FAST HUMAN DETECTION IN VIDEOS USING JOINT APPEARANCE AND FOREGROUND LEARNING FROM COVARIANCES OF IMAGE FEATURE SUBSETS

Jian Yao Jean-Marc Odobez

Idiap-RR-19-2009

JULY 2009

Fast Human Detection in Videos using Joint Appearance and Foreground Learning and Covariances of Image Feature Subsets

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. Traditional approaches exploit background subtraction as an attentive filter, by only applying the detectors on foreground regions. This doesn't take into account that foreground observations contains human shape information which can be used for detection. In this paper, we propose a method to learn the correlation between appearance and foreground information. It is based on a cascade of LogitBoost classifiers which uses covariance matrices computed from input image features as object descriptors. We account for the fact that covariance matrices lie in a Riemanian space, introduce different novelties -like exploiting only covariance sub-matrices- to reduce the induced computation load, as well as an image rectification scheme to remove the slant of people from the image when dealing with wide angle cameras. Evaluation on a large set of videos show that our approach performs better than the attentive filter paradigm while processing from 5 to 20 frames/sec. In addition, on the INRIA human (static image) benchmark database, our sub-matrix approach performs better than the full covariance case while reducing the computation cost by more than one order of magnitude.

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. It is often a first and essential step before more complex activity or behavior analysis module can be applied, and has therefore many applications related in surveillance and the monitoring of smart spaces such as meeting rooms or offices. Indeed, improving human modeling and detection is crucial for tracking algorithms, especially when scenes become crowded. However, human detection can also be used directly. For instance, in [3], a human detector was continuously applied to evaluate the numbers 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 on a given time and a given day of the week.

In this paper, we address the fast detection of humans in videos recorded by stationary cameras. This task has important challenges 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 difference in camera view point or illumination variations.

Email address: `Corresponding author, odobez@idiap.ch` (Jean-Marc Odobez)

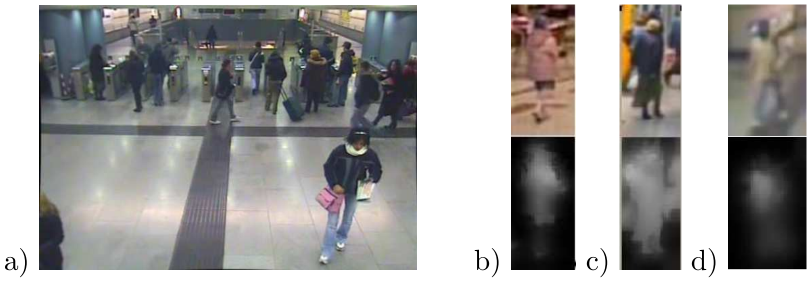

Fig. 1: (a) Example of surveillance metro image. (b)-(d) Appearance and foreground samples from other data: (b) the person is camouflaged by the background, while foreground is more precise; (c) and (d) foreground is noisier but appearance remains discriminative. More examples can be found in Fig. 3.

In addition, especially in the surveillance context, as illustrated in Fig. 1, 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 partially occlude each other, and the color of their clothes are often similar (and similar to the background too). On the other hand, surveillance cameras are often static, which allows to exploit background subtraction techniques.

In this paper, we investigate the joint use of appearance and foreground features to detect humans in video sequences. Examples of such features are provided in Figs. 1(b)-(d) and Fig. 3. As can be seen in Fig. 3, foreground images can provide quite discriminant shape information for detecting people. In crowded scenes, occlusion between people make the interpretation of foreground information more difficult, as is the lack of contrast between a person and the background. Still, parts of the foreground images can be characteristic of the human shape (head or legs), and the shape and texture of the input image will provide the sufficient information to correctly localize people, as illustrated in Fig.1(c)-(d). Conversely, in cluttered regions (e.g. in presence of high texture background), appearance might not be sufficient to assess the presence of a person. In this case, foreground information, which is more difficult to camouflage, will be helpful to reinforce the appearance observations, as illustrated in Fig. 1(b).

Overall, the high degree of correlation between the appearance and the foreground features at different places of a human template (head, body sides and legs) is a reliable indicator for human detection. Thus, in this paper, we rely on the learning framework of Tuzel et al. [25] which uses covariance matrices computed on subwindows of the detection window as human descriptors, and is based on a cascade of LogitBoost classifiers.

We extended the framework of Tuzel et al in several ways to account for the foreground information and to speed up the computation. Fusion between foreground and appearance information was performed by using both static and temporal features from the still and foreground images. This has several advantages. First, due to the cascade approach, the foreground features plays a Region Of Interest (ROI) role allowing for faster processing. This will be achieved in a more informative way as rejection will be based on correlation analysis between static and foreground features rather than foreground alone. Secondly, we propose to use *continuous* foreground probabilities rather than background subtraction *binary* masks. This choice alleviates the need for setting the background detection threshold, which is a sensitive issue in practice. When a too low threshold is used, the resulting over-detection produces less informative masks. However,when a too high threshold is used, there will be missed detections. Our choice should thus be more robust against variations in the contrast between humans and the background.

Secondly, in [25], one of the crucial step of the algorithm is to map 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. However, this mapping, which is performed for each weak classifier at run time, is slow for high dimensional image feature space. Also, as there might not always exist consistent correlation between *all features* in the training data, the learned weak classifiers may have poor generalization performance at test time, when the training data is not large enough. To address these issues we propose to only exploit a subset of the complete image feature space to build a given weak classifier. 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. Embedded in the LogitBoost framework, subsets with the most consistent and discriminant covariances are selected.

Thirdly, we investigated the use of image features' mean as additional input for the weak classifiers. Intuitively, it should be useful at describing the presence of strong edges or foreground information at different positions of the template.

The final novelty of the paper is an image rectification step allowing to reduce people geometric appearance variability in images due to the use of 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 distorsions, as illustrated in Fig. 5. This is a problem for human detectors which often consist of applying a classifier on rectangular regions, or in other tasks (e.g. tracking) when integral images are used to efficiently extract features over boxes. The variation of people orientation in the image affects the consistency of the extracted features (with respect to an upright standing) and will ultimately harm the detection or tracking processes. To remove this variability, we propose a simple rectification scheme which is applied to the input image as a pre-processing step. It consists of mapping the 3D vertical lines into 2D vertical image lines, as illustrated in Fig. 5. The method is shown to introduce negligible image distorsions.

Experiments were conducted on a large dataset comprising videos from different publicly available databases to assess the different algorithm components and demonstrate the validity of our approach. In addition, 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 works. Section 3 introduces the covariance features. In Section 4 we present a brief description of the LogitBoost classification algorithm for Riemanian 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

There is an abundant literature on human detection. We first survey techniques which were applied to still images, and then review more specifical works on human detection in videos.

A first category of techniques to detect human in still images consists of modeling the human by body parts whose locations are constrained by a geometric model [12, 16, 13]. In [16], body parts were represented by combinations of joint orientation and position histograms. Separate Adaboost detectors were trained for the face and head as well as front and side profiles of upper and lower body parts. Human localization was then obtained by optimizing the likelihood of part occurrence along with the geometric relation. As another example, Leibe et al [13], proposed a probabilistic human detector for crowded scenes that combines evidence from local features with a top-down segmentation and verification step. However, while these techniques usually attempt to provide a general framework that can be applied to complex objects [15], they usually do not lend themselves to fast implementations. In addition, while they usually take into account, in a quite accurate fashion, the articulated nature of the human body, this might not be so appropriate when dealing with low resolution human images such as those often encountered in surveillance videos.

A second category for detection in still images is based on applying a fixed-template human detector for all possible subwindows in a given image. Methods differ by the types of input features and the training approaches. In [10], a direct approach was used in which edge images were matched to a set of human examplars using a chamfer distance. In [18], a SVM classifier was learned using Haar wavelets as human descriptors. Recently, Dalal and Triggs [4] proposed a very good detector that relied on a linear SVM classifier applied to densely sampled histograms of orientation gradient (HOG). As this approach is relatively slow, the application of the cascade and boosting framework to the HOG features was proposed in [32, 1]. Finally, very recently, Tuzel et al [25] proposed their method based on cascade of LogitBoost classifiers using covariance as object descriptors which outperformed [32, 4]. These techniques proved to be robust but were mainly used to images with enough human image resolution. Their performance on surveillance data or the use of foreground information was not investigated.

Although human tracking is the topic of intensive research, few works have actually investigated the human detection task from videos. Optical flow has been used to detect human in videos. To detect pedestrian in front of a moving car, Elzein et al [7] first extracted candidate detection via optical flow analysis, followed by a verification step relying on a still image human detector based on shape wavelets. In [21], Sidenbladh used a SVM trained on optical flow patterns to create a human classifier. Dalal et al [5] presented a more robust approach by extending their still image HOG detector to videos using histograms of differential optical flow features in addition to HOG, which was shown to greatly improve over the static one. While these techniques do not assume a static camera, they require good quality optical flow computation (to avoid flow smoothing at discontinuities), which is usually expensive to compute. Also, it is not very clear how these techniques perform in presence of partial occlusion.

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 [27, 7], which are further classified into human or non-human entities using blob shape features if necessary [31]. 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. For instance, Beleznai et al [2] proposed to apply a mean-shift algorithm inside the blob to cluster color information. In [29], a Bayesian segmentation of foreground blob images by optimizing the layered configuration of people using a data driven MCMC approach is conducted. Head candidates are extracted from foreground images and intensity. The criterion, however, does not include shape information, which is quite important for good detection, and the foreground criterion often becomes ambiguous in case of large overlap. Other authors applied a static human detector [7, 31, 11] on extracted foreground regions, thus following a common trend of using background subtraction results as a ROI selection process. Multiple object tracking techniques can help to solve this issue. Color models of individuals allow the separation of people within the same blobs, in addition to criterions which favors the configuration of people which best covers the foreground blobs [8, 30, 22]. Still, how to initialize and learn the color models remains an issue when groups of people enter the scene, and color models may not be so discriminative between people.

