Human Detection via Classification on Riemannian Manifolds

Oncel Tuzel
Rutgers University, CS
Piscataway, NJ 08854
otuzel@caip.rutgers.edu

Fatih Porikli
Mitsubishi Electric Research Labs
Cambridge, MA 02139
fatih@merl.com

Peter Meer
Rutgers University, ECE
Piscataway, NJ 08854
meer@caip.rutgers.edu

Abstract

We present a new algorithm to detect humans in still images utilizing covariance matrices as object descriptors. Since these descriptors do not lie on a vector space, well known machine learning techniques are not adequate to learn the classifiers. The space of d-dimensional nonsingular covariance matrices can be represented as a connected Riemannian manifold. We present a novel approach for classifying points lying on a Riemannian manifold by incorporating the a priori information about the geometry of the space. The algorithm is tested on INRIA human database where superior detection rates are observed over the previous approaches.

1. Introduction

Human detection in still images is considered among the hardest examples of object detection problems. The articulated structure and variable appearance of the human body, combined with illumination and pose variations, contribute to the complexity of the problem.

The leading approaches in human detection can be separated into two groups based on the search method. The *first group of methods* is based on sequentially applying a classifier at all the possible subwindows in a given image. In [16], a polynomial support vector machine (SVM) was learned using Haar wavelets as human descriptors. Later, the work was extended to multiple classifiers trained to detect human parts, and the responses inside the detection window are combined to give the final decision [14]. Similar to still images, in [23], a real time moving human detection algorithm was described also using Haar wavelet descriptors but extracted from space-time differences in video. Using AdaBoost, the most discriminative features were selected, and multiple classifiers were combined to form a rejection cascade, such that if any classifier rejects a hypothesis then it is considered a negative example. In [4], an excellent human detector was described by training an SVM classifier using densely sampled histogram of oriented gradients (similar to SIFT descriptors) inside the detection window. The performance of the proposed descriptors was shown on INRIA human database and all the previous methods had false positive rates of at least one-two orders of magnitude higher at the same detection rates. Recently in a similar approach [24], near real time detection performances were achieved by training a cascade model using histogram of oriented gradients (HOG) features.

The *second group of methods* is based on detecting human parts [5, 9, 19] or common shapes [12] and assembling these local features according to geometric constraints to form the final human model. In [13], parts were represented by co-occurrences of local orientation features and separate detectors were trained for each part using AdaBoost. Human location was determined by maximizing the joint likelihood of part occurrences combined according to the geometric relations. A human detection system for crowded scenes was described in [11]. The approach combined local appearance features and their geometric relations with global cues by top-down segmentation based on per pixel likelihoods. Other approaches include using silhouette information either in matching [8] or in classification framework [15].

Our approach belongs to the first group, and is most similar to [23] and [24], but instead of Haar wavelets or HOG features we use covariance features as human descriptors. Covariance features were introduced in [21] for matching and texture classification problems, and later were extended to tracking [18]. A region was represented by the covariance matrix of image features, such as spatial location, intensity, higher order derivatives, etc. Similarly, we represent a human with several covariance matrices of overlapping regions. It is not adequate to use classical machine learning techniques to train the classifiers since the covariance matrices do not lie on a vector space.

Symmetric positive definite matrices (nonsingular covariance matrices) can be formulated as a connected Riemannian manifold. The main contribution of this paper is a novel approach for classifying points lying on a Riemannian manifold by incorporating the a priori information about the

1-4244-1180-7/07/$25.00 ©2007 IEEE

geometry of the space. Some of the relevant papers for clustering data points lying on differentiable manifolds can be found in [1, 20, 22].

The paper is organized as follows. In Section 2, we briefly describe the covariance descriptors. In Section 3, we present an introduction to Riemannian geometry focussing on the space of symmetric positive definite matrices. In Sections 4 and 5, we describe our algorithm for classification on Riemannian manifolds and its application to human detection. The experiments are presented in Section 6.

2. Covariance Descriptors

Here we present a brief overview of covariance descriptors [21] and its specialization for human detection. Let I be one-dimensional intensity or three-dimensional color image, and F be the $W \times H \times d$ dimensional feature image extracted from I

$$F(x, y) = \Phi(I, x, y) \tag{1}$$

where the function Φ can be any mapping such as intensity, color, gradients, filter responses, etc. For a given rectangular region $R \subset F$, let $\{\mathbf{z}_i\}_{i=1..S}$ be the d-dimensional feature points inside R. The region R is represented with the $d \times d$ covariance matrix of the feature points

$$\mathbf{C}_R = \frac{1}{S-1} \sum_{i=1}^{S} (\mathbf{z}_i - \boldsymbol{\mu})(\mathbf{z}_i - \boldsymbol{\mu})^T \tag{2}$$

where $\boldsymbol{\mu}$ is the mean of the points.

