p) - p_k(\mathbf{Q}_n) > th_b$ where we set $th_b = 0.2$. Finally, at test time, a new example $\mathbf{Q}$ will be rejected by the cascade level k if $p_k(\mathbf{Q}) \leq \tau_k$, with τ_k equal to the value $p_k(\mathbf{Q}_n)$ computed at the last iteration of the training algorithm, i.e. when the margin criterion is met. In practice, adding this constraint increases the probability for true positives to be actually detected at run time.

To obtain the N_p and N_n training samples for cascade level k, we used a standard bootstrap procedure. The detector up to the $k-1^{th}$ level was applied to a set of N_{ptot} positive examples, and the N_p examples with the least probability of being positive at the last level were kept for training. Similarly, negative examples were selected as the false positive examples of the current detector applied to negative images containing no positive data until N_n examples are collected.

Training and selecting weak classifiers: A standard modification to the base LogitBoost algorithm was made: at each iteration l, there is not only one single weak classifier available. Rather, a collection of weak classifiers are learned and the one that minimizes the negative binomial log-likelihood given by Eq. (5) is actually added as f_l to form the decision function F, as described in Algorithm 1. The collection of tested classifiers $\{f_{r,s}, r = 1 \ldots N_w, s = 1 \ldots N_s\}$ is constructed by selecting N_w subwindows r of the detection window R, whose sizes are at least of 1/10 of the width and height of the detection window. Then, for each subwindow, a set of N_s m-dimensional feature subsets are selected for testing as follows.

When using the covariance between all image features, $N_s = 1$, i.e. there exist only one weak classifier for a given subwindow. However, when using subsets, we have the choice between several feature combinations. Rather than using random selection of the feature subsets to test, we adopted the following approach. For subsets of size 2, an exhaustive optimization for all combinations is feasible, as the training and testing of the weak classifiers is fast. For subsets of size $m > 2$, the training cost is much higher. Thus, we first perform an exhaustive training for all subsets of size 2, and then use the training results to predict and select for testing the k best subsets of size m most likely to provide good results for classification. This approach, detailed in [22], provides a better way of selecting good m-subset features than random selection, and saves a significant amount of time in training.

6.2. Post-processing

At test time, the binary detector is applied to windows of different positions and sizes. Usually, for one actual person in the image, several positive detections are obtained, as shown in Fig. 7(a). To merge these detections, we propose the following method. Let $\mathcal{X}_p = \{\mathbf{Q}_p\}_{p=1}^P$ be the set of P positive detections and $\{\mathbf{x}_p^c\}_{p=1}^P$ denote their image location. We define the reliability of each output $\mathbf{Q}_p$ as the sum of the weighted probabilities from each cascade level:

$$d_{rel}(\mathbf{Q}_p) = \sum_{k=1}^{K} \left((f_{\max})^{K-k} \times p_k(\mathbf{Q}_p) \right). \quad (14)$$

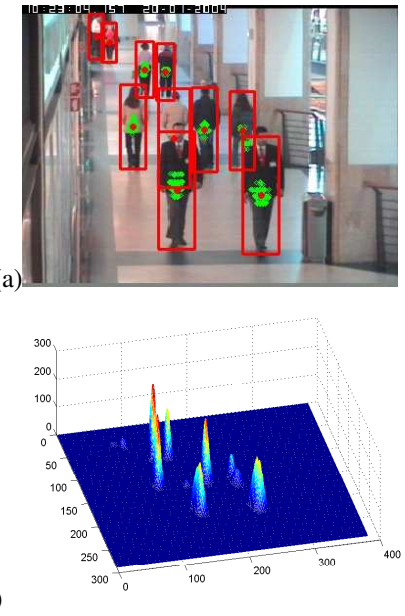

Fig. 7: Post-processing of detection outputs: (a) Green dots: positive detection window centers. Red dots: final detection extrema found via post-processing. (b) Smoothed reliability image corresponding to the image in (a).

