Pedestrian Detection via Classification on Riemannian Manifolds

Oncel Tuzel, *Student Member*, *IEEE*, Fatih Porikli, *Senior Member*, *IEEE*, and Peter Meer, *Senior Member*, *IEEE*

Abstract—We present a new algorithm to detect pedestrians in still images utilizing covariance matrices as object descriptors. Since the descriptors do not form a vector space, well-known machine learning techniques are not well suited to learn the classifiers. The space of d-dimensional nonsingular covariance matrices can be represented as a connected Riemannian manifold. The main contribution of the paper is a novel approach for classifying points lying on a connected Riemannian manifold using the geometry of the space. The algorithm is tested on the INRIA and DaimlerChrysler pedestrian data sets where superior detection rates are observed over the previous approaches.

Index Terms—Pedestrian detection, classification, Riemannian manifolds, symmetric positive definite matrices, boosting, object descriptors.

1 Introduction

Detecting different categories of objects in image and video content is one of the fundamental tasks in computer vision research. The success of many applications such as visual surveillance, image retrieval, robotics, autonomous vehicles, and smart cameras are conditioned on the accuracy of the detection process.

Two main processing steps can be distinguished in a typical object detection algorithm. The first task is *feature extraction*, in which the most informative object descriptors regarding the detection process are obtained from the visual content. The second task is *detection*, in which the obtained object descriptors are utilized in a classification framework to detect the objects of interest.

The feature extraction methods can be further categorized into two groups based on the representation. The first group of methods is the *sparse representations*, where a set of representative local regions is obtained as the result of an interest point detection algorithm. Reliable interest points should encapsulate valuable information about the local image content and remain stable under changes, such as in viewpoint and/or illumination. There exists an extensive literature on interest point detectors, and [14], [18], [21], [25], and [27] are only a few of the most commonly used methods that satisfy consistency over a large range of operating conditions.

Earlier approaches for part descriptors utilized intensity-based features. However, histogram-based representations of image gradients and edges in spatial context, such as scale-invariant feature transform (SIFT) descriptors [25] or shape contexts [3], were shown to yield more robust and distinctive descriptors. Several object detection algorithms were proposed by assembling the detected parts according to spatial relationships in probabilistic frameworks [12], [52], by discriminative approaches [1], [32], or via matching shapes [4], [24].

The second group of feature extraction methods is the *dense representations*, where object descriptors are obtained inside a detection window. The entire image is scanned densely (possibly each pixel), and a learned classifier of the object model is evaluated. Earlier studies utilized image intensities such as intensity templates [39], [45] or principal component analysis (PCA) coefficients [43], [47] to represent the object model. More recently, Haar-wavelet-based descriptors, which are a set of basis functions encoding intensity differences between two regions, became increasingly popular due to efficient computation and superiority to encode visual patterns. In [34], an overcomplete dictionary of basis functions were computed from overlapping regions utilizing horizontal, vertical, and diagonal features inside the detection window. Instead of sampling among an overcomplete dictionary of features, in [50], a small set of important features were selected via a greedy selection method using AdaBoost.

Most of the leading approaches in object detection are discriminative methods such as neural networks (NNs) [19], support vector machines (SVMs) [7], and boosting [41]. These methods became increasingly popular since they can cope with high-dimensional state spaces and/or are able to select relevant descriptors among a large set. In [39] and [45], NNs and, in [34], SVMs were utilized as a single strong classifier for detection of various categories of objects. The NNs and SVMs were also utilized for intermediate representations [10], [31] for final object detectors. In [50],

- *O. Tuzel is with Rutgers University, Department of Computer Science, 110 Frelinghuysen Road, Piscataway, NJ 08854. E-mail: otuzel@caip.rutgers.edu.*
- *F. Porikli is with Mitsubishi Electric Research Laboratories, 201 Broadway, Cambridge, MA 02139. E-mail: fatih@merl.com.*
- *P. Meer is with Rutgers University, Department of Electrical and Computer Engineering, 94 Brett Road, Piscataway, NJ 08854. E-mail: meer@caip.rutgers.edu.*

Manuscript received 12 Oct. 2007; revised 12 Mar. 2008; accepted 13 Mar. 2008; published online 20 Mar. 2008.
Recommended for acceptance by S. Baker, J. Matas, and R. Zabih.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMISI-2007-10-0703.
Digital Object Identifier no. 10.1109/TPAMI.2008.75.