For human detection problem we define the mapping $\Phi(I, x, y)$ as

$$\left[x \;\, y \;\, |I_x| \;\, |I_y| \;\, \sqrt{I_x^2 + I_y^2} \;\, |I_{xx}| \;\, |I_{yy}| \;\, \arctan \frac{|I_x|}{|I_y|} \right]^T \tag{3}$$

where x and y are pixel location, $I_x, I_{xx}, ..$ are intensity derivatives and the last term is the edge orientation. With the defined mapping the input image is mapped to a $d = 8$ dimensional feature image. The covariance descriptor of a region is an 8×8 matrix and due to symmetry only upper triangular part is stored, which has only 36 different values. The descriptor encodes information of the variances of the defined features inside the region, their correlations with each other and spatial layout.

There is an efficient way to compute covariance descriptors using integral images [21]. After constructing $d(d+1)/2$ integral images, the covariance descriptor of any rectangular region can be computed in $O(d^2)$ time independent of the region size. We refer readers to [21] for more details of the descriptors and computational method.

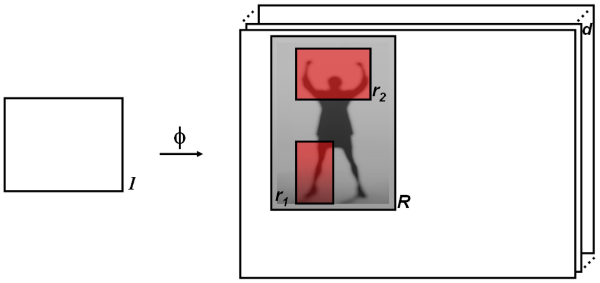

Figure 1. Covariance descriptor. The d dimensional feature image F is constructed from input image I through mapping Φ. The detection window is R and r_1, r_2 are two possible descriptor subwindows.

Given an arbitrary sized detection window R, there are a very large number of covariance descriptors that can be computed from subwindows $r_{1,2,...}$, as shown in Figure 1. We perform sampling and consider all the subwindows r starting with minimum size of $1/10$ of the width and height of the detection window R, at all possible locations. The size of r is incremented in steps of $1/10$ along the horizontal or vertical, or both, until $r = R$. Although the approach might be considered redundant due to overlaps, there is significant evidence that the overlapping regions are an important factor in detection performances [4, 24]. The boosting mechanism, that will be described later, allows us to search for the best regions.

The covariance descriptors are robust towards illumination changes. We would like to enhance this property to also include local illumination variations in an image. Let r be a possible feature subwindow inside the detection window R. We compute the covariance of the detection window $\mathbf{C}_R$ and subwindow $\mathbf{C}_r$ using integral representation. The normalized covariance matrix is computed by dividing the columns and rows of $\mathbf{C}_r$ with the respective diagonal entries of $\mathbf{C}_R$. The method described is equivalent to first normalizing the feature vectors inside the region R to have zero mean and unit standard deviation, and after that computing the covariance descriptor of subwindow r. The process only requires d^2 extra division operations.

3. Riemannian Geometry

We present a brief introduction to Riemannian geometry focussing on the space of symmetric positive definite matrices. See [2] for a more detailed description. We refer to points lying on a vector space with small bold letters $\mathbf{x} \in \mathbb{R}^m$, whereas points lying on the manifold with capital bold letters $\mathbf{X} \in \mathcal{M}$.

3.1. Riemannian Manifolds

A manifold is a topological space which is locally similar to an Euclidean space. Every point on the manifold has a neighborhood for which there exists a homeomorphism (one-to-one, onto and continuous mapping in both directions), mapping the neighborhood to $\mathbb{R}^m$. For differentiable manifolds, it is possible to define the derivatives of the curves on the manifold. The derivatives at a point $\mathbf{X}$ on the manifold lies in a vector space $T_{\mathbf{X}}$, which is the tangent space at that point. A Riemannian manifold $\mathcal{M}$ is a differentiable manifold in which each tangent space has an inner product $<,>_{\mathbf{X}}$ which varies smoothly from point to point. The inner product induces a norm for the tangent vectors on the tangent space, such that, $\|\mathbf{y}\|^2_{\mathbf{X}} =< \mathbf{y}, \mathbf{y} >_{\mathbf{X}}$.

The minimum length curve connecting two points on the manifold is called the geodesic, and the distance between the points $d(\mathbf{X}, \mathbf{Y})$ is given by the length of this curve. Let $\mathbf{y} \in T_{\mathbf{X}}$ and $\mathbf{X} \in \mathcal{M}$. From $\mathbf{X}$ there exists a unique geodesic starting with the tangent vector $\mathbf{y}$. The exponential map, $\exp_{\mathbf{X}} : T_{\mathbf{X}} \mapsto \mathcal{M}$, maps the vector $\mathbf{y}$ to the point reached by this geodesic, and the distance of the geodesic is given by $d(\mathbf{X}, \exp_{\mathbf{X}}(\mathbf{y})) = \|\mathbf{y}\|_{\mathbf{X}}$.

In general, the exponential map $\exp_{\mathbf{X}}$ is onto but only one-to-one in a neighborhood of $\mathbf{X}$. Therefore, the inverse mapping $\log_{\mathbf{X}} : \mathcal{M} \mapsto T_{\mathbf{X}}$ is uniquely defined only around the neighborhood of the point $\mathbf{X}$. If for any $\mathbf{Y} \in \mathcal{M}$, there exists several $\mathbf{y} \in T_{\mathbf{X}}$ such that $\mathbf{Y} = \exp_{\mathbf{X}}(\mathbf{y})$, then $\log_{\mathbf{X}}(\mathbf{Y})$ is given by the tangent vector with the smallest norm. Notice that both operators are point dependent where the dependence is made explicit with the subscript.