Finally, the only work we found which was using spatio-temporal features for human detection in surveillance is the work of Viola et al [26]. They built an efficient detector applicable to videos using a cascade of Adaboost classifiers relying on Haar wavelet descriptors but extracted from spatio-temporal differences. While invariance scaling 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 like features are somewhat crude and recent works has shown that better shape features can be exploited (e.g. HOG features [4] or covariances [25]).

3. 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 features such as the intensity, the gradient, filter responses, or any feature which is expected to characterize well the appearance of a human person. Let us denote by d the number of features computed at each point location. Accordingly, we can define a $W \times H \times d$ feature image $\mathbf{H}$. To detect persons in still images, Tuzel et al. [25] proposed to use the following set $\mathbf{H}(\mathbf{x})$:

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

where $\mathbf{I}_x$, $\mathbf{I}_y$, $\mathbf{I}_{xx}$ and $\mathbf{I}_{yy}$ denote the first-order and second-order intensity derivatives, and $\arctan\frac{|\mathbf{I}_y(\mathbf{x})|}{|\mathbf{I}_x(\mathbf{x})|}$ represents the orientation of the gradient at the pixel position $\mathbf{x}$. With the above choice of features, the input image is mapped to a $d = 8$ dimensional feature image.

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:

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

where $\mathbf{m}_R$ is the mean vector in the region R, i.e. $\mathbf{m}_R = \frac{1}{|R|}\sum_{\mathbf{x} \in R} \mathbf{H}(\mathbf{x})$, and $|\cdot|$ denotes the set size operator. The covariance descriptor of a region with the above selected features is an 8×8 matrix. The covariance matrix is a very informative descriptor. It encodes information about the variances of the features, their correlations with each other, and also the spatial layout of the features inside the window since the feature correlation with the pixel position $\mathbf{x}$ is also computed. Since covariance matrices are symmetric, we can represent them by a vector of dimension $\frac{d \times (d+1)}{2}$, e.g. a 36-dimension vector for the above 8×8 covariance matrix[1]. When such covariance matrices need to be computed a large number of times, integral images (one per correlation coefficient plus one per feature to extract the mean values) can be used to gain efficiency [24].

Covariance normalization: By construction, the covariance features are robust with respect to 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

[1] Note however that since we are not interested in the variance of the x or y features, and in the $x \times y$, only 33 dimensions are really useful.

a larger test window R (the window in which we test the presence of a person). We first compute the covariance of 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 as follows:

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

where $\mathbf{C}'_r$ denotes the resulting normalized covariance.

4. LogitBoost Learning on Riemannian Space

In [25], Tuzel et al used a cascade of LogitBoost rejection classifiers to detect humans using covariance features. In the following, we first briefly introduce the standard LogitBoost algorithm on vector spaces [9], which is a variant of the popular Adaboost algorithm. Then we describe the modifications which were proposed in [25] to account for the fact that covariance matrices do not lie in the Euclidian space.

4.1. The LogitBoost Algorithm

In this section, let $\{\mathbf{x}_i, y_i\}_{i=1\ldots N}$ 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

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

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

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

through Newton iterations [9]. At each iteration l, this is achieved by solving a weighted least-square regression problem:

$$\sum_{i=1}^{N} w_i \|f_l(\mathbf{x}_i) - z_i\|^2, \tag{5}$$

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

$$w_i = p(\mathbf{x}_i)(1 - p(\mathbf{x}_i). \tag{6}$$

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.

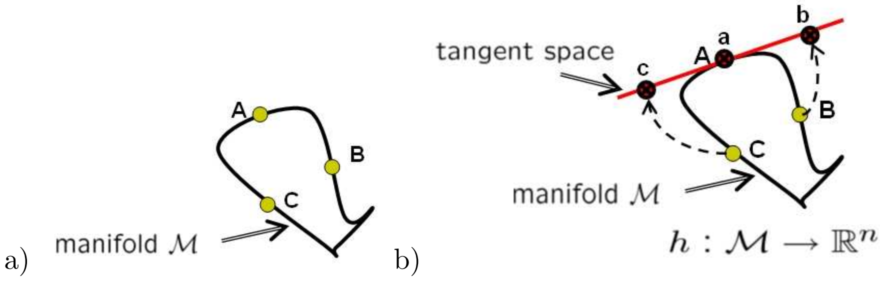

Fig. 2: Manifold learning. a) points which are closer in the Euclidian space (e.g. B and C compared to A and B) might not be closer in the manifold space. Thus, learning a decision function directly in the Euclidian space will not take advantage of the specific structure of the manifold. b) a solution to this issue consists of mapping the input points into another space in which the Euclidian distance reflects the distance on the manifold. This can be done locally by considering the tangent space of the manifold at a given point (point A in the example).

4.2. LogitBoost for Riemannian Manifolds

One could use the covariance features directly as input to the LogitBoost algorithm described in the previous section. This would implicitly assume that we consider covariance matrices as 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. Thus the standard Euclidian distance of $\mathbb{R}^n$ may not reflect well the actual distance between matrices in the manifold, as illustrated in Fig. 2. To take advantage of the specific structure of the covariance space $\mathcal{M}$, Tuzel et al [25] introduced an appropriate mapping h from the Riemannian manifold into tangent spaces at points in the manifold, where the Euclidian distance between mapped points better corresponds to the distance between the original points in the manifold. This is illustrated in Fig. 2.

More specifically, the mapping $h : \mathcal{M} \to \mathbb{R}^n$ was defined as the transformation that maps a covariance matrix into the Euclidian tangent space at a point $\boldsymbol{\mu}_l$ of the manifold $\mathcal{M}$. We will denote by $\mathcal{T}_{\boldsymbol{\mu}_l}$ this tangent space. Formally, the mapping is defined by [19]:

$$h : \mathbf{X} \mapsto \mathbf{x} = h(\mathbf{X}) = \operatorname{vec}_{\boldsymbol{\mu}_l}\left(\log_{\boldsymbol{\mu}_l}(\mathbf{X})\right). \tag{7}$$

where the vec and log operators are defined matrix-wise by $\operatorname{vec}_{\mathbf{Z}}(\mathbf{y}) = \mathit{upper}(\mathbf{Z}^{-\frac{1}{2}}\mathbf{y}\mathbf{Z}^{-\frac{1}{2}})$ with *upper* denoting the vector form of the upper triangular matrix part, and

$$\log_{\mathbf{Z}}(\mathbf{Y}) = \mathbf{Z}^{\frac{1}{2}} \log(\mathbf{Z}^{-\frac{1}{2}}\mathbf{Y}\mathbf{Z}^{-\frac{1}{2}})\mathbf{Z}^{\frac{1}{2}}. \tag{8}$$

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

$$\log(\Sigma) = \mathbf{U}\log(\mathbf{D})\mathbf{U}^\top \text{ where } \Sigma = \mathbf{U}\mathbf{D}\mathbf{U}^\top \tag{9}$$

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 $\mathbf{D}$ [19, 25].

The mapping h is the composition of the log and vec operators. By definition, the $\log_{\boldsymbol{\mu}_l}$ operator is performing the mapping from the manifold $\mathcal{M}$ into the tangent space $\mathcal{T}_{\boldsymbol{\mu}_l}$ such that for each point $\mathbf{X} \in \mathcal{M}$ in the neighborhood of $\boldsymbol{\mu}_l$ the distance $d(\boldsymbol{\mu}_l, \mathbf{X})$ on the manifold[2] between $\boldsymbol{\mu}_l$ and $\mathbf{X}$ is given by the norm $\| \log_{\boldsymbol{\mu}_l}(\mathbf{X}) \|_{\boldsymbol{\mu}_l}$, where $\|.\|_{\boldsymbol{\mu}_l}$ denotes

[2] The distance on the manifold is called the geodesic distance, defined as the distance of the minimum length curve connecting two points.

the Riemannian norm in the tangent plane induced by the curve derivatives at the point $\boldsymbol{\mu}_l$ [19]. The vec operator allows to extract orthogonal coordinates of the points in the tangent space, i.e. the mapping $\text{vec}_{\boldsymbol{\mu}_l}$ relates the Riemannian norm on the tangent space to the standard canonical metric. In other words, after the mapping $\mathbf{x} = h(\mathbf{X})$ we have $d(\boldsymbol{\mu}_l, \mathbf{X}) = \|\mathbf{x}\|_2$ i.e. the geodesic distance is given by the standard Euclidian norm of $\mathbf{x}$.

Qualitatively, the mapping h allows to project points on the manifold into a vector space where the canonical Euclidian distance reflects the manifold geodesic distance. This property is however only valid in the neighborhood of the point $\boldsymbol{\mu}_l$, so the selection of this point is important. Intuitively, when building a weak classifier f_l, the point $\boldsymbol{\mu}_l$ should be as close as possible to the data points one wishes to classify. Thus, one natural way is to select this point as the weighted mean (in the Riemannian sense) of training examples $\mathbf{X}_i$, which is defined by:

$$\boldsymbol{\mu} = \arg\min_{\mathbf{Y} \in \mathcal{M}} \sum_{i=1}^{N} w_i d^2(\mathbf{X}_i, \mathbf{Y}). \tag{10}$$

where $d^2(\mathbf{X}, \mathbf{Y})$ is the (squared) geodesic distance between two points $\mathbf{X}$ and $\mathbf{Y}$ in the Riemannian space $\mathcal{M}$. Since the weights, defined in Eq. (6), are adjusted through boosting, at a given iteration l, this process will focus the classification on the current decision boundary. 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. The minimization of Eq. (10) can be conducted using an iterative procedure [19] provided in Appendix A.1.

In summary, when performing the LogitBoost training taking into account the Riemannian geometry, a weak classifier is defined as:

$$f_l(\mathbf{X}) = g_l(h(\mathbf{X})) = g_l\left(\text{vec}_{\boldsymbol{\mu}_l}\left(\log_{\boldsymbol{\mu}_l}(\mathbf{X})\right)\right) \tag{11}$$

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 video, due to the use of wide angle cameras, standing people in a given scene may appear with different slants in the image depending on their position in the image, as illustrated in Fig. 5. 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. 5. This transformation should thus help in obtaining better detection results while keeping the computation efficiency of integral images.