Published by the IEEE Computer Society

multiple weak classifiers trained using AdaBoost were combined to form a rejection cascade such that if any classifier rejects a hypothesis, then it is considered a negative example. The approach provides an efficient algorithm due to sparse feature selection; besides, only a few classifiers are evaluated at most of the regions due to the cascade structure. Variants of the algorithm were also proposed to share features among different object categories, thereby providing a more efficient solution for multiple-class object detection [46]. The other approaches include the probabilistic methods [12], [52], where the conditional densities of object and nonobject classes are modeled explicitly.

Pedestrian 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.

Sparse representations for pedestrian detection include models for detecting body parts [20], [38] or common shapes [26], where these local features were assembled according to geometric constraints to form the final pedestrian model. In [28], parts were represented by co-occurrences of local orientation features, and separate detectors were trained for each part using AdaBoost. Target location was determined by maximizing the joint likelihood of part occurrences combined according to the geometric relations. A pedestrian detection system for crowded scenes was described in [23]. 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 [17] or in a classification framework [33].

Dense representations for pedestrian detection include [11], where a cost function is defined based on the part likelihoods and their joint configuration. The minimizer of the function with respect to all the possible part locations in the image plane is efficiently found using dynamic programming. In [34], a polynomial SVM was learned using Haar wavelets as pedestrian descriptors. Later, the work was extended to multiple classifiers trained to detect human parts, and the responses inside the detection window were combined to give the final decision [30]. Similar to still images, in [51], a real-time moving pedestrian 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. In [9], an excellent pedestrian detector was described by training an SVM classifier using a densely sampled histogram of oriented gradients (HOG, similar to SIFT descriptors) inside the detection window. The performance of the proposed descriptors was shown on the INRIA human data set. In a similar approach [53], near-real-time detection performances were achieved by training a cascade model using HOG features. Recently, in [40], a two-stage AdaBoost classifier was learned using shapelet features, and superior performances were reported on the INRIA human data set over all the existing methods. The initial stage learns a set of classifiers that are a combination of oriented-gradient responses, whereas the second stage combines the classifier responses to form the final detector. We refer to [8] for a detailed survey on object and pedestrian detection methods.

In this paper, we present a dense model where covariance features are utilized as pedestrian descriptors inside a detection window. Covariance features were first introduced in [48] for matching and texture classification problems and were later extended to tracking [37]. A region was represented by the covariance matrix of image features such as spatial location, intensity, higher order derivatives, etc. Similarly, we represent a pedestrian with several covariance descriptors of overlapping regions, where the best descriptors are determined with a greedy feature selection algorithm combined with boosting.

It is not trivial to build a classifier where the domain is the space of covariance matrices. Covariance matrices do not form a vector space; therefore, it is not adequate to use classical machine learning techniques to learn the classifiers. The space of nonsingular covariance matrices (symmetric positive definite 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 knowledge of the geometry of the space. Although there have been previous approaches for clustering data points lying on differentiable manifolds [2], [44], [49], to our knowledge, this paper is one of the first studies aiming at the classification problem.

The paper is organized as follows: In Section 2, we describe the covariance descriptors. In Section 3, we present an introduction to Riemannian geometry, focusing 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 pedestrian detection. The experiments are presented in Section 6.