3.2. Space of Symmetric Positive Definite Matrices

The $d \times d$ dimensional symmetric positive definite matrices (nonsingular covariance matrices), Sym^+_d, can be formulated as a connected Riemannian manifold and an invariant Riemannian metric on the tangent space of Sym^+_d is given by [17]

$$< \mathbf{y}, \mathbf{z} >_{\mathbf{X}} = \operatorname{tr}\left(\mathbf{X}^{-\frac{1}{2}} \mathbf{y} \mathbf{X}^{-1} \mathbf{z} \mathbf{X}^{-\frac{1}{2}}\right). \tag{4}$$

The exponential map associated to the Riemannian metric

$$\exp_{\mathbf{X}}(\mathbf{y}) = \mathbf{X}^{\frac{1}{2}} \exp\left(\mathbf{X}^{-\frac{1}{2}} \mathbf{y} \mathbf{X}^{-\frac{1}{2}}\right) \mathbf{X}^{\frac{1}{2}} \tag{5}$$

is a global diffeomorphism (one-to-one, onto and continuously differentiable mapping in both directions). Therefore, the logarithm is uniquely defined at all the points on the manifold

$$\log_{\mathbf{X}}(\mathbf{Y}) = \mathbf{X}^{\frac{1}{2}} \log\left(\mathbf{X}^{-\frac{1}{2}} \mathbf{Y} \mathbf{X}^{-\frac{1}{2}}\right) \mathbf{X}^{\frac{1}{2}}. \tag{6}$$

The $\exp$ and $\log$ are the ordinary matrix exponential and logarithm operators. Not to be confused, $\exp_{\mathbf{X}}$ and $\log_{\mathbf{X}}$ are manifold specific operators which are also point dependent, $\mathbf{X} \in Sym^+_d$. The tangent space of Sym^+_d is the space of $d \times d$ symmetric matrices and both the manifold and the tangent spaces are $m = d(d + 1)/2$ dimensional.

For symmetric matrices, the ordinary matrix exponential and logarithm operators can be computed easily. Let $\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ be the eigenvalue decomposition of a symmetric matrix. The exponential series is

$$\exp(\mathbf{\Sigma}) = \sum_{k=0}^{\infty} \frac{\mathbf{\Sigma}^k}{k!} = \mathbf{U}\exp(\mathbf{D})\mathbf{U}^T \tag{7}$$

where $\exp(\mathbf{D})$ is the diagonal matrix of the eigenvalue exponentials. Similarly, the logarithm is given by

$$\log(\mathbf{\Sigma}) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} (\mathbf{\Sigma} - \mathbf{I})^k = \mathbf{U}\log(\mathbf{D})\mathbf{U}^T. \tag{8}$$

The exponential operator is always defined, whereas the logarithms only exist for symmetric matrices with positive eigenvalues, Sym^+_d.

From the definition of the geodesic given in the previous section, the distance between two points on Sym^+_d is measured by substituting (6) into (4)

$$\begin{aligned} d^2(\mathbf{X}, \mathbf{Y}) &= < \log_{\mathbf{X}}(\mathbf{Y}), \log_{\mathbf{X}}(\mathbf{Y}) >_{\mathbf{X}} \\ &= \operatorname{tr}\left(\log^2(\mathbf{X}^{-\frac{1}{2}} \mathbf{Y} \mathbf{X}^{-\frac{1}{2}})\right). \end{aligned} \tag{9}$$

We note that an equivalent form of the affine invariant distance metric was first given in [6], in terms of joint eigenvalues of $\mathbf{X}$ and $\mathbf{Y}$.

We define an orthogonal coordinate system on the tangent space with the vector operation. The orthogonal coordinates of a vector $\mathbf{y}$ on the tangent space at point $\mathbf{X}$ is given by

$$\operatorname{vec}_{\mathbf{X}}(\mathbf{y}) = \operatorname{upper}(\mathbf{X}^{-\frac{1}{2}} \mathbf{y} \mathbf{X}^{-\frac{1}{2}}) \tag{10}$$

where upper refers to the vector form of the upper triangular part of the matrix. The mapping $\operatorname{vec}_{\mathbf{X}}$, relates the Riemannian metric (4) on the tangent space to the canonical metric defined in $\mathbb{R}^m$.

3.3. Mean of the Points on Riemannian Manifolds

Let $\{\mathbf{X}_i\}_{i=1...N}$ be the set of points on a Riemannian manifold $\mathcal{M}$. Similar to Euclidean spaces, the Karcher mean [10] of points on Riemannian manifold, is the point on $\mathcal{M}$ which minimizes the sum of squared distances

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

where in our case d^2 is the distance metric (9). Differentiating the error function with respect to $\mathbf{Y}$ and setting it equal to zero, gives the following gradient descent procedure [17]

$$\boldsymbol{\mu}^{t+1} = \exp_{\boldsymbol{\mu}^t} \left[\frac{1}{N} \sum_{i=1}^{N} \log_{\boldsymbol{\mu}^t}(\mathbf{X}_i) \right] \tag{12}$$

which finds a local minimum of the error function. The method iterates by computing first order approximations to the mean on the tangent space. The weighted mean computation is similar to (12). We replace inside of the exponential, the mean of the tangent vectors with the weighted mean $\frac{1}{\sum_{i=1}^N w_i} \sum_{i=1}^N w_i \log_{\boldsymbol{\mu}^t}(\mathbf{X}_i)$.