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. As the above mapping alone is not

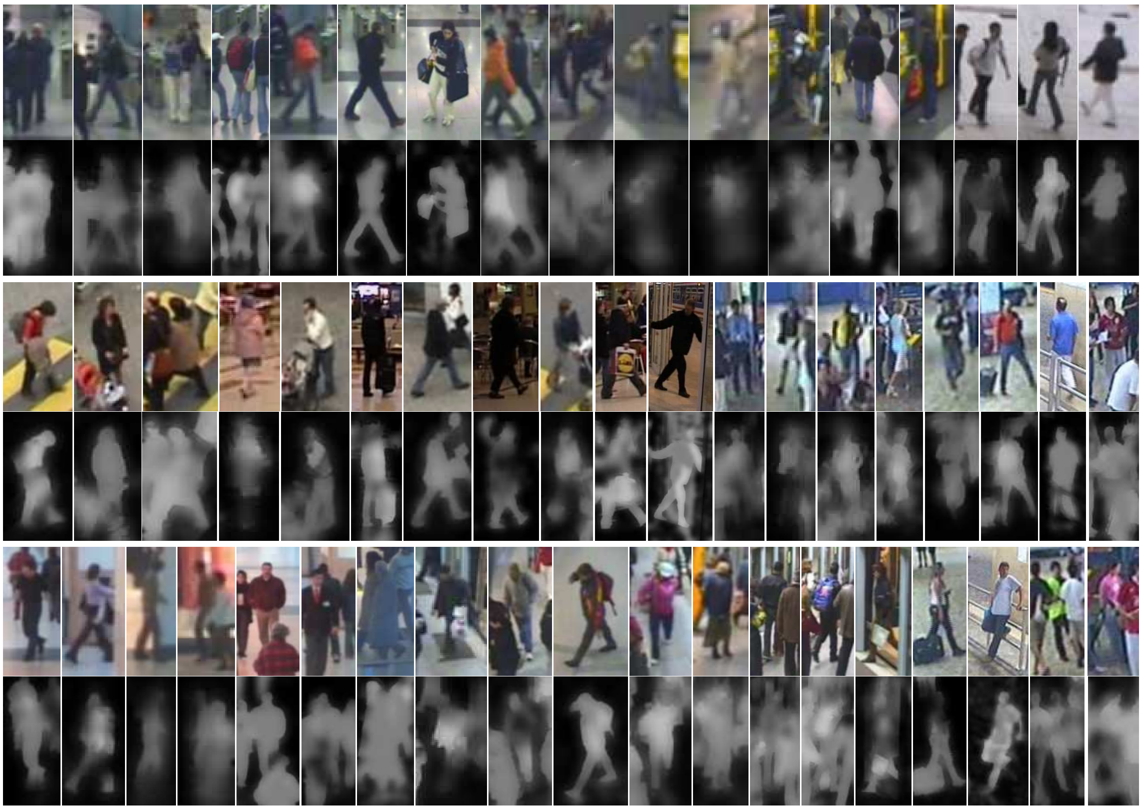

Fig. 3: Positive examples with corresponding foreground probability maps (light - high probability, dark - low probability).

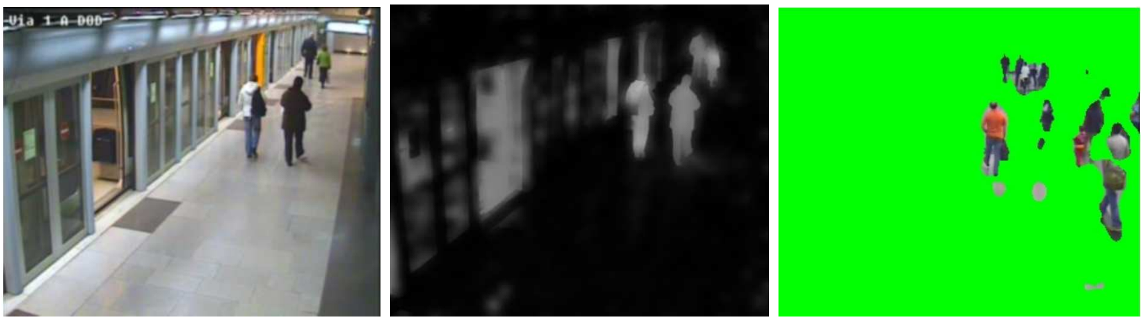

Fig. 4: Example of background subtraction processing, and issues for human detection. Foreground regions may contain other objects (e.g. cars for outdoor scenes). Specular reflection and cast shadow can generate background false alarms. People might only be partially visible (e.g. split into mutiple 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.

sufficient to fully define the homography, we must enforce additional constraints. In order 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 $\mathbf{x}_0$ of the image. This is meant that the first order approximation of the transform in the neighborhood of $\mathbf{x}_0$ should be a rotation rather than a general affine transform. An appropriate choice $\mathbf{x}_0$ to enforce such as constraint can be the image center. Technical details are given in Appendix A.3.

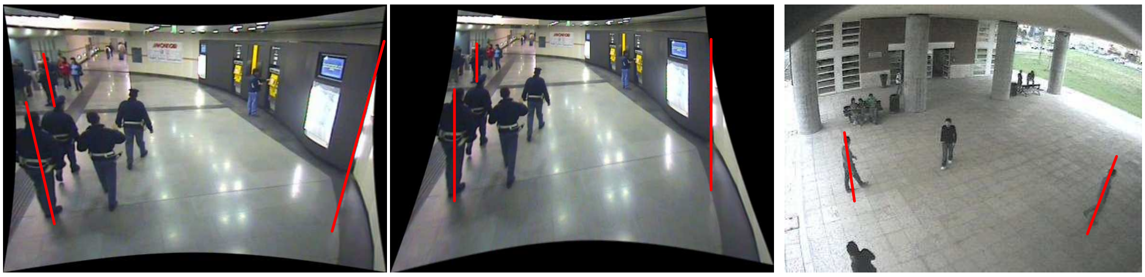

Fig. 5: Vertical vanishing point mapping. *Left:* after distorsion removal and before the mapping. We can observe people slant according to their position. *Central:* after the mapping to infinity. Bounding-boxes fit more closely the silhouette of people. *Right:* another example. See Fig. 14 to see the image after the mapping.

5.2. Integrating Foreground Information

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

$$\mathbf{H}(\mathbf{x}) = \left[\mathbf{x}\ |\mathbf{I}_x(\mathbf{x})|\ |\mathbf{I}_y(\mathbf{x})|\ \sqrt{\mathbf{I}_x^2(\mathbf{x})+\mathbf{I}_y^2(\mathbf{x})}\ \arctan\frac{|\mathbf{I}_y(\mathbf{x})|}{|\mathbf{I}_x(\mathbf{x})|}\ \mathbf{G}(\mathbf{x})\ \sqrt{\mathbf{G}_x^2(\mathbf{x})+\mathbf{G}_y^2(\mathbf{x})}\right]^\top \quad (12)$$

where $\mathbf{I}_x$, $\mathbf{I}_y$ and $\arctan\frac{|\mathbf{I}_y(\mathbf{x})|}{|\mathbf{I}_x(\mathbf{x})|}$ have the same meanings as in Eq. (1). $\mathbf{G}(\mathbf{x})$ 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 the features of Eq. (1) [25], the main difference is the use of the two foreground related measures instead of the second-order intensity derivatives $\mathbf{I}_{xx}$ and $\mathbf{I}_{yy}$ of the original images. We expect the former to be more informative in the context of video surveillance than the latter ones.

To extract these foreground features, we rely on the robust background subtraction technique described in [28]. In short, its main characteristics are the use of a multi-layer approach similar to the Mixture of Gaussian (MoG) [23], the use of Local Binary Pattern features as well as a perceptual distance in the color space to avoid the detection of shadows, and the use of hysteresis values to model the temporal dynamics of the mixture weights. Examples of intensity and foreground images are shown in Fig. 3.

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 some shape information, as can be noticed from the examples in Fig. 3. However, due to the small size of people in most input images, the foreground map is quite blurry in many cases. In addition, as people can remain static for a long period of time (and thus start being incorporated into the background), or wear clothes with very similar color to the background color, the foreground probability can be very low (see for instance the top right example in Fig. 3). If one would extract a binary foreground map (e.g. by thresholding), important information would be lost in these cases. Regions could be labelled as belonging to the background although they provide a small but non zero probability measure which can still suggest the presence of a foreground object. In addition, the binarization would introduce some shape artefacts, like for instance boundaries inside people's body, as in the Fig. 4 on the right. This is one of the reasons why we prefer to keep the real values of the foreground probability map as input feature. Another issue is that when multiple people are close and occluding each other partially, the resulting foreground 'patch' only contains partial information about the body shape. By using the covariance features, we expect

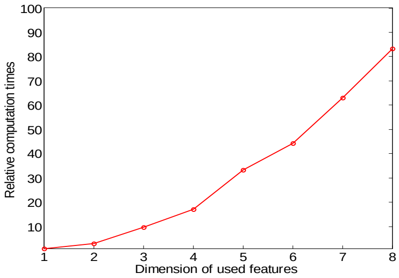

Fig. 6: Relative computation time of LogitBoost classifiers, for different feature sizes. Size one is taken as reference.

the algorithm to learn the correlation existing between the foreground shape and intensity shape information and be more robust against partial occlusion or false alarms due to cast shadow or the presence of other objects.

Finally, note that in the LogitBoost cascade context, the foreground feature will also help to quickly discard windows candidates which contain no or badly localized foreground information.

5.3. Weak Classifiers with Feature Subsets

Exploiting the covariance between all features in the Riemannian space has been shown to be powerful and working better than using the covariance features directly inside the LogitBoost algorithm [25]. The covariance mapping step is thus important. However, one main issue with this mapping is that it involves *costly* matrix operations at *run-time*, and more specifically it requires to compute a SVD eigenvalue decomposition to compute the logarithm of a matrix (cf Eq. (7), Eq. (8) and Eq. (9)). Thus, even embedded in cascade of LogitBoost classifiers, the detector can be quite slow at run-time. Of course, the load depends on the feature dimension, as illustrated in Fig. 6, which shows the relative computation time of a LogitBoost classifier composed of 10 weak classifiers built according to the approach described in Section 4. The computation increase is almost quadratic with respect to the feature dimensions, which mainly correspond to the complexity of the SVD decomposition algorithm. 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 could be at the cost of performance, since some information is obviously definitively lost. We propose instead to buld the weak classifiers relying on *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.