Then, associating the reliability scores of the detected windows to their centers, we build a reliability image $\mathbf{D}_{rel}$ according to:

$$\mathbf{D}_{rel}(\mathbf{x}) = \max_{p} \left(d_{rel}(\mathbf{Q}_p) \times \delta(\mathbf{x} - \mathbf{x}_p^c) \right),$$

where $\delta()$ is the dirac function. Then, we smooth $\mathbf{D}_{rel}$ with a Gaussian kernel filter of bandwidth (σ_w, σ_h) corresponding to 1/10 of the average window size of all the detections. This is illustrated in Fig. 7(b). All local maxima in the smoothed image are considered as possible detection results, with their window sizes obtained as the weighted means of the window sizes of the neighboring detections. To further filter the detections, extrema are ranked according to their reliability. Then, a detection is removed if its overlap with another extrema with greater reliability is too large, where this removal process process starts with the less reliable detections.

7. Experimental Results

Different experiments were conducted to evaluate our method. In Section 7.1, we present a thorough evaluation on our target applications: the detection of people in surveillance data acquired from stationary cameras. In Section 7.2, we report results illustrating the benefit of the slant removal step. Finally, in Section 7.3, we used the INRIA still image database to compare our approach (use of feature subsets and mean features) against previous works [3, 18].

7.1. Human Detection in Video Sequences

In this section, we present our experiments on a large video database. We first describe our datasets and evaluation protocol, and finally report our results.

7.1.1. Training and Testing Datasets

We collected 10 indoor and 5 outdoor video sequences from the shopping center CAVIAR data[3], the PETS data[4], and several metro station cameras. Around 10000 positive examples were extracted from this dataset (see examples in Fig. 2). In addition to the presence of luggage and partial occlusions, there are large variations of appearances, poses, camera view-points, and extracted foreground images. Negative examples were obtained by: (i) collecting 1000 still images without people and coupling them with inconsistent foreground images; (ii) cropping about 10000 regions from the dataset which don't contain full human bodies; (iii) bootstrapping, i.e. by collecting more negative samples which 'look like' people after each cascade level, as explained in Subsection 6.1. In practice, a total of $N_p = 4000$ positive and $N_n = 8000$ negative examples were used to train a given LogitBoost classifier.

For testing, we set apart 523 images from video clips belonging to the above sequences but not used for training, and added data from 2 new videos. 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.

7.1.2. Evaluation Methodology

The detectors were applied to image subwindows with different locations, scales, and aspect ratios, according to: the width ranged from 25 to 100 pixels; the aspect ratio (height divided by width) ranged from 1.8 to 3.0. Positive detections were filtered out by keeping local detection maxima (cf Section 6.2). Two types of performance curves were measured by adding cascade levels one by one to the detectors.

Detection Error Tradeoff (DET) curves have been used to quantify the classifier performance at the window level [3, 13, 18]. They plot the miss rate, $\frac{\texttt{\#FalseNeg}}{\texttt{\#TruePos+\#FalseNeg}}$, versus false positives (here FPPW, the False Positives Per tested Window) on a log-log scale. The 1927 positive test samples are used to evaluate the miss-rate, while FPPW is obtained by testing all windows of the test data overlaping 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 test window.

Recall-Precision (RP) curves: RP curves are more appropriate to measure the detection accuracy from a user point of view [5, 15]. They integrate the post-processing steps, and thus, detectors with similar miss-rate for the same FPPW value may exhibit different RP curve behaviours: detectors with multiple but spatially consistent detections tend to produce less false alarms at the object level than detectors spreading their detection over multiple locations. Recall and precision are defined as $\frac{\texttt{\#TruePos}}{\texttt{\#TruePos+\#FalseNeg}}$ and $\frac{\texttt{\#TruePos}}{\texttt{\#TruePos+\#FalsePos}}$, respectively. A detected output is said to match a ground truth object if their F_{area}