4. Classification on Riemannian Manifolds

Let $\{(\mathbf{X}_i, y_i)\}_{i=1...N}$ be the training set with respect to class labels, where $\mathbf{X}_i \in \mathcal{M}$, $y_i \in \{0,1\}$ and $\mathcal{M}$ is a Riemannian manifold. We want to find a function $F(\mathbf{X}) : \mathcal{M} \mapsto \{0,1\}$, which divides the manifold into two based on the training set of labeled items.

A function which divides the manifold is rather a complicated notion compared to Euclidean space. For example, consider the simplest form a linear classifier on $\mathbb{R}^2$. A point and a direction vector on $\mathbb{R}^2$ defines a line which separates $\mathbb{R}^2$ into two. Equivalently, on a two-dimensional differentiable manifold, we can consider a point on the manifold and a tangent vector on the tangent space of the point, which together defines a curve on the manifold via exponential map. For example, if we consider the image of the lines on the 2-torus, the curves never divide the manifold into two.

A straightforward approach for classification would be to map the manifold to a higher dimensional Euclidean space, which can be considered as flattening the manifold. However in a general case, there is no such mapping that globally preserves the distances between the points on the manifold. Therefore a classifier trained on the flattened space does not reflect the global structure of the points.

4.1. Local Maps and Boosting

We propose an incremental approach by training several weak classifiers on the tangent space and combining them through boosting. We start by defining mappings from neighborhoods on the manifold to the Euclidean space, similar to coordinate charts. Our maps are the logarithm maps, $\log_{\mathbf{X}}$, that map the neighborhood of points $\mathbf{X} \in \mathcal{M}$ to the tangent spaces $T_{\mathbf{X}}$. Since this mapping is a homeomorphism around the neighborhood of the point, the structure of the manifold is preserved locally. The tangent space is a vector space and we learn the classifiers on this space. The classifiers can be trained on the tangent space at any point on the manifold. The mean of the points (11) minimizes the sum of squared distances on the manifold, therefore it is a good approximation up to a first order.

At each iteration, we compute the weighted mean of the points where the weights are adjusted through boosting. We map the points to the tangent space at the mean and learn a weak classifier on this vector space. Since the weights of the samples which are misclassified during earlier stages of boosting increase, the weighted mean moves towards these points producing more accurate classifiers for these points. The approach minimizes the approximation error through averaging over several weak classifiers.

```
Input: Training set $\{(\mathbf{X}_i, y_i)\}_{i=1...N}$, $\mathbf{X}_i \in \mathcal{M}$, $y_i \in \{0,1\}$
• Start with weights $w_i = 1/N$, $i = 1...N$,
  $F(\mathbf{X}) = 0$ and $p(\mathbf{X}_i) = \frac{1}{2}$
• Repeat for $l = 1...L$
    – Compute the response values and weights
      $z_i = \frac{y_i - p(\mathbf{X}_i)}{p(\mathbf{X}_i)(1-p(\mathbf{X}_i))}$
      $w_i = p(\mathbf{X}_i)(1 - p(\mathbf{X}_i))$.
    – Compute weighted mean of the points
      $\boldsymbol{\mu}_l = \arg\min_{\mathbf{Y}\in\mathcal{M}} \sum_{i=1}^{N} w_i d^2(\mathbf{X}_i, \mathbf{Y})$ (12). $(*)$
    – Map the data points to the tangent space at $\boldsymbol{\mu}_l$
      $\mathbf{x}_i = \mathrm{vec}_{\boldsymbol{\mu}_l}(\log_{\boldsymbol{\mu}_l}(\mathbf{X}_i))$. $(*)$
    – Fit the function $g_l(\mathbf{x})$ by weighted least-square regression of $z_i$ to $\mathbf{x}_i$ using weights $w_i$.
    – Update $F(\mathbf{X}) \leftarrow F(\mathbf{X}) + \frac{1}{2} f_l(\mathbf{X})$ where $f_l$ is defined in (15) and $p(\mathbf{X}) \leftarrow \frac{e^{F(\mathbf{X})}}{e^{F(\mathbf{X})}+e^{-F(\mathbf{X})}}$.
• Output the classifier sign
  $[F(\mathbf{X})]$ = sign $[\sum_{l=1}^{L} f_l(\mathbf{X})]$
```

Figure 2. LogitBoost on Riemannian Manifolds.