Another reason to use feature subsets is that there might not always exist consistent correlation between all features of the input space for a given class. In other words, the set of training data points in the high-dimensional Riemannian manifold might be quite complex, and the resulting mapping used for classification can be quite noisy. In this sense, using low-dimensional covariance matrices can be interpreted as a dimension reduction technique, and can appear to be more effective for classification.

Selecting the feature subsets:

As explained later in Subsection 6.1, during training, at each iteration of the boosting algorithm, several weak classifiers corresponding to randomly picked subwindows are tested

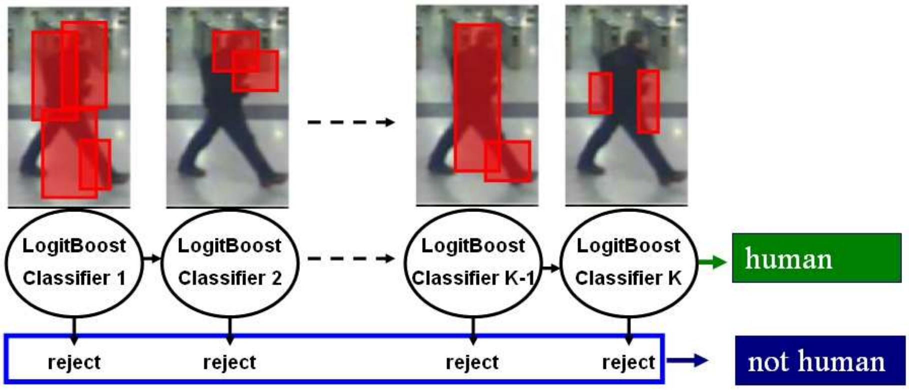

Fig. 7: Cascade of LogitBoost Classifiers. Each LogitBoost classifier is composed of weak classifiers relying on feature subset covariances computed from subwindows.

and the best one is kept. When using the covariance between all image features, 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 very 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 which are most likely to provide good results for classification. This approach, described in appendix A.2, provides a better way of selecting good m-subset features than uniform random selection, and saves a significant amount of time in training.

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 of these features, 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 in the test window, 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 for training and detection in the LogitBoost algorithm. Since these features directly lie in a d-dimensional Euclidean space 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 (7) that projects the normalized covariance $\mathbf{C}'_r$ features into the tangent space at the weighted-mean matrix, as explained in Section 4. In other words, we use the concatenation of the mapped covariance features with the normalized mean features[3] as input to the linear function g_l used in the LogitBoost classifier

[3] Note that only the feature means of the subset for which we compute the covariance are used in the weak classifier.

6. Algorithm Description

In this section, we describe the technical details about the cascade training and our proposed post-processing approach for detection.

6.1. Training the Cascade

Algorithm 1 Training one Logit-Boost Classifier in the cascade.

Input: Training set of positive and negative examples $\{\mathbf{Q}_i, y_i\}_{i\ldots N}$, where $\mathbf{Q}_i$ are the image examples, and $y_i \in \{0, 1\}$ are the class label.

Parameters for Stop Condition:

$f_{\max}$: maximum acceptable false positive rate per cascade level

$d_{\min}$: minimum acceptable detection per cascade level

th_b: margin constraint threshold between the positive examples and the decision boundary

```
• Initialization: $F(\mathbf{Q}_i) = 0$ and $p(\mathbf{Q}_i) = 0.5$
• Repeat for weak classifiers $l = 1, \ldots, L_{max}$
    ∗ Compute the response values and weights, $z_i = \frac{y_i - p(\mathbf{Q}_i)}{p(\mathbf{Q}_i)(1-p(\mathbf{Q}_i))}$, $w_i = p(\mathbf{Q}_i)(1 - p(\mathbf{Q}_i))$
    ∗ Compute the weak classifier $l$, $f_l = \mathit{TrainAndSelect}(\{\mathbf{Q}_i, y_i, z_i, w_i\}_{i\ldots N}, F)$.
    ∗ Update $F(\mathbf{Q}_i) \leftarrow F(\mathbf{Q}_i) + \frac{1}{2} f_l(\mathbf{Q}_i)$ and $p(\mathbf{Q}_i) = e^{F(\mathbf{Q}_i)} / \big(e^{F(\mathbf{Q}_i)} + e^{-F(\mathbf{Q}_i)}\big)$
    ∗ Find $\mathbf{Q}_p$ and $\mathbf{Q}_n$ where $\mathbf{Q}_p$ is the positive example that has the $(d_{\min} \times N_p)$-th largest
      probability among all the positive examples and $\mathbf{Q}_n$ is the negative example that has
      the $(f_{\max} \times N_n)$-th smallest probability among all the negative examples.
    ∗ If $p(\mathbf{Q}_p) > p(\mathbf{Q}_n) + th_b$, exit repeating and set the rejection threshold of $k$-th cascade
      level to $p(\mathbf{Q}_n)$
```

The human detector was implemented within a cascade of LogitBoost rejection classifiers framework, as illustrated in Fig. 7. The procedure to train a LogitBoost classifier is the same for each level of the cascade. Only the training set of images differs. Algorithm 1 describes the different steps for training the LogitBoost classifier at a cascade level. It follows the approach described in Section 4.1. 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. (4) is actually added as f_l to form the decision function F. These weak classifiers are trained to account for the Riemannian geometry, as explained in Section 4.2. The procedure summarizing this training process is given by Algorithm 2. Below, we provide some explanations and details on the different steps of the algorithms.

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

Algorithm 2 Training and selection of weak classifiers based on covariances of m-dimensional feature subsets. $f^* = \mathit{TrainAndSelect}(\{\mathbf{Q}_i, y_i, z_i, w_i,\}_{i\ldots N}, F)$