2 Covariance Descriptors

Let I be a one-dimensional intensity or three-dimensional color image and F be the $W \times H \times d$ dimensional feature image extracted from I,

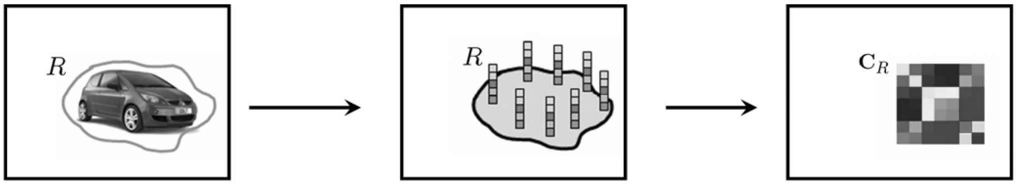
$$F(x,y) = \Phi(I,x,y), \tag{1}$$

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. In Fig. 1, we delineate the construction of covariance descriptors.

The diagonal entries of the covariance matrix represent the variance of each feature, and the nondiagonal entries are their respective correlations. There are several advantages of using covariance matrices as region descriptors.



$$\mathbf{z}_i = [x \quad y \quad |I_x| \quad |I_y| \quad |I_{xx}| \quad \dots \quad]^T \qquad \mathbf{C}_R = \frac{1}{S-1}\sum_{i=1}^{S}(\mathbf{z}_i - \mu)(\mathbf{z}_i - \mu)^T$$

Fig. 1. Covariance descriptor. The d-dimensional feature image F is constructed from input image I through mapping Φ. The region R is represented with the covariance matrix $\mathbf{C}_R$ of the features $\{\mathbf{z}_i\}_{i=1..S}$.

The representation proposes a natural way of fusing multiple features that might be correlated. A single covariance matrix extracted from a region is usually enough to match the region in different views and poses. The noise-corrupting individual samples are largely filtered out with the average filter during covariance computation. The descriptors are low-dimensional, and due to symmetry, $\mathbf{C}_R$ has only $(d^2 + d)/2$ different values. Given a region R, its covariance $\mathbf{C}_R$ does not have any information regarding the ordering and the number of points. This implies a certain scale and rotation invariance over the regions in different images. Nevertheless, if information regarding the orientation of the points is represented, such as the gradient with respect to x and y, the covariance descriptor is no longer rotationally invariant. The same argument is also correct for illumination.

2.1 Integral Images for Fast Covariance Computation

Integral images are intermediate image representations used for the fast calculation of region sums [42], [50]. Each pixel of the integral image is the sum of all the pixels inside the rectangle bounded by the upper left corner of the image and the pixel of interest. For an intensity image I, its integral image is defined as

$$\text{Integral Image }(x', y') = \sum_{x \le x', y \le y'} I(x, y). \tag{3}$$

Using this representation, any rectangular region sum can be computed in constant time. In [36], the integral images were extended to higher dimensions for the fast calculation of region histograms. Here, we follow a similar idea for the fast calculation of region covariances.

We can write the (i, j)th element of the covariance matrix defined in (2) as

$$C_R(i, j) = \frac{1}{S-1}\sum_{k=1}^{S}(z_k(i) - \mu(i))(z_k(j) - \mu(j)). \tag{4}$$

Expanding the mean and rearranging the terms, we can write

$$C_R(i, j) = \frac{1}{S-1}\left[\sum_{k=1}^{S} z_k(i)z_k(j) - \frac{1}{S}\sum_{k=1}^{S} z_k(i)\sum_{k=1}^{S} z_k(j)\right]. \tag{5}$$

To find the covariance in a given rectangular region R, we have to compute the sum of each feature dimension, $z(i)_{i=1\dots d}$, as well as the sum of the multiplication of any two feature dimensions, $z(i)z(j)_{i,j=1\dots d}$. We construct $d + d^2$ integral images for each feature dimension $z(i)$ and multiplication of any two feature dimensions $z(i)z(j)$.