4.2. LogitBoost on Riemannian Manifolds

We start with brief description of LogitBoost algorithm [7] on vector spaces. We consider the binary classification problem, $y_i \in \{0,1\}$. The probability of $\mathbf{x}$ being in class 1 is represented by

$$p(\mathbf{x}) = \frac{e^{F(\mathbf{x})}}{e^{F(\mathbf{x})} + e^{-F(\mathbf{x})}} \qquad F(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{L} f_l(\mathbf{x}). \quad (13)$$

The LogitBoost algorithm learns the set of regression functions $\{f_l(\mathbf{x})\}_{l=1...L}$ (weak learners) by minimizing the negative binomial log-likelihood of the data $l(y, p(\mathbf{x}))$

$$-\sum_{i=1}^{N} [y_i log(p(\mathbf{x}_i)) + (1 - y_i) log(1 - p(\mathbf{x}_i))] \quad (14)$$

through Newton iterations. At the core of the algorithm, LogitBoost fits a weighted least square regression, $f_l(\mathbf{x})$ of training points $\mathbf{x}_i \in \mathbb{R}^m$ to response values $z_i \in \mathbb{R}$ with weights w_i.

The LogitBoost algorithm on Riemannian manifolds is similar to original LogitBoost, except differences at the level of weak learners. In our case, the domain of the weak learners are in $\mathcal{M}$ such that $f_l(\mathbf{X}) : \mathcal{M} \mapsto \mathbb{R}$. Following the discussion of the previous section, we learn the regression functions in the tangent space at the weighted mean of the points on the manifold. We define the weak learners as

$$f_l(\mathbf{X}) = g_l(\mathrm{vec}_{\boldsymbol{\mu}_l}(\log_{\boldsymbol{\mu}_l}(\mathbf{X}))) \quad (15)$$

and learn the functions $g_l(\mathbf{x}) : \mathbb{R}^m \mapsto \mathbb{R}$ and the weighted mean of the points $\boldsymbol{\mu}_l \in \mathcal{M}$. Notice that, the mapping vec

(10), gives the orthogonal coordinates of the tangent vectors.

The algorithm is presented in Figure 2. The steps marked with $(*)$ are the only differences from original LogitBoost algorithm. For functions $\{g_l\}_{l=1 \dots L}$, it is possible to use any form of weighted least squares regression such as linear functions, regression stumps, etc., since the domain of the functions are in $\mathbb{R}^m$.

5. Human Detection

For human detection, we combine $K = 30$ LogitBoost classifiers on Sym_8^+ with rejection cascade, as shown in Figure 3. Weak classifiers $\{g_l\}_{l=1 \dots L}$ are linear regression functions learned on the tangent space of Sym_8^+. The tangent space is $m = 36$ dimensional vector space.

Let N_{pi} and N_{ni} be the number of positive and negative images in the training set. Since any detection window sampled from a negative image is a negative sample, it is possible to generate much more negative examples than the number of negative images.

Assume that we are training the kth cascade level. We classify all the possible detection windows on the negative training images with the cascade of the previous $(k-1)$ LogitBoost classifiers. The samples which are misclassified form the possible negative set (samples classified as positive). Since the cardinality of the possible negative set is very large, we sample $N_n = 10000$ examples from this set as the negative examples at cascade level k. At every cascade level, we consider all the positive training images as the positive training set. There is a single human at each of the positive images, so $N_p = N_{pi}$.

A very large number of covariance descriptors can be computed from a single detection window and it is computationally intractable to test all of them. At each boosting iteration of kth LogitBoost level, we sample 200 subwindows among all the possible subwindows, and construct normalized covariance descriptors as described in Section 2. We learn the weak classifiers representing each subwindow, and add the best classifier which minimizes negative binomial log-likelihood (14) to the cascade level k.

Each level of cascade detector is optimized to correctly detect at least 99.8% of the positive examples, while rejecting at least 35% of the negative examples. In addition, we enforce a margin constraint between the positive samples and the decision boundary. Let $p_k(\mathbf{X})$ be the probability of a sample being positive at cascade level k, evaluated through (13). Let $\mathbf{X}_p$ be the positive example that has the $(0.998N_p)$th largest probability among all the positive examples. Let $\mathbf{X}_n$ be the negative example that has the $(0.35N_n)$th smallest probability among all the negative examples.

We continue to add weak classifiers to cascade level k until $p_k(\mathbf{X}_p) - p_k(\mathbf{X}_n) > th_b$, where we set $th_b = 0.2$. When the constraint is satisfied, a new sample is classified as positive by cascade level k if $p_k(\mathbf{X}) > p_k(\mathbf{X}_p) - th_b > p_k(\mathbf{X}_n)$ or equivalently $F_k(\mathbf{X}) > F_k(\mathbf{X}_n)$. With the proposed method, any of the positive training samples in the top 99.8 percentile have at least th_b more probability than the decision boundary. The process continues with the training of $(k+1)$th cascade level, until $k = K$.

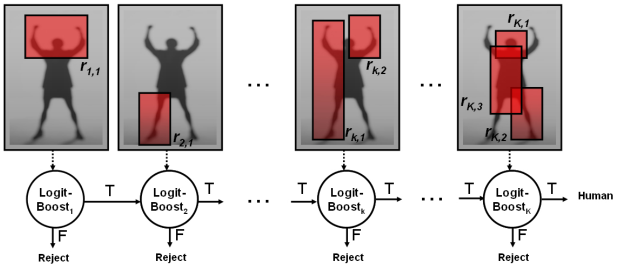

Figure 3. Cascade of LogitBoost classifiers. The kth LogitBoost classifier selects normalized covariance descriptors of subwindows $r_{k,i}$.