Input: $\{\mathbf{Q}_i, y_i, z_i, w_i\}_{i\ldots N}$ 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, characterized by a subwindow, a feature subset, its mapping parameters (the point $\boldsymbol{\mu}$ at which the tangent space is defined) and the coefficients of the linear regressor function 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 matrix $\mathbf{X}_i$ and the normalized mean vector $\mathbf{m}_i$ ($\mathbf{m}_i = \varnothing$ if we don't integrate mean feature) of the subwindow $r$ from the example $\mathbf{Q}_i$
    - Compute weighted mean $\boldsymbol{\mu}_{r,s}$ of all the data points $\{\mathbf{X}_i\}_{i=1\ldots N}$.
    - 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 of $z_i$ to $(\mathbf{x}_i, \mathbf{m}_i)$ using weights $w_i$ according to Eq. (5).
    - Define $F_{r,s}(\mathbf{Q}_i) \leftarrow F(\mathbf{Q}_i) + \frac{1}{2} f_{r,s}(\mathbf{Q}_i)$ and $p(\mathbf{Q}_i) = e^{F_{r,s}(\mathbf{Q}_i)} / \big(e^{F_{r,s}(\mathbf{Q}_i)} + e^{-F_{r,s}(\mathbf{Q}_i)}\big)$
    - Compute $\mathsf{L}_{r,s}$, the negative binomial log-likelihood of the data using Eq. (4).
• Return $f^*$, the weak classifier for which $\mathsf{L}_{r,s}$ is the minimum.
```

between the probability of the positive examples during training and the classifier decision boundary. More precisely, 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. (3). 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 numbers of positive and negative examples used for training. Weak classifiers are added to the cascade level k 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 $p_k(\mathbf{Q}) \leq \tau_k$ where the threshold τ_k is selected as equal to $p_k(\mathbf{Q}_n)$ 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. In a similar way, the negative examples were selected as the false positive examples of the $k - 1^{th}$ detector applied to a collection of negative training images containing no positive data until N_n examples were collected.

Training and selecting weak classifiers: As mentioned above, to obtain the weak classifier at iteration l of the LogitBoost learning, a collection of weak classifiers are actually trained

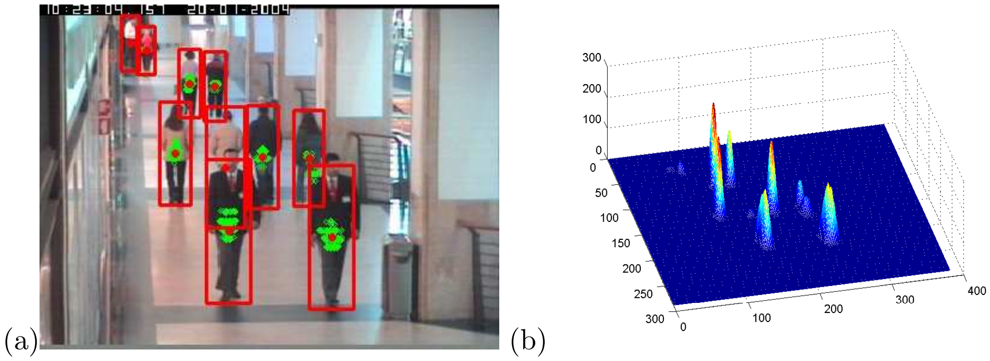

Fig. 8: Example of post-processing on detection outputs: (a) Green dots show all the window centers of the positive detection outputs. Red dots are the final detection extrema found via post-processing. (b) The smoothed reliability image corresponding to the image in (a).

and the best one is kept, as described in Algorithm 2. The collection of tested classifiers $\{f_{r,s}\}_{r=1..N_w,s=1..N_s}$ 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 according to the scheme described in Section 5.3 and detailed in Appendix A.2.

6.2. Post-processing

To detect people in a test image, the trained binary detector is applied to a large number of windows of different positions and sizes. Usually, for one actual real person in the image, several positive detections are obtained and a procedure has to be defined to merge these detections, as shown in Fig. 8(a). In this paper, we propose a simple and effective method for this task. Let $\mathcal{X}_p = \{\mathbf{Q}_p\}_{p=1}^{P}$ be the set of P positive detection outputs and $\mathbf{x}_p^c$ be the image center of the bounding box of the detection output $\mathbf{Q}_p$. For each detection output $\mathbf{Q}_p$, we define its reliability as:

$$d_{rel}(\mathbf{Q}_p) = \sum_{k=1}^{K} \left((f_{\max})^{K-k} \times p_k(\mathbf{Q}_p) \right). \tag{13}$$

Then, we build a reliability image $\mathbf{D}_{rel}$ by associating the reliability scores of detected windows to their center, i.e. by setting $\mathbf{D}_{rel}(\mathbf{x}_p^c) = d_{rel}(\mathbf{Q}_p)$. For positions where no detection was found, the value is 0. Also, in the cases when two detected windows (with different sizes) have the same center, the maximum of their reliability is used. Then, we smooth the reliability image 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. 8(b). All local maxima points on the smoothed image are then considered as possible detection results. Their associated window sizes are obtained as the weighted means (where the weights depends on reliability and distance to the extrema position) of the window sizes of the positive detections found in their neighborhood. A last simple constraint is used to further filter the detection results. 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 the process starts with the less reliable detections.

7. Experimental Results

In this section, we report the results of our method applied to several databases. Different experiments were conducted to evaluate the different aspects of our method. In Section 7.1, we present a general and complete evaluation on our target applications: the detection of people in surveillance data acquired from stationary cameras. In Section 7.2, we report the results of the detection algorithm on a small video database to illustrate the benefit of the slant removal step. Finally, in Section 7.3, we used the INRIA database of still images to compare our approach against previous works [4, 25], and more precisely, to evaluate the impact of the use of feature subsets and mean features on the performance.

7.1. Human Detection in Video Sequences

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

7.1.1. 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, the PETS data, and several metro station cameras. Around 10000 positive examples were extracted from these 15 video sequences. Typical examples are shown in Figure 3. Note that in these examples, there are large variations of appearances, poses, camera viewpoints, the presence of luggage or trolleys, partial occlusions, and variability in the extracted foreground images. Negative examples were obtained by: (i) collecting 1000 still images without people and coupling them with inconsistent foreground detection results; (ii) cropping about 10000 regions from the collected video data which don't contain full human bodies; (iii) bootstrapping, i.e. by collecting more negative samples which 'look like' people after each LogitBoost training in the cascade, i.e. by applying the current detector on training data without people, 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 10 of the above sequences, but not used for training and containing different people, 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.

7.1.2. 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. Positive detections were then filtered out by keeping local maxima of these detection outputs as described in Section 6.2. 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 [4, 17, 25], 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

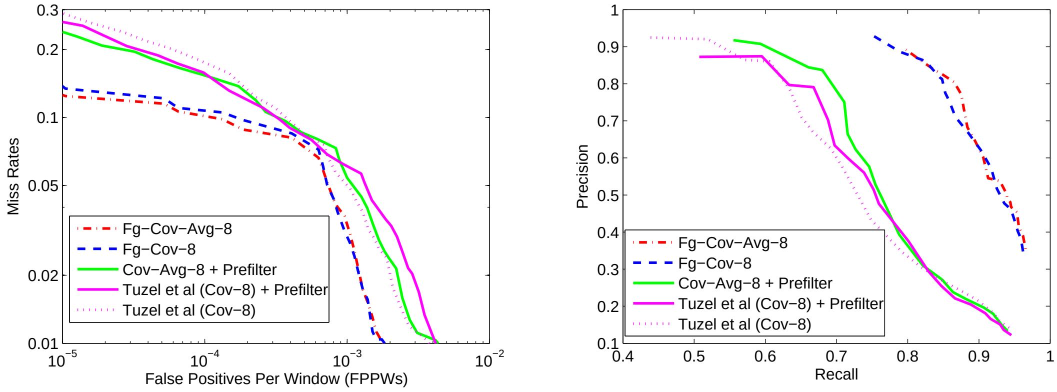

Fig. 9: The performance of different approaches with 8-dimensional features. Left, Miss rate vs false positive rate. Right, precision-recall curves, which also take into account the post-processing step.

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.

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 [6, 20]. RP curves integrates the post-processing steps, i.e. how to combine several raw detector positive output into one or several detected humans. Thus, detectors with similar miss-rate for the same FPPW value may exhibit different behaviours in the RP curves: detectors which provides multiple but spatially consistent detections will tend to produce overall less false alarms at the object level than detectors which spread their detection over multiple locations. 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.

7.1.3. Results