[3] Available via http://homepages.inf.ed.ac.uk/rbf/CAVIAR/
[4] Available via http://www.cvg.rdg.ac.uk/PETS2006/data.html

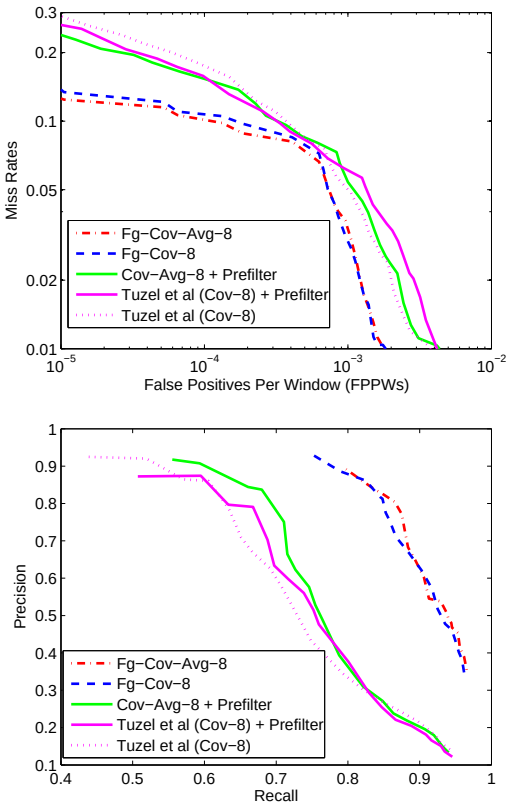

Fig. 8: Performance of different approaches with 8-dimensional features. Left, Miss rate vs false positive rate. Right, precision-recall curves.

measure is above 0.5. Only one-to-one matches are allowed between detected and ground truth regions.

7.1.3. Results

We consider the method of Tuzel *et al.* [18] as our baseline. Three main improvements to this method were made: integration of foreground probability features, use of the feature average in addition to covariance, and selection of feature subsets. We trained several detectors with or without these improvements and named them accordingly. For example, the detector *Fg-Cov-Avg-8* uses the covariance and the average of the 8-dimensional features containing foreground information (cf Eq. 13).

Foreground features. We trained four detectors with/without the use of foreground information and average features. To allow a fair comparison, a prefilter is applied to the baseline [18]: only windows containing a sufficient percentage of foreground pixels are tested. A percentage of 20% was used. Thresholds above this value were reducing the performance, by rejecting more true positive than false alarms.

The plots in Fig. 8 show that the use of foreground observations in the learning process rather than as a preprocessing step provides much better detection performance. For instance, for a precision of 0.9, 60% of the people are actually detected with [18], compared to 80% using our approach. In addition, note that the use of the feature averages provides similar results.

There are two main reasons why the foreground prefilter does not improve much the detection performance. First, foreground information is only used to reject detection, not to accumulate evidence of a person's presence. Thus, when the grayscale image is not sufficient to differentiate a person from the background clutter, the foreground does not help to take the decision, as illustrated in the left images of Fig. 9. The second reason is that the percentage of foreground pixels inside a window is only a crude indicator for rejection, insufficient to properly distinguish between false alarms and true detection in the presence of cluttered foreground (presence of multiple people, shadow), as shown in the right image of Fig. 9.

Performance of feature subsets. We trained three new detectors relying on 2, 3 and 4-subset features (*Fg-Cov-Avg-2* to *Fg-Cov-Avg-4*, respectively), and a combined detector based on 2, 3 and 4-subset features for the cascade levels 1 to 15, 16 to 25, and 26 to 30, respectively (*Fg-Cov-Avg-[2,3,4]*). Fig. 10(a) shows the obtained RP curves. Interestingly, the use of subset features results in detection performance similar to those obtained with the full 8-dimensional set, with *Fg-Cov-Avg-[2,3,4]* providing the best results overall. This confirms our hypothesis that the selection of the most consistent correlation terms between features is enough to achieve good detection performance. Figures 11 provide statistics about the frequency of the selected features. The image gradient orientation is the dominant feature (it confirms the importance of edge orientation as in HOG features), and is often selected along with foreground probability value and gradient. This further demonstrates the interest of exploiting jointly the appearance and foreground features.