The method presented here is a slight modification of the LogitBoost classifier on Riemannian manifolds described in Section 4.2. We compute the weighted means of only the positive examples since negative set is not well characterized for detection tasks. Although rarely happens, if some of the features are totally correlated, there will be singularities in the covariance descriptor. We ignore those cases by adding very small identity matrix to the covariance.

6. Experiments

We perform the experiments on INRIA human database [4]. The database contains 1774 human annotations (3548 with reflections) and 1671 person free images. Detection on INRIA human database is challenging since it includes subjects with a wide range of variations in pose, clothing, illumination, background and partial occlusions. We perform the same separation of training - testing sets to directly compare the results with the methods of Dalal & Triggs [4] and Zhu et.al. [24]. To our knowledge, these two methods produce the best results published on the given database, and a detailed comparison with the other previous methods is given in [4].

In the *first experiment*, we compare our results with [4] and [24]. Although it has been noted that kernel SVM is computationally expensive, we consider both the linear and kernel SVM method of [4]. The method of [24] trains a boosted classifier using HOG features, and two different results were reported based on the normalization. Here we consider only the best performing result, the L2-norm.

In Figure 4, we plot the detection error tradeoff curves on a log-log scale. The y-axis corresponds to the miss rate, $FalseNeg/(FalseNeg + TruePos)$, and the x-axis corresponds to false positives per window (FPPW),

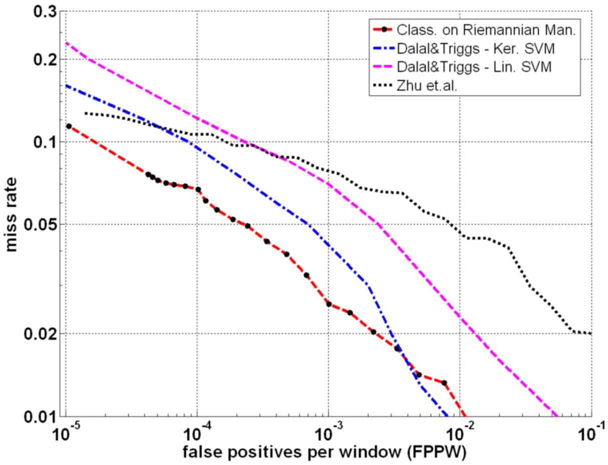

Figure 4. Comparison with methods of Dalal & Triggs [4] and Zhu et.al. [24]. The curves for other approaches are generated from the respective papers. See text for details.

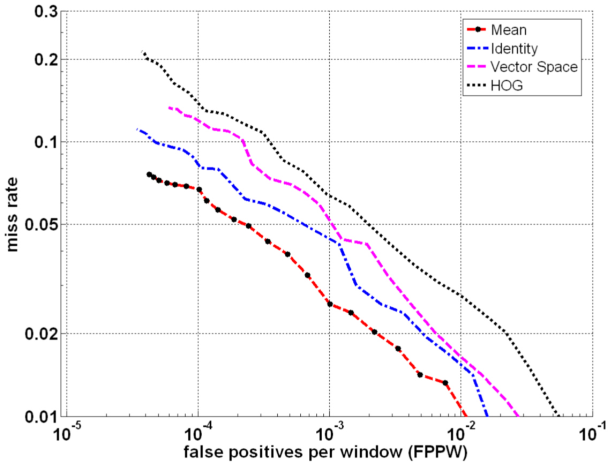

Figure 5. Detection rates of different approaches for our method. See text for details.

$FalsePos/(TrueNeg + FalsePos)$. The curve for our method is generated by adding one cascade level at a time. For example, in our case the rightmost marker at $7.5 * 10^{-3}$ FPPW corresponds to detection using only the first 11 levels of cascade, whereas the marker positioned at $4 * 10^{-5}$ FPPW corresponds to cascade of all 30 levels. The markers between the two extremes correspond to a cascade of between 11 to 30 levels.

To generate the result at 10^{-5} FPPW (leftmost marker), we shifted the decision boundaries of all the cascade levels to produce less false positives at the cost of higher miss rates. We place the decision boundary to $p_k(\mathbf{X}) > (p_k(\mathbf{X}_n) + p_k(\mathbf{X}_p))/2$, such that the margin th_b is reduced by half. See Section 5 for details. We see that at almost all the false positive rates, our miss rates are significantly lower than other approaches. The closest result to our method is the kernel SVM classifier of [4], which requires kernel evaluation at 1024 dimensional space to classify a single detection window. If we consider 10^{-4} as an acceptable false positive rate per window, our miss rate is 6.8%, where the the second best result is 9.3%.

Since the method removes samples which are rejected by the previous levels of cascade, during the training of last levels only very small amount of negative samples, order of 10^2 remained. At these levels, the training error did not generalize well, such that the same detection rates are not achieved on the test set. This can be seen by the dense markers around FPPW $< 7 * 10^{-5}$. We believe that better detection rates can be achieved at low false positive rates with introduction of more negative images. We also note that, in our method 25% of false positives come from a single textured image, where the training set does not include a similar image.