We consider the method of Tuzel et al [25] as our baseline. Three main improvements to this method were made to handle video data: integration of foreground probability features, use of mean (average) features in addition to covariance, and selection of feature subsets. 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 defined in Eq. (12), which integrates intensity and foreground information, as well as the average (mean) of these features. When foreground features are not used, we used the 8-dimensional features defined in Eq. (1), and used in [25].

Foreground features. In the first experiment, whose results are shown in Fig. 9, we trained four detectors with/without the use of foreground information and average features. In addition, to allow a fair comparison, a prefilter based on background subtraction is also applied with the baseline approach [25]: only windows which contain a sufficient percentage

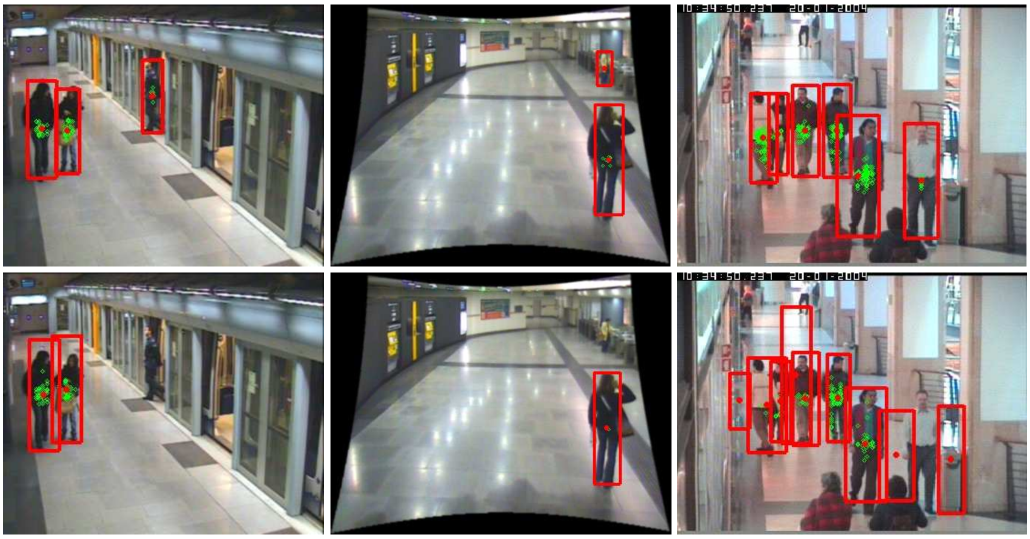

Fig. 10: Detection results of three images by the detector *Fg-Cov-8* integrating foreground information into covariance matrices (Top) and Tuzel et al's method with foreground prefilter (Bottom).

of foreground pixels are tested. A percentage of 20% was used. Thresholds above this value were reducing the performance, by starting rejecting more true positive than false alarms.

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 precision of 0.9, only around 60% of the people are actually detected with [25] and the prefilter, while when using the foreground features, around 80% of the people are detected. Besides, we can see that the use of the mean features improves the results almost systematically, but usually not significantly.

Surprisingly, we can observe that the foreground prefilter scheme does not improve the detection performance too much. Indeed, there are two problems with this approach: foreground information is used only to reject detection, but not to accumulate evidence of the presence of a person. Thus, when the static image is not sufficient to differentiate a person from the background clutter, foreground does not help to take the decision. This is the case of the two images on the left of Fig. 10. On the opposite, the consistency of foreground and static observations allows our approach to succeed. The second related issue is that the percentage of pixels inside the window is only a crude indicator for rejection, insufficient to properly distinguish between false alarms and true detection in the presence of cluttered foreground due to the presence of multiple people and shadows, as in the right image of Fig. 10. Overall, we observe that directly integrating foreground information into the detector *Fg-Cov-8* generates better results, especially for people in cluttered background, small people with few texture information, and people occluded by other people.

Performance of feature subsets. To investigate the impact of using feature subsets of different sizes, 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. 11(a) shows the RP curves that we obtained, with *Fg-Cov-Avg-8* for comparison. Interestingly, while the use

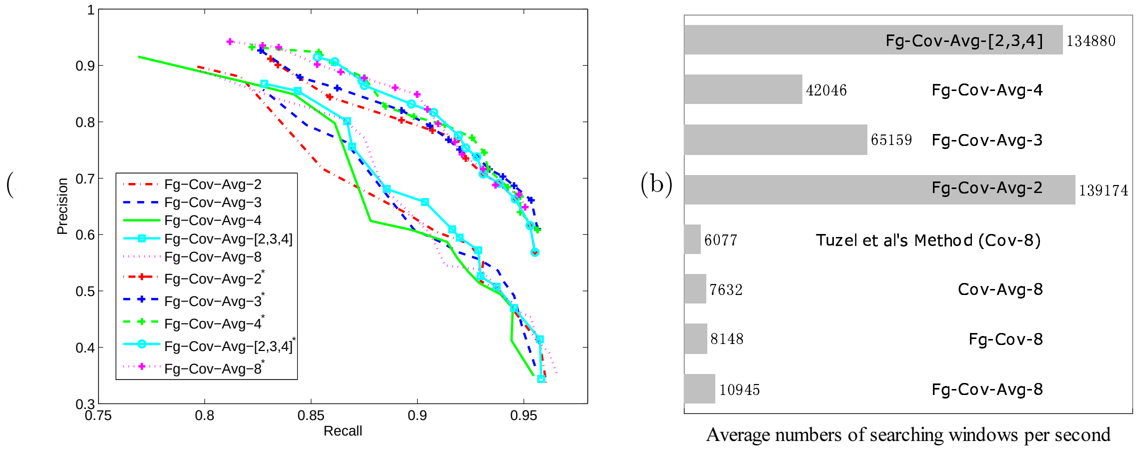

Fig. 11: (a) Performance of different approaches with (labels with a superscript $*$) or without a ground-plane geometrical constraint; (b) Average numbers of tested windows (per second)

.

of subsets of size 2 are slightly lower than *Fg-Cov-Avg-8* for some recall values, we observe that overall, 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. This confirms our hypothesis that the selection of the most consistent correlation terms between features is enough to achieve good detection performance. The Figures 12 provide statistics about the frequency of the selected feature types in subsets. Interestingly, for subsets of size 2, we can notice that the image gradient orientation is the dominant feature (which confirms the importance of edge orientation as in HOG features), and often selected along with foreground probability value and gradient. This further demonstrates the interest of exploiting the joint correlation of appearance and foreground features in the training. In addition, the use of the average features tends to reduce the selection of the x and y components[4]. Finally, we can notice that using larger subsets tends to smooth out the types of selected features, since features might be selected by a weak classifier even if they are not really discriminative and do not contribute to the fit.

Computational speed. The level of performance achieved by our detectors comes with a significant gain in processing speed w.r.t. our baseline. The computational complexity was evaluated by applying the different detectors to the test data and measuring the average numbers of windows per second that each detector can process[5]. The same computer was used in all the cases. The results are shown in Fig. 11(b) . 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 [25] with *Cov-Avg-8*). Secondly, as could be

[4] This effect might be explained by the fact that since the average of these features are not input dependent, they were not used when computing the linear fit to the data, ultimately introducing a bias towards weak classifiers with more variables for the fit.

[5] The numbers only take into account the detection time. All other parts, e.g. image scanning or post-processing, are the same for all detectors.

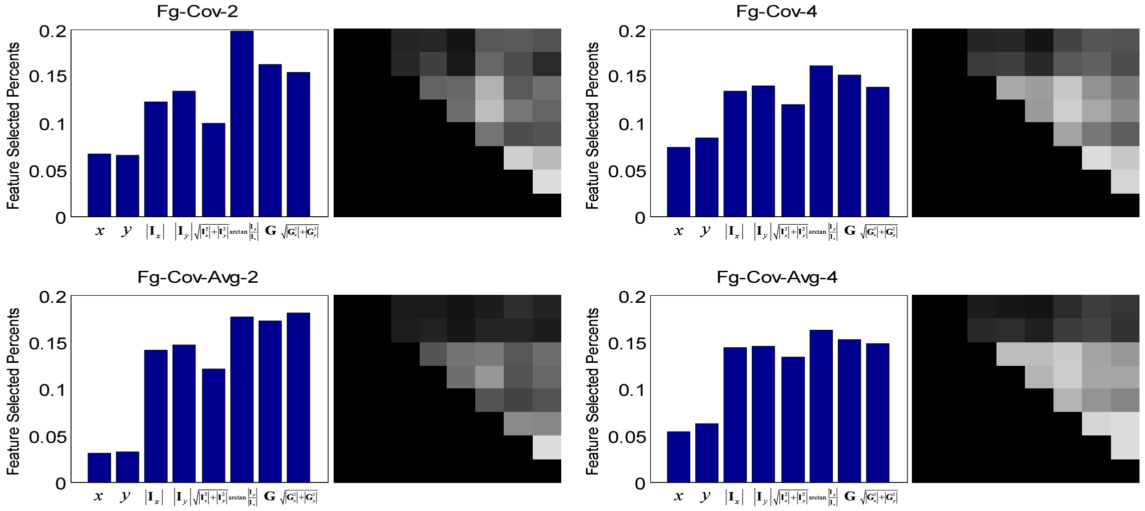

Fig. 12: Percentage of times a given image feature (left: blue bars) and a given image feature pair (right: light, high percentage; dark, low percentage) is selected as part of a feature subset for classification in the 30 cascade levels of four detectors.

expected, in addition to improving performance, the use of the foreground features also helps in increasing 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 detector *Fg-Cov-Avg-2* runs around 13 times faster than *Fg-Cov-Avg-8* (and more than 20 times faster than [25]). The combined detector *Fg-Cov-Avg-[2,3,4]* achieves a similar speed while slightly improving the performance (see Fig. 11) on this dataset. We can apply these two detectors to videos of size 384x288 (e.g. CAVIAR data) and process around 5 frames/sec. Indeed, with our new approach, most of the time is now spent in the computation of the image features, like our adaptive background subtraction process and the integral images, rather than in the detection part itself.

Exploiting 3D geometry 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. Results are shown in Fig. 11(a), and show a consistent gain due to the removal of some of the false positives windows.

Result illustrations. Fig. 13 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 post-processing step. Despite the large variability of appearances, poses and view points and the partial occlusions, and the overall small people size, there are only few false positive and false 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.

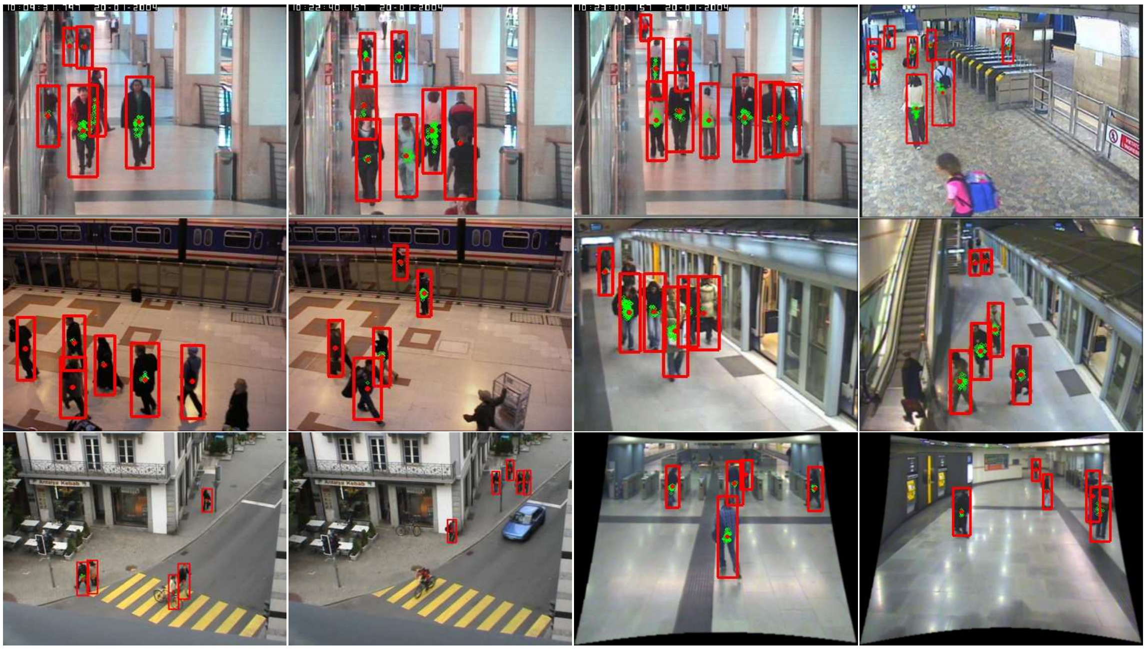

Fig. 13: Detection examples. Green dots show all the detection results. Red dots are final detection results via post-processing and the bounding boxes are average detection window sizes. Note that as the detector was designed to detect full standing people, people only appearing in part are not detected.

7.2. Test on Slant Removal Data

In surveillance scenarios, when using cameras with short focus lenses, people may appear slanted in the image. To suppress this effect, we can first apply the homography transform introduced in 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 from the 3 video streams shown in Fig. 14 representing a typical case of the issue. More precisely, 29 images were extracted and annotated, leading to a database of 89 people in the ground truth.