Let P be the $W \times H \times d$ tensor of the integral images

$$P(x', y', i) = \sum_{x \le x', y \le y'} F(x, y, i) \qquad i = 1 \dots d \tag{6}$$

and Q be the $W \times H \times d \times d$ tensor of the second-order integral images

$$Q(x', y', i, j) = \sum_{x \le x', y \le y'} F(x, y, i)F(x, y, j) \qquad i, j = 1 \dots d. \tag{7}$$

In [50], it is shown that the integral image can be computed in one pass over the image. In our notation, $\mathbf{p}_{x,y}$ is the d-dimensional vector and $\mathbf{Q}_{x,y}$ is the $d \times d$ dimensional matrix

$$\begin{aligned} \mathbf{p}_{x,y} &= [P(x, y, 1) \dots P(x, y, d)]^T, \\ \mathbf{Q}_{x,y} &= \begin{pmatrix} Q(x, y, 1, 1) & \dots & Q(x, y, 1, d) \\ & \vdots & \\ Q(x, y, d, 1) & \dots & Q(x, y, d, d) \end{pmatrix}. \end{aligned} \tag{8}$$

Note that $\mathbf{Q}_{x,y}$ is a symmetric matrix, and $d + (d^2 + d)/2$ passes over the image are enough to compute both P and Q. The computational complexity of constructing the integral images is $O(d^2WH)$.

Let $R(x', y'; x'', y'')$ be the rectangular region, where (x', y') is the upper left coordinate and (x'', y'') is the lower right coordinate, as shown in Fig. 2. The covariance of the region bounded by (1, 1) and (x', y') is

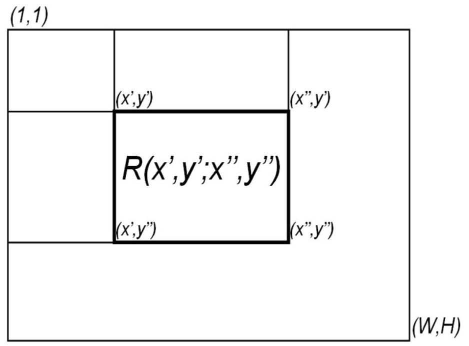


Fig. 2. Integral image. The rectangle $R(x', y'; x'', y'')$ is defined by its upper left (x', y') and lower right (x'', y'') corners in the image, and each point is a d dimensional vector.

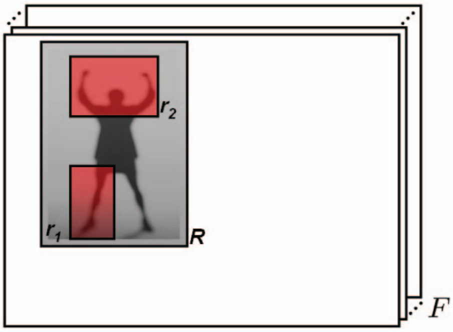


Fig. 3. The detection window is R, and r_1 and r_2 are two possible descriptor subwindows.

$$\mathbf{C}_{R(1,1;x',y')} = \frac{1}{S-1}\left[\mathbf{Q}_{x',y'} - \frac{1}{S}\mathbf{p}_{x',y'}\mathbf{p}^T_{x',y'}\right], \tag{9}$$

where $S = x' \cdot y'$. Similarly, after a few rearrangements, the covariance of the region $R(x',y';x'',y'')$ can be computed as

$$\begin{aligned}
&\mathbf{C}_{R(x',y';x'',y'')} = \\
&\quad \frac{1}{S-1}\Big[\mathbf{Q}_{x'',y''} + \mathbf{Q}_{x'-1,y'-1} - \mathbf{Q}_{x'',y'-1} - \mathbf{Q}_{x'-1,y''} \\
&\qquad - \frac{1}{S}\left(\mathbf{p}_{x'',y''} + \mathbf{p}_{x'-1,y'-1} - \mathbf{p}_{x'-1,y''} - \mathbf{p}_{x'',y'-1}\right) \\
&\qquad \left(\mathbf{p}_{x'',y''} + \mathbf{p}_{x'-1,y'-1} - \mathbf{p}_{x'-1,y''} - \mathbf{p}_{x'',y'-1}\right)^T\Big],
\end{aligned} \tag{10}$$

where $S = (x'' - x' + 1) \cdot (y'' - y' + 1)$. Therefore, after constructing integral images, the covariance of any rectangular region can be computed in $O(d^2)$ time.