In the *second experiment*, we consider an empirical validation of the presented classification algorithm on Riemannian manifolds. In Figure 5, we present the detection error tradeoff curves for four different approaches.

- The original method, which maps the points to the tangent spaces at the weighted means.
- The mean computation step is removed from the original algorithm and points are always mapped to the tangent space at the identity matrix.
- We ignore the geometry of Sym_8^+, and stack the upper triangular part of the covariance matrix into a vector, such that learning is performed on the vector space.
- We replace the covariance descriptors with HOG descriptors, and perform original LogitBoost classification.

The original method outperforms all the other approaches significantly. The second best result is achieved by mapping points to the tangent space at the identity matrix followed by the vector space approaches.

In Figure 6, we plot the number of weak classifiers at each cascade level and the accumulated rejection rate over the cascade levels. There are very few classifiers on early levels of cascade and the first five level reject 90% of the negative examples. On average our method requires evaluation of 8.45 covariance descriptors per negative detection window, whereas on average 15.62 HOG evaluations are required in [24].

In Figure 7, several detection examples are shown for crowded scenes with humans having variable illumination, appearance, pose and partial occlusion. We search the images at five different scales and the white regions show all the detection results. We filter the detection results with adaptive bandwidth mean shift filtering [3], with bandwidth $1/10$ of the window width and height. Black dots show the modes, and ellipses are generated by averaging the detection window sizes converging to the mode.

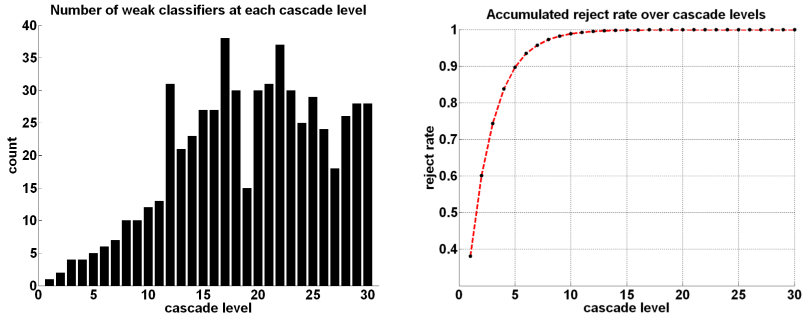

Figure 6. The number of weak classifiers at each cascade level and the accumulated rejection rate over the cascade levels. See text for details.

The training of the classifiers took two days on a current state of art PC, which is a reasonable time to train a cascade model. Given a novel image, on average the method can search around 3000 detection windows per second. The most computationally expensive operation of our method is the eigenvalue decomposition to compute the logarithm of a matrix, which requires $O(d^3)$ arithmetic operations. Compared to previous approaches, the search time is faster than [4] but slower than [24] which produces significantly lower detection rates.

7. Conclusion

We presented a new approach for human detection problem utilizing covariance matrices as object descriptors and a novel learning algorithm on the Riemannian manifolds. The proposed learning algorithm is not specific to Sym_d^+, and can be used to train classifiers for points lying on any connected Riemannian manifold. The superior performance of the proposed approach is shown on INRIA human database, where previous methods have significantly higher miss rates at almost all the false positive rates per window.

References

[1] E. Begelfor and M. Werman. Affine invariance revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume 2, pages 2087–2094, 2006.

[2] W. M. Boothby. *An Introduction to Differentiable Manifolds and Riemannian Geometry*. Academic Press, 2002.

[3] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, 2002.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, volume 1, pages 886–893, 2005.

[5] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *Intl. J. of Computer Vision*, 61(1):55–79, 2005.

[6] W. Förstner and B. Moonen. A metric for covariance matrices. Technical report, Dept. of Geodesy and Geoinformatics, Stuttgart University, 1999.

[7] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Ann. Statist.*, 28(2):337–407, 2000.

[8] D. Gavrila and V. Philomin. Real-time object detection for smart vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* Fort Collins, CO, pages 87–93, 1999.

[9] S. Ioffe and D. A. Forsyth. Probabilistic methods for finding people. *Intl. J. of Computer Vision*, 43(1):45–68, 2001.

[10] H. Karcher. Riemannian center of mass and mollifier smoothing. *Commun. Pure Appl. Math.*, 30:509–541, 1977.

[11] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* San Diego, CA, volume 1, pages 878–885, 2005.

[12] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume 1, pages 26–36, 2006.

[13] K. Mikolajczyk, C. Schmid, and A. Zisserman. Human detection based on a probabilistic assembly of robust part detectors. In *Proc. European Conf. on Computer Vision,* Prague, Czech Republic, volume 1, pages 69–81, 2004.

[14] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(4):349–360, 2001.

[15] A. Opelt, A. Pinz, and A. Zisserman. Incremental learning of object detectors using a visual shape alphabet. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume 1, pages 3–10, 2006.

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

[17] X. Pennec, P. Fillard, and N. Ayache. A Riemannian framework for tensor computing. *Intl. J. of Computer Vision*, 66(1):41–66, 2006.

[18] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on Lie algebra. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume 1, pages 728–735, 2006.