We applied the detector *Fg-Cov-Avg-[2,3,4]* on the test data with and without the vertical vanishing point geometrical correction. Results on three typical examples are shown in Fig. 14, and clearly illustrate the benefits of the approach. These benefits are confirmed by the performance curves plotted in Fig. 15 from which we observe that the detection performance on slant-removed images are always better than those obtained on the original images.

7.3. Test on INRIA database

To further evaluate the efficiency of the use of feature subsets for building detectors using covariance features, we performed experiments on the INRIA human database [4, 25], which is a still image database without foreground information. The database contains 1774 human positive examples and 1671 negative images without people. We followed the same protocol as described in [4, 25]. We used 1208 human positive examples (and their reflections w.r.t. a central vertical axis) and 1218 negative images for training. For each

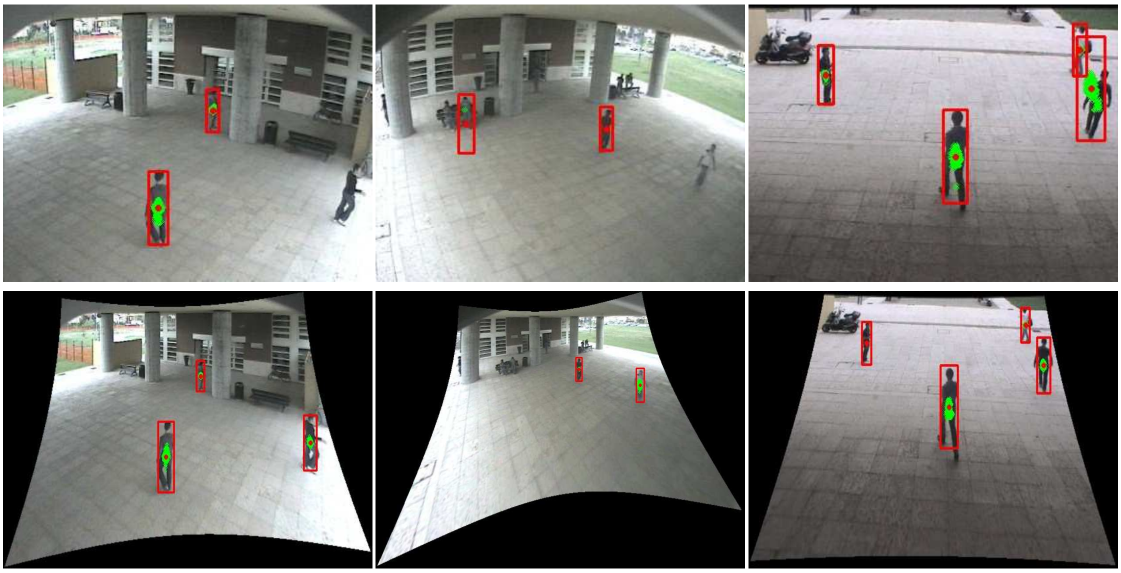

Fig. 14: Detection results on original images (Top) and warped images with infinite vertical vanishing point (Bottom).

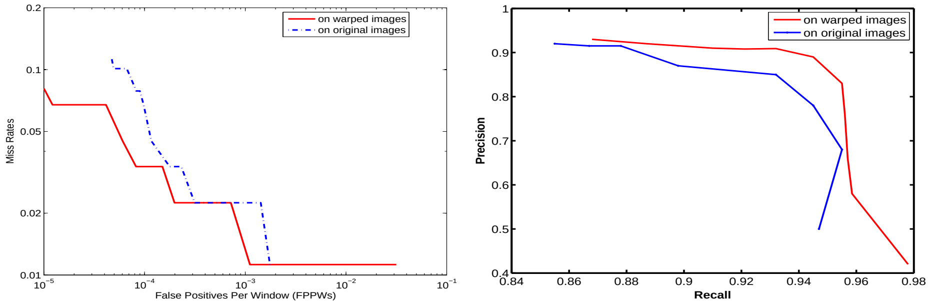

Fig. 15: Detection performance on data with slanted people.

cascade level, the Logitboost algorithm was trained using all the positive examples (hence $N_p = 2 * 1208 = 2416$) and $N_n = 10000$ negative examples generated by boostrapping after each cascade level. The rest of the data (566 positive examples and 453 negative images) was used for testing, and the DET curve were constructed [4, 25], as explained in Subsection 7.1.2. Detection on the INRIA database is challenging since it includes human with a wide range of variations in pose, clothing, illumination, background and partial occlusions. To our knowledge, Tuzel et al. [25] obtained the best detection results on this database, and outperformed the methods of Dalal & Triggs [4] and Zhu et al. [32].

In total, we trained 10 detectors, 5 with and 5 without the mean features. The results are shown in Figs. 16 and 17 (to allow better comparison). The first observation is that our implementation of Tuzel et al's method led to very similar detection performance as described in [25] (e.g. with a miss-rate of 7.5% for a 10^{-4} FPPW rate vs 6.8% reported in [25]). Secondly, unlike in the video case, the use of low-dimensional subset features consistently lead to better results than the use of the full covariance features, (e.g. compare *Cov-2* with *Cov-8* [25]). For instance, at 10^{-4} FPPW, we obtain a 4% miss-rate when using subsets of size 2. This result might be explained by the smaller amount of positive training data (around 2400 here, vs a pool of 10000 examples in the video case) available which makes the training of the weak-classifier in the 33 dimensional full covariance space noisier

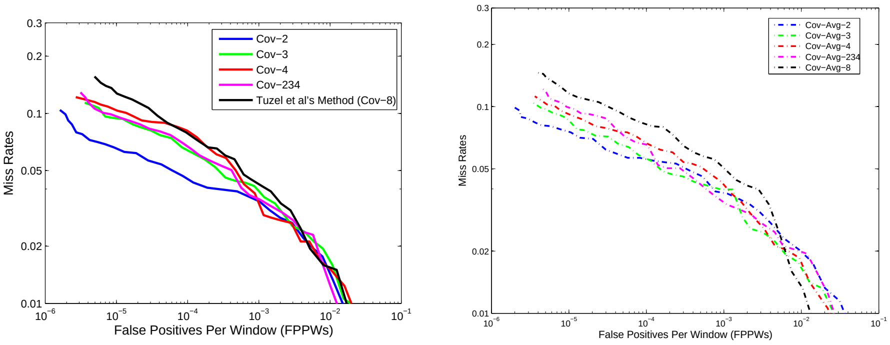

Fig. 16: The performance of five detectors without mean features (Left) and with mean features (Right) on INRIA database.

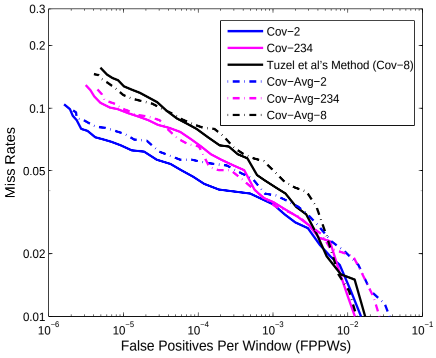

Fig. 17: Comparison of detectors with/without mean features on INRIA database.

than when using subsets. Thus, rather than forcing the use of all covariance coefficients, the selection of the most consistent ones as we propose can be 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*. Still, as in the video case, the use of mean features increases the detection speeds. This is illustrated in Fig. 18 which displays the number of weak classifiers at each cascade level of four detectors. As a general trend, there are very few classifiers at early cascade stages, where most of the negative examples are rejected. We can observe that there are fewer classifiers in detectors using mean features than in the detectors without using mean features. Overall, as with the video case, the feature subset detectors are much faster than the detectors based on the full feature set. The *Cov-Avg-2* can process around 15 times more windows than *Cov-8*, while performing better.

Finally, Fig. 19 shows several detection results on challenging images.

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 [25].

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,

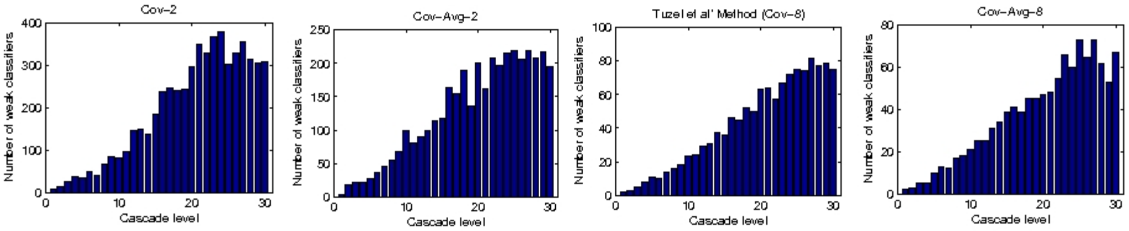

Fig. 18: The number of weak classifiers of each cascade level in four detectors (2 without mean features and 2 with mean features), learned from INRIA database.

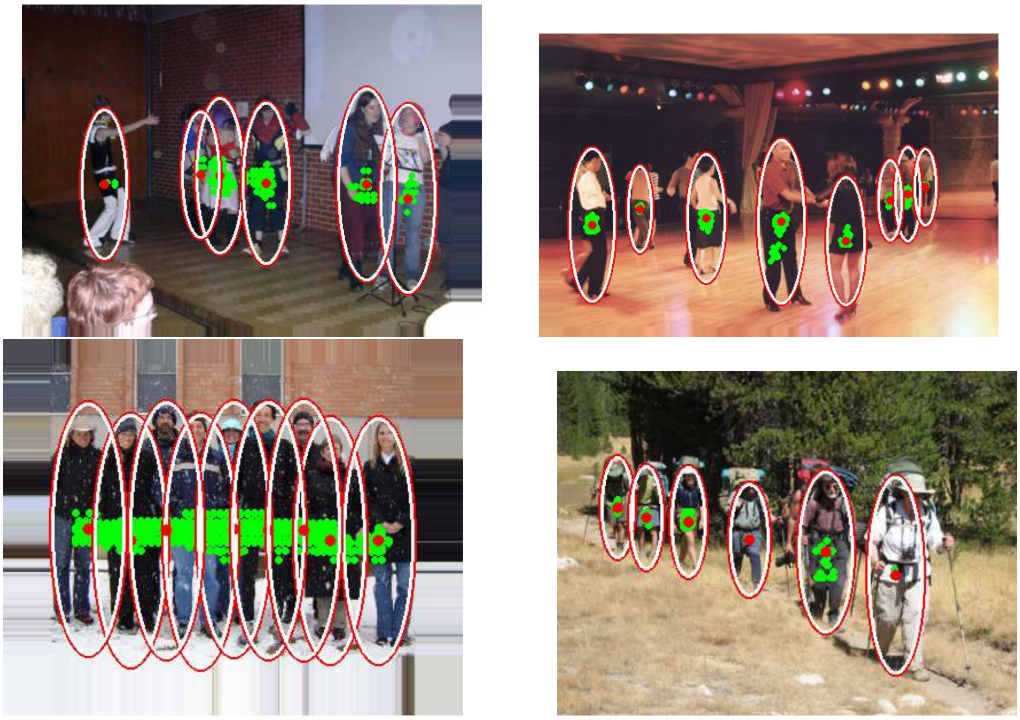

Fig. 19: Detection examples on still images.

allowing to improve the detection performance while keeping the computational efficiency due to the use of integral images. Second, by learning the consistent correlation 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 in the background. In particular, the approach showed the mutual benefit of using both types of feature, by particularly improving detection in cluttered background (where static features are not enough to validate the detection of a person) and in cluttered foreground, e.g. when multiple people occlude each other and thus the shape foreground features are quite noisy. Finally, to build the weak classifiers of the LogitBoost 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 around 15 to 22 times w.r.t. using the covariance between all features, while providing equivalent or sometimes even better performance. These novelties resulted in a near realtime detector that performs accurately, as shown by our experiments on real, large and challenging datasets.

There are several areas for future work. Although our algorithm is very fast, the bottleneck to achieve full realtime speed for large images lies in the number of integral images which need to be computed. Reducing this number can be obtained by simply using less image features, although this might be at the cost of significant performance decrease. A better alternative, possible only when using our approach 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 on different clusters of the data if no pose labels are available), or learn classification tree, as done for instance in multi-view face detectors [14].

Currently, 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 learn their response in different occlusion configuration. We plan to investigate this approach in the future.

9. Acknowledgement

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

References

[1] J. Begard, N. Allezard, and P. Sayd. Real-time humans detection in urban scenes. In *BMCV*, 2007.

[2] C. Beleznai, B. Fruhstuck, and H. Bischof. Human detection in groups using fast mean shift procedure. In *IEEE Int. Conf. on Image Processing*, volume 1, pages 349–352, 2004.

[3] 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 *11th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, Bejing, October 2008.

[4] 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.

[5] 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.

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

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

[8] Francois Fleuret, Jerome Berclaz, Richard Lengagne, and Pascal Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(2):267–282, 2008.

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

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

[11] 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.

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

[13] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 878–885vol.1, 20-25 June 2005.

[14] S. Li and Z. Zhang. Floatboost learning and statistical face detection. *IEEE PAMI*, 26:1112–1123, 2004.

[15] 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.

[16] 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.

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

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

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

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