Computational speed. The level of performance achieved by our detectors comes with a significant computational gain w.r.t. our baseline. The computational complexity was evaluated by applying the detectors to the test data and measuring the average number of windows per second that they can process. The same computer was used in all the cases. Results are shown in Fig. 10(b). The first observation is that the mean features offer a speed gain of nearly 30% (e.g. compare Tuzel *et al.* [18] with *Cov-Avg-8*). Secondly, in addition to improving performance, foreground features also increase the speed by rejecting false hypothesis more quickly (compare *Fg-Cov-Avg-8* against *Cov-Avg-8*). Finally, the main computational gain is obtained by using feature subsets. For instance, the detectors *Fg-Cov-Avg-2* and *Fg-Cov-Avg-[2,3,4]* run around 13 times faster than *Fg-Cov-Avg-8* (and more than 20 times faster than [18]). We can apply these two detectors to videos of size 384x288 and process around 5 frames/sec. Indeed, most of the time is spent on the computation of the image features (background subtraction, covariance integral images), rather than in the detection part itself.

Exploiting 3D geometry. Finally, to further speed up the process and improve detection performance, we can exploit rough ground plane geometrical constraints to limit the human heights from 150cm to 220cm. Results are shown in Fig. 10(a), and show a consistent gain due to the removal of some of the false positives windows.

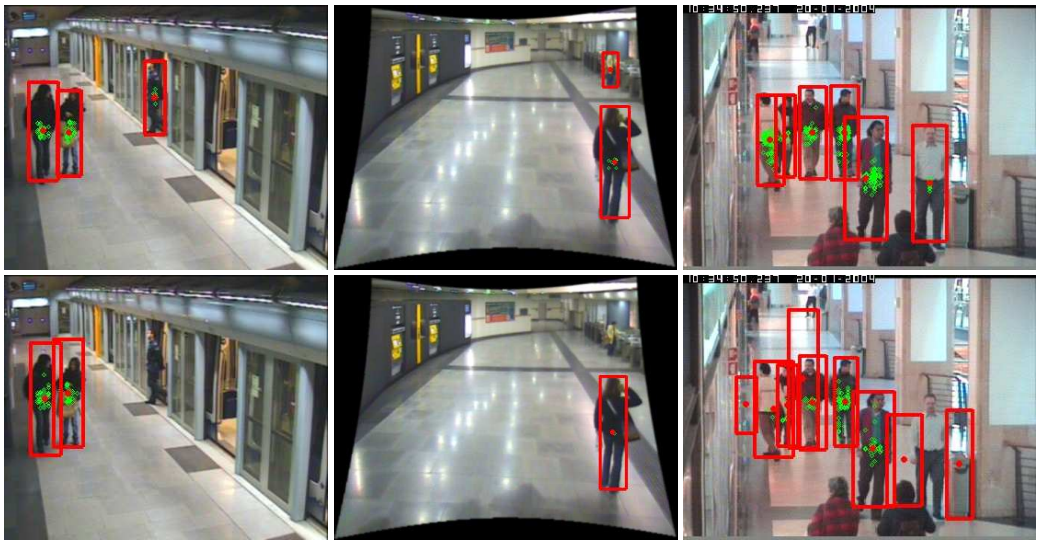

Fig. 9: Results with the *Fg-Cov-8* detector exploiting foreground features in the covariances (Top) and Tuzel *et al.*'s method with foreground prefilter (Bottom).