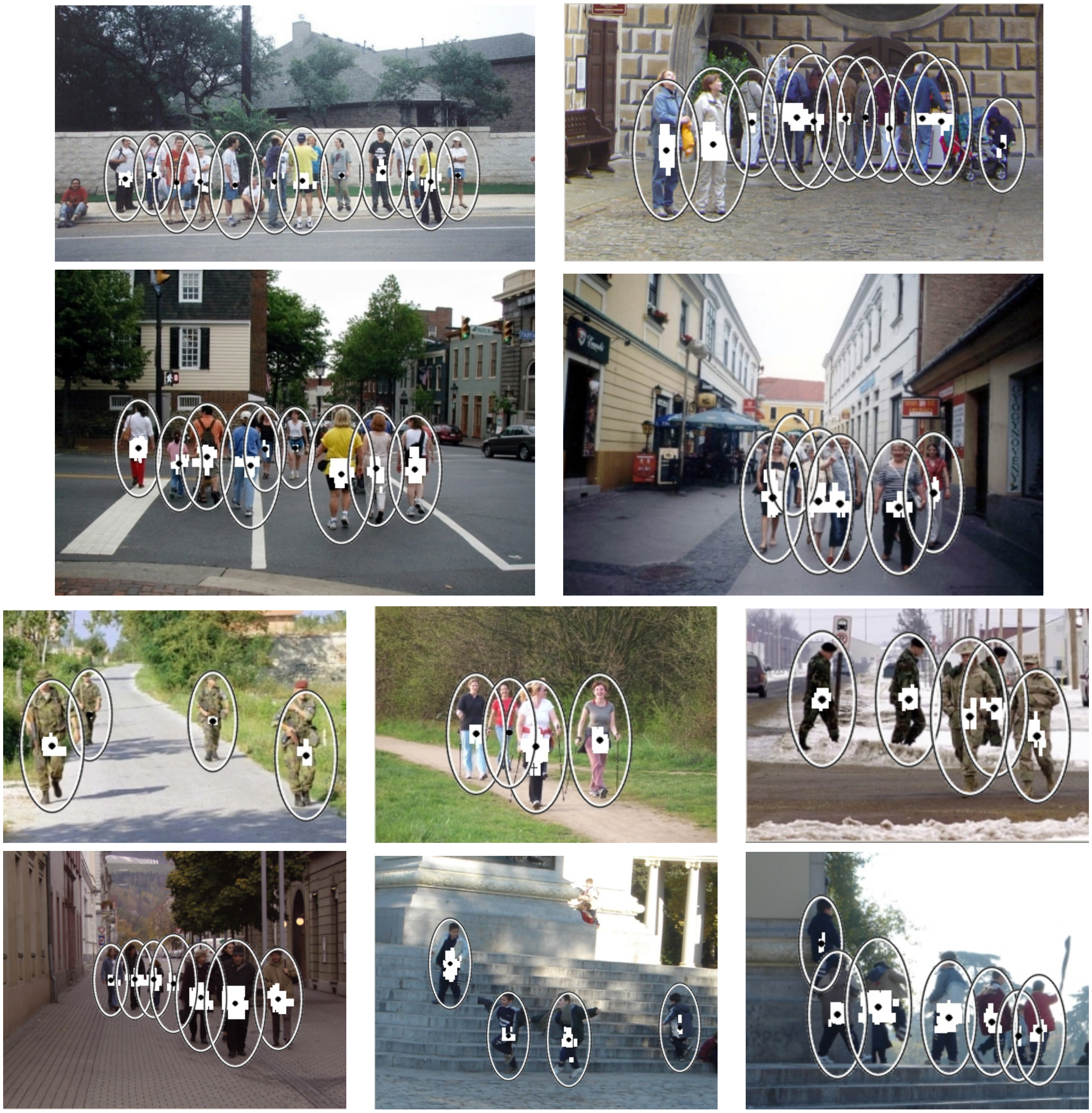

Figure 7. Detection examples. White dots show all the detection results. Black dots are the modes generated by mean shift smoothing and the ellipses are average detection window sizes. There are extremely few false positives and negatives.

[19] R. Ronfard, C. Schmid, and B. Triggs. Learning to parse pictures of people. In *Proc. European Conf. on Computer Vision,* Copenhagen, Denmark, volume 4, pages 700–714, 2002.

[20] R. Subbarao and P. Meer. Nonlinear mean shift for clustering over analytic manifolds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume 1, pages 1168–1175, 2006.

[21] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *Proc. European Conf. on Computer Vision,* Graz, Austria, volume 2, pages 589–600, 2006.

[22] O. Tuzel, R. Subbarao, and P. Meer. Simultaneous multiple 3D motion estimation via mode finding on Lie groups. In *Proc. 10th Intl. Conf. on Computer Vision,* Beijing, China, volume 1, pages 18–25, 2005.

[23] P. Viola, M. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume 1, pages 734–741, 2003.

[24] Q. Zhu, S. Avidan, M. C. Yeh, and K. T. Cheng. Fast human detection using a cascade of histograms of oriented gradients. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition,* New York, NY, volume 2, pages 1491 – 1498, 2006.