[21] Hedvig Sidenbladh. Detecting human motion with support vector machines. In *IEEE Int. Conf. on Pattern Recognition*, volume 2, pages 188–191, 2004.

[22] K. Smith, D. Gatica-Perez, and J.M. Odobez. Using particles to track varying numbers of objects. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, San Diego, June 2005.

[23] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, volume 2, pages 246–252, 1999.

[24] Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In *Europe Conf. Comp. Vision (ECCV)*, volume 2, pages 589–600, 2006.

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

[26] 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.

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

[28] J. Yao and J-M. Odobez. Multi-layer background subtraction based on color and texture. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Workshop on Visual Surveillance (CVPR-VS)*, pages 1–8, June 2007.

[29] T. Zhao and R. Nevatia. Bayesian human segmentation in crowded situations. In *IEEE CVPR*, 2003.

[30] Tao Zhao, Ram Nevatia, and Bo Wu. Segmentation and tracking of multiple humans in crowded environments. *IEEE Trans. Pattern Anal. Machine Intell.*, 30(7):1198–1211, 2008.

[31] J. Zhou and J. Hoang. Real time robust human detection and tracking system. In *IEEE CVPR workshop on Object Tracking and Classification in and Beyond the Visible Spectrum*, 2005.

[32] Q. Zhu, M.C. Yeh, K.T. Cheng, and S. Avidan. Fast human detection using a cascade of histograms of oriented gradients. In *IEEE Conf. Comp. Vision & Pattern Recognition (CVPR)*, number II, pages 1491–1498, 2006.

A. Appendix

A.1. Computing the weighted mean of points in the Riemannian manifold

Let $\{\mathbf{X}_i, w_i\}_{i=1\ldots N}$ be a set of points in the Riemannian manifold $\mathcal{M}$. The weighted mean of these points in the manifold is the point on $\mathcal{M}$ which minimizes the weighted sum of squared geodesic distances:

$$\boldsymbol{\mu} = \arg\min_{\mathbf{Y}\in\mathcal{M}} \sum_{i=1}^{N} w_i d^2(\mathbf{X}_i, \mathbf{Y}). \tag{14}$$

The minimization in Eq. (10) can be conducted using the following iterative gradient descent procedure described in [19], obtained by differentiating the error w.r.t. $\mathbf{Y}$:

$$\boldsymbol{\mu}^{n+1} = \exp_{\boldsymbol{\mu}^n}\left(\frac{1}{\sum_{i=1}^{N} w_i}\sum_{i=1}^{N} w_i \log_{\boldsymbol{\mu}^n}(\mathbf{X}_i)\right) \tag{15}$$

where the $\exp_{\mathbf{Z}}$ operator is the inverse mapping of the log operator and maps a point on a tangent space $\mathcal{T}_{\mathbf{Z}}$ at $\mathbf{Z}$ into the manifold. It is defined by [19]:

$$\exp_{\mathbf{Z}}(\mathbf{y}) = \mathbf{Z}^{\frac{1}{2}} \exp(\mathbf{Z}^{-\frac{1}{2}}\mathbf{y}\mathbf{Z}^{-\frac{1}{2}})\mathbf{Z}^{\frac{1}{2}}$$

where the matrix exponential $\exp(\Sigma)$ for symmetric matrices Σ is defined similarly to the log operator as:

$$\exp(\Sigma) = \mathbf{U}\exp(\mathbf{D})\mathbf{U}^{\top}$$

(with $\Sigma = \mathbf{UDU}$ cf Eq. (9)) and $\exp(\mathbf{D})$ is a diagonal matrix whose entries are the exponential of the diagonal terms of $\mathbf{D}$. The method in Eq. (15) iterates first order approximation to the mean in the tangent space: training points are mapped into the tangent space of the current mean, where an approximate weighted mean can be computed in the usual sense, and then the obtained result is mapped back into the manifold.

A.2. Selection of the feature subsets for classification

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 subsets of size m. At each step of the LogitBoost algorithm, we would like to find the best subwindow-subset couple $(r^\star, i^\star)$ that provides the minimum negative binomial log-likelihood, i.e.: $(r^\star, i^\star) = \arg\min_{r,i} \mathsf{L}_r(\mathrm{S}_{m,i})$, where $\mathsf{L}_r(\mathrm{S}_{m,i})$ denotes the negative binomial log-likelihood defined in Eq. (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.

We thus adopted the following approach to test the subsets which have higher probability to correspond to the best one. First we fully test all the 2-subsets, whose corresponding weak classifiers can be trained very fast, and obtain the set $\{\mathsf{L}_r(\mathrm{S}_{2,i})\}_{i=1\dots C_2^n}$ 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 $m > 2$, we compute a *substitute value* $\tilde{\mathsf{L}}_r(\mathrm{S}_{m,i})$ for negative binomial log-likelihood defined as $\tilde{\mathsf{L}}_r(\mathrm{S}_{m,i}) = \sum_{\mathrm{S}_{2,s}\in\mathrm{S}_{m,i}} \mathsf{L}_r(\mathrm{S}_{2,s})$ 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 subset of higher dimension. We examined this principle using the following experiments. We trained a human detector with 20 cascade levels consisting of weak classifiers learned from m-subsets. For each tested subwindow and for all the m-subsets, we computed L and their substitute values $\tilde{\mathsf{L}}$, to compare the ranks according to the ground truth L and those according to the substitute $\tilde{\mathsf{L}}$. Fig. 20 shows the obtained results for $m = 3$ and 4. The different curves plot the probability that within the first q values of $\tilde{\mathsf{L}}$ (horizontal axis), we find at least one of the k best subset (curve tag, k=1,3,5, or 10) according to the ground truth L, or in mathematical form: $P(\exists i|\mathrm{Rank}(\mathsf{L}_r(\mathrm{S}_{m,i})) \leq k \text{ and } \mathrm{Rank}(\tilde{\mathsf{L}}_r(\mathrm{S}_{m,i})) \leq q)$. As can be seen, by selecting $q = 8$ subsets (out of 56) for $m = 3$ and $q = 12$ for $m = 4$ (out of 70) using $\tilde{\mathsf{L}}$, we can see that the chances that one of them is actually one of the top 3 best are higher than 94%.

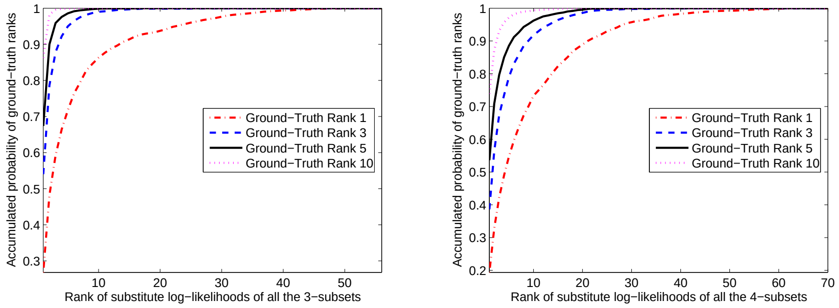

Fig. 20: Grouth-truth ranks of $\{\mathsf{L}_r(\mathrm{S}_{m,i})\}$ vs. approximated ranks of $\{\tilde{\mathsf{L}}_r(\mathrm{S}_{m,i})\}$, for $m = 3$ and 4.

A.3. Homography computation for slant removal

The objective is to find a 2D homography $\mathbf{K}_\perp$ that maps the image vertical vanishing point $\mathbf{v}_\perp = (x_\perp, y_\perp, 1)^\top$ to a vanishing point at infinity $(0, y_\infty, 0)^\top$, while reducing distorsion around an origin point $\mathbf{x}_0$. For the moment, assume that $\mathbf{x}_0$ is the coordinate system origin, and that $\mathbf{v}_\perp$ is located on the y axis, i.e. $\mathbf{v}_\perp = (0, y_\infty, 1)^\top$. Then we can consider the homography $\mathbf{K_G}$ which maps the vertical vanishing point to a point at infinity $(0, y_\infty, 0)^\top$ as required, and maps a 2D point (x, y) into a 2D point (x', y') according to:

$$\mathbf{K_G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & \frac{-1}{y_\infty} & 1 \end{bmatrix}, \begin{bmatrix} x' \\ y' \end{bmatrix} = \frac{y_\infty}{(y_\infty - y)} \begin{bmatrix} x \\ y \end{bmatrix}, \frac{\partial(x', y')}{\partial(x, y)} = \frac{y_\infty}{(y - y_\infty)^2} \begin{bmatrix} y_\infty - y & x \\ 0 & y_\infty \end{bmatrix}.$$

The last part above provides the Jacobian of the transform and models the linear distorsions. It shows that, at the origin $(0, 0)$, the Jacobian is equal to the identity matrix, which means that no linear distorsions are introduced by the transform at this point.

In the general case, it is easy to show that for an arbitrarily placed point of interest $\mathbf{x}_0 = (x_0, y_0, 1)^\top$ and vertical vanishing point $\mathbf{v}_\perp = (x_\perp, y_\perp, 1)^\top$, we can reach the above special case by applying first the translation $\mathbf{T}$ that maps the origin of the coordinate system to the selected point $\mathbf{x}_0$, and then the rotation $\mathbf{R}$ which brings the translated vertical vanishing point on the y axis. The required mapping $\mathbf{K}_\perp$ is then given by $\mathbf{K}_\perp = \mathbf{K_G}\mathbf{R}\mathbf{T}$, and can be used to warp the undistorted image and obtain the wanted image (central image in Fig. 5).