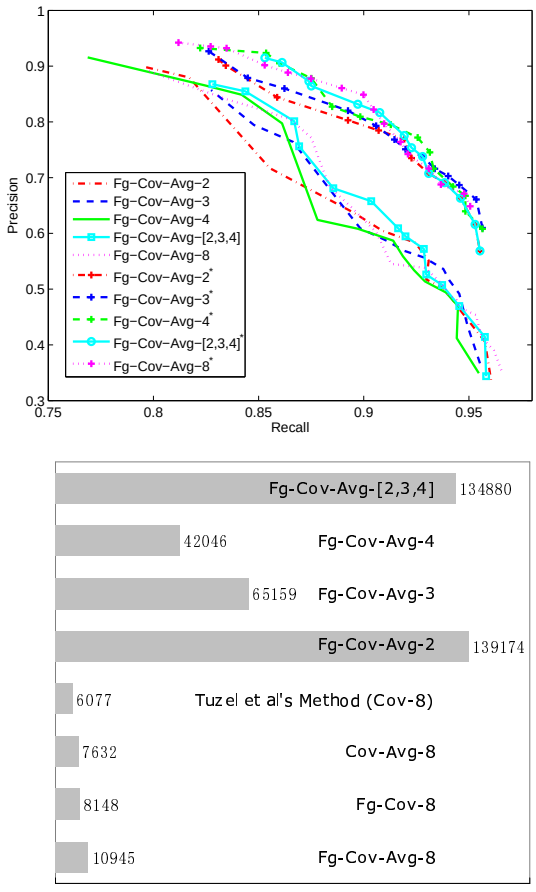

Fig. 10: Top: Performance of detectors with (labels with a $*$ superscript) or without a ground-plane geometrical constraint. Bottom: Average number of tested windows (per second).

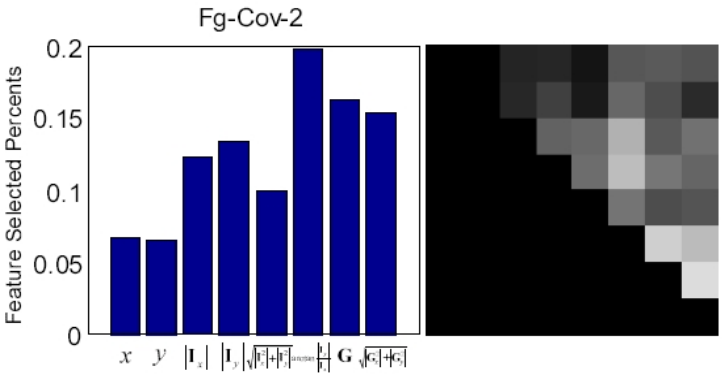

Fig. 11: Percentage of times a given image feature (left: blue bars) and a given image feature pair (right: light, high percentage; dark, low percentage) is selected as part of a feature subset for classification.

Illustrations. Fig. 12 shows detection examples obtained with the *Fg-Cov-Avg-2** detector. Despite the large variability of appearances and view points, partial occlusions, and the overall small people size, there are only few false positive and false negative detections. In general, the main errors come from strong specular reflections and cast shadow, bad foreground results produced by moving objects (moving escalator in the Metro scene), or occlusions by other persons or objects.

7.2. Test on Slant Removal Data

To suppress the slant of people in images acquired using cameras with short focus lenses, we can apply to the input images the homography transform that we propose (cf Section 5.1) which maps the vertical vanishing point to infinity. The human detector is then applied on the resulting video streams.

To evaluate the impact of such a preprocessing step on the performance, we collected test samples (29 images containing 89 people) from the 3 video streams shown in Fig. 13 representing a typical case of the issue. We applied the detector *Fg-*

Fig. 12: Detection examples. Green dots: positive detection. Red dots and bounding boxes: final detections after post-processing.