2.2 Covariance Descriptors for Pedestrian Detection

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

$$\left[x \quad y \quad |I_x| \quad |I_y| \quad \sqrt{I_x^2 + I_y^2} \quad |I_{xx}| \quad |I_{yy}| \quad \arctan\frac{|I_x|}{|I_y|}\right]^T, \tag{11}$$

where x and y are the 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 the 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 the spatial layout.

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 Fig. 3. We perform sampling and consider all the subwindows r, starting with a 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, 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 [9], [53]. The greedy feature selection mechanism, which will be described later, allows us to search for the best regions during learning classifiers.

Although it has been mentioned that the covariance descriptors are robust toward illumination changes, we would like to enhance the robustness 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 an integral representation. The normalized covariance descriptor of region r, denoted as $\hat{\mathbf{C}}_r$, is computed by dividing the columns and rows of $\mathbf{C}_r$ with the square root of the respective diagonal entries of $\mathbf{C}_R$,

$$\hat{\mathbf{C}}_r = diag(\mathbf{C}_R)^{-\frac{1}{2}}\mathbf{C}_r diag(\mathbf{C}_R)^{-\frac{1}{2}}, \tag{12}$$

where $diag(\mathbf{C}_R)$ is equal to $\mathbf{C}_R$ at the diagonal entries and the rest is truncated to zero. 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. Notice that, under the transformation, $\hat{\mathbf{C}}_R$ is equal to the correlation matrix of the features inside the region R. The process only requires d^2 extra division operations.

3 Riemannian Geometry

We present a brief introduction to Riemannian geometry, focusing on the space of symmetric positive definite matrices. See [5] for a more detailed description. We refer to points in a vector space with small bold letters, $\mathbf{x} \in \mathbb{R}^m$, and to points on the manifold with capital bold letters, $\mathbf{X} \in \mathcal{M}$. Unless explicitly specified by a subscript, all the matrix norms are computed by the Frobenius norm $\|\mathbf{X}\|^2 = \mathrm{trace}(\mathbf{X}\mathbf{X}^T)$, and the vector norms are computed by the L_2 norm.

3.1 Riemannian Manifolds

A manifold is a topological space that is locally similar to a 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 lie 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 $<,>_{\mathbf{X} \in \mathcal{M}}$, which varies smoothly from point to point. The inner product induces a norm for the tangent vectors in 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 a small neighborhood of the point $\mathbf{X}$. If for any $\mathbf{Y} \in \mathcal{M}$, there exist 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 [35]

$$< \mathbf{y}, \mathbf{z} >_{\mathbf{X}} = \operatorname{trace}\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{y}\mathbf{X}^{-1}\mathbf{z}\mathbf{X}^{-\frac{1}{2}}\right). \tag{13}$$

The exponential map associated to the Riemannian metric

$$\exp_{\mathbf{X}}(\mathbf{y}) = \mathbf{X}^{\frac{1}{2}} \exp\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{y}\mathbf{X}^{-\frac{1}{2}}\right)\mathbf{X}^{\frac{1}{2}} \tag{14}$$

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

$$\log_{\mathbf{X}}(\mathbf{Y}) = \mathbf{X}^{\frac{1}{2}} \log\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{Y}\mathbf{X}^{-\frac{1}{2}}\right)\mathbf{X}^{\frac{1}{2}}. \tag{15}$$

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

$$\exp(\mathbf{\Sigma}) = \sum_{k=0}^{\infty} \frac{\mathbf{\Sigma}^k}{k!} = \mathbf{U}\exp(\mathbf{D})\mathbf{U}^T, \tag{16}$$

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

$$\log(\mathbf{\Sigma}) = \sum_{k=1}^{\infty} \frac{(-1)^{k-1}}{k} (\mathbf{\Sigma} - \mathbf{I})^k = \mathbf{U}\log(\mathbf{D})\mathbf{U}^T. \tag{17}$$

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 (15) into (13)

$$\begin{aligned} d^2(\mathbf{X}, \mathbf{Y}) &= < \log_{\mathbf{X}}(\mathbf{Y}), \log_{\mathbf{X}}(\mathbf{Y}) >_{\mathbf{X}} \\ &= \operatorname{trace}\left(\log^2\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{Y}\mathbf{X}^{-\frac{1}{2}}\right)\right). \end{aligned} \tag{18}$$

We note that an equivalent form of the affine-invariant distance metric was first given in [15] in terms of joint eigenvalues of $\mathbf{X}$ and $\mathbf{Y}$.