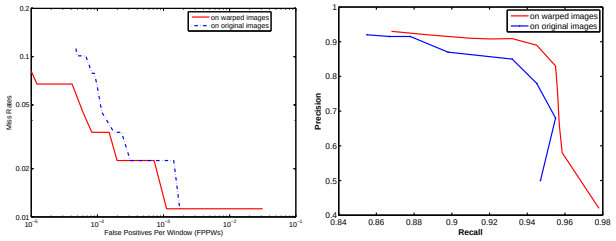

Fig. 14: Detection performance on data with slanted people.

Cov-Avg-[2,3,4] on this data with and without the geometrical correction. Typical results shown in Fig. 13 clearly illustrate the benefits of the approach. Performance curves plotted in Fig. 14 show that the detection performance on slant-removed images are always better than those obtained on the original images.

7.3. Test on INRIA database

We further evaluated the efficiency of feature subsets on the INRIA still image human database [3], following the experimental protocol described in [3, 18]. The database contains 1774 human positive examples and 1671 negative images without people. We used 1208 positive samples (and their reflections w.r.t. a central vertical axis) and 1218 negative images for training. For each cascade level, the Logitboost algorithm was trained using all the positive examples and $N_n = 10000$ negative examples generated by boostrapping. The rest of the data (566 positive examples and 453 negative images) was used for testing and building the DET curve [3, 18] (cf Subsection 7.1.2). To our knowledge, at the time of the paper submission, Tuzel *et al.* [18] obtained the best detection results, on this database, and outperforming the methods of Dalal & Triggs [3] and Zhu et al. [25].

Fig. 16 shows detection results on challenging images, while quantitative results are displayed in Fig. 15. They confirm the performance reported in [18] (e.g. with a miss-rate of 7.5% at 10^{-4} FPPW rate vs 6.8% in [18]). Secondly, unlike in the video case, low-dimensional subset features consistently lead to better results than full covariance features (e.g. compare *Cov-2* with *Cov-8* [18]). This result might be explained by the smaller amount of positive training data (around 2400 here vs 1000 in the video case) which makes the training of weak-classifiers in the 33 dimensional full covariance space noisier than when using subsets. This shows that the selection of the most consistent covariance coefficients instead of using them all is a better strategy. Thirdly, while the use of mean features provides similar results for most of the detectors, it actually slightly degrades the results of the best performing one *Cov-Avg-2*. Finally, the use of the mean features and importantly the feature subsets dramatically increases the detection speed. The *Cov-Avg-2* detector can process around 15 times more windows than *Cov-8*.

8. Conclusions

In this paper, we investigated a fast method to detect humans from surveillance videos. We proposed to take advantage of the stationary cameras to perform background subtraction and jointly learn the appearance and the foreground shape of people in videos. To this end, we relied on a cascade of Logitboost classifier learning framework using covariance matrices as object descriptors [18].

The novelties of the approach are summarized as follows. First, we proposed a simple preprocessing step to remove people slant in images taken from large field-of-view cameras, allowing to improve the detection performance while keeping the computational efficiency of integral images. Second, by learning the correlation degree between appearance and foreground shape features, the method proved to be powerful and much better than using the correlation between appearance features alone, even when using background subtraction to remove false alarms. Finally, to build our weak classifiers, we proposed to only rely on subsets of the complete image feature space, and to exploit the means of the image features along with their covariance. This reduced the computational cost by 15 to 22 times

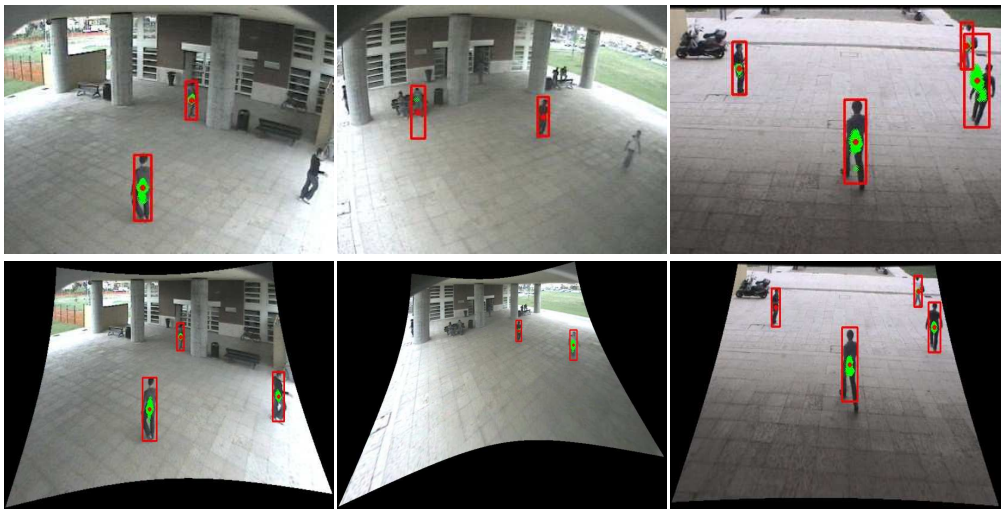

Fig. 13: Detection results on original images (Top) and warped images with infinite vertical vanishing point (Bottom).

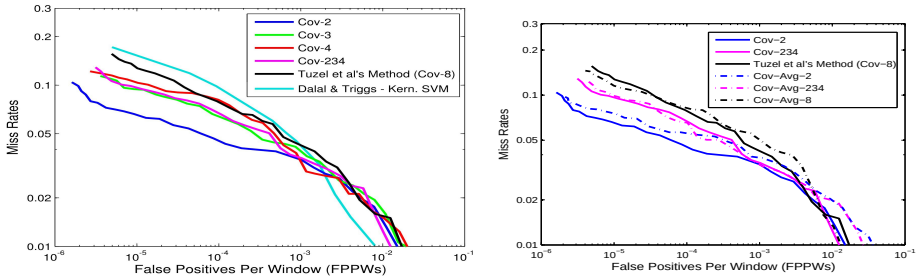

Fig. 15: INRIA database. Performance of five detectors (Left) and comparison of detectors with/without mean features (Right). The HOG curve displayed in the left plot are the best results reported in [3].

w.r.t. using the covariance between all features, while providing equivalent or sometimes even better performance. These novelties resulted in an accurate and near realtime detector, as shown by experiments on real, large and challenging datasets.

There are several areas for future work. Although our algorithm is very fast, the bottleneck lies in the number of integral images which need to be computed. Reducing this number can be obtained by simply using fewer image features, although this might be at the cost of significant performance decrease. A better alternative, possible only when relying on covariances of size 2 feature subsets, might be to only build our weak-classifiers from a reduced number of image feature pairs.

The fixed template approach that we used provided good results. Yet it does not account for the articulated nature of the human body and the appearance and shape variability that it creates in the training data. To account for this, one possibility is to train a collection of classifiers for different body poses or learn classification trees, as done for instance in multi-view face detectors [11].

Finally, most of the errors are made in cluttered foreground when multiple people occlude each other partially. One promising direction of research to handle this issue would be to train a classifier to perform the joint detection of humans in occlusion situations. This could be done by building different body part detectors, and by learning their response in different occlusion configurations.

9. Acknowledgement

This work was supported by the European Union through the Information Society Technologies CARETAKER project (FP6-027231), and VANAHEIM (FP7-248907) projects.

References

[1] J. Begard, N. Allezard, and P. Sayd. Real-time humans detection in urban scenes. In *British Machine Vision Conf. (BMVC)*, 2007.

[2] C. Carincotte, X. Naturel, M. Hick, J-M. Odobez, J. Yao, A. Bastide, and B. Corbucci. Understanding metro station usage using closed circuit television cameras analysis. In *IEEE Int. Conf. on Intelligent Transportation Systems (ITSC)*, Bejing, October 2008.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Conf. Comp. Vis. & Patt. Recognition*, volume 1, pages 886–893, June 2005.

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

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

[6] H. Elzein, S. Lakshmanan, and P. Watta. A motion and shape-based pedestrian detection algorithm. In *IEEE Intelligent Vehicule Symposium*, pages 500–504, 2003.

[7] J. Friedman, T. Hastie, and R. Tibshira. Additive logistic regression: a statistical view of boosting. *Ann. Statist.*, 23(2):337407, 2000.

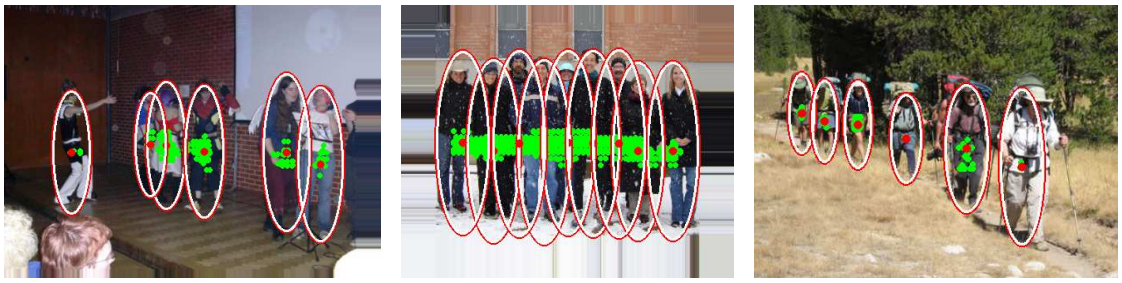

Fig. 16: Detection examples on still images.

[8] D. Gavrila and V. Philomin. Real-time object detection for "smart" vehicles. In *Conf. Comp. Vis. & Patt. Recognition*, pages 87–93, 1999.

[9] M. Hussein, W. Abd-Almageed, Y. Ran, and L. 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.

[10] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition*, pages 878–885, 20-25 June 2005.

[11] S. Li and Z. Zhang. Floatboost learning and statistical face detection. *Trans. Pattern Anal. Machine Intell.*, 26:1112–1123, 2004.

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

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

[14] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. *Int. Journal of Comp. Vision*, 66(1):41–66, 2006.

[15] E. Seemann, M. Fritz, and B. Schiele. Towards robust pedestrian detection in crowded image sequences. In *Conf. Comp. Vis. & Patt. Recognition*, 2007.

[16] H. Sidenbladh. Detecting human motion with support vector machines. In *Int. Conf. on Pattern Recognition (ICPR)*, volume 2, pages 188–191, 2004.

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

[18] O. Tuzel, F. Porikli, and P. Meer. Human detection via classification on riemannian manifolds. In *Conf. Comp. Vis. & Patt. Recognition*, 2007.

[19] 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.

[20] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *Trans. Pattern Anal. Machine Intell.*, 19(7):780–785, July 1997.

[21] J. Yao and J-M. Odobez. Multi-layer background subtraction based on color and texture. In *CVPR Workshop on Visual Surveillance (CVPR-VS)*, June 2007.

[22] J. Yao and J.M. Odobez. Fast human detection in videos using joint appearance and foreground learning and covariances of image feature subsets. Technical Report 19, Idiap research institute, 2009.

[23] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *Conf. Comp. Vis. & Patt. Recognition*, 2003.

[24] J. Zhou and J. Hoang. Real time robust human detection and tracking system. In *Work. on Object Tracking and Classif. in and Beyond the Visible Spectrum*, 2005.

[25] Q. Zhu, M.C. Yeh, K.T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *Conf. Comp. Vis. & Patt. Recognition*, pages 1491–1498, 2006.