3.3 Minimal Representation on the Tangent Space

For classification, we need a minimal representation of the points in the tangent space. Since the tangent space is the space of symmetric matrices, there are only $d(d+1)/2$ independent coefficients, which are the upper triangular or lower triangular part of the matrix. The off-diagonal entries are counted twice during norm computation. We define an orthonormal coordinate system for the tangent space with the vector operation. The orthonormal coordinates of a tangent vector $\mathbf{y}$ in the tangent space at point $\mathbf{X}$ is given by the vector operator

$$\operatorname{vec}_{\mathbf{X}}(\mathbf{y}) = \operatorname{vec}_{\mathbf{I}}\left(\mathbf{X}^{-\frac{1}{2}}\mathbf{y}\mathbf{X}^{-\frac{1}{2}}\right), \tag{19}$$

where $\mathbf{I}$ is the identity matrix, and the vector operator at identity is defined as

$$\operatorname{vec}_{\mathbf{I}}(\mathbf{y}) = \left[y_{1,1}\ \sqrt{2}y_{1,2}\ \sqrt{2}y_{1,3}\ \ldots\ y_{2,2}\ \sqrt{2}y_{2,3}\ \ldots\ y_{d,d}\right]^T. \tag{20}$$

Notice that the tangent vector $\mathbf{y}$ is a symmetric matrix, and with the vector operator $\operatorname{vec}_{\mathbf{X}}(\mathbf{y})$, we get the orthonormal coordinates of $\mathbf{y}$, which is in $\mathbb{R}^m$.

The vector operator relates the Riemannian metric (13) on the tangent space to the canonical metric defined in $\mathbb{R}^m$

$$< \mathbf{y}, \mathbf{y} >_{\mathbf{X}} = \|\operatorname{vec}_{\mathbf{X}}(\mathbf{y})\|_2^2. \tag{21}$$

3.4 Mean of the Points on Riemannian Manifolds

Let $\{\mathbf{X}_i\}_{i=1\ldots N}$ be the set of points on a Riemannian manifold $\mathcal{M}$. Similar to euclidean spaces, the Karcher mean [22] of points on the Riemannian manifold is the point on $\mathcal{M}$ that minimizes the sum of squared Riemannian distances

$$\boldsymbol{\mu} = \arg\min_{\mathbf{X} \in \mathcal{M}} \sum_{i=1}^{N} d^2(\mathbf{X}_i, \mathbf{X}), \tag{22}$$

where, in our case, d^2 is the distance metric (18). In general, the Riemannian mean for a set of points is not necessarily unique. This can be easily verified by considering two points at antipodal positions on a sphere, where the error function is minimal for any point lying on the equator. However, it is shown in several studies that the mean is unique, and the gradient descent algorithm is convergent for Sym_d^+ [13], [29], [35].

Differentiating the error function with respect to $\mathbf{X}$, we see that the mean is the solution to the nonlinear matrix equation

$$\sum_{i=1}^{N} \log_{\mathbf{X}}(\mathbf{X}_i) = 0, \tag{23}$$

which gives the following gradient descent procedure [35]:

$$\boldsymbol{\mu}^{t+1} = \exp_{\boldsymbol{\mu}^t}\left[\frac{1}{N}\sum_{i=1}^{N} \log_{\boldsymbol{\mu}^t}(\mathbf{X}_i)\right]. \tag{24}$$

The method iterates by computing first-order approximations to the mean on the tangent space. The weighted mean computation is similar to (24). 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 $\{(\mathbf{X}_i, y_i)\}_{i=1\dots N}$ 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 that divides the manifold is a rather complicated notion compared with the euclidean space. For example, consider the simplest form of a linear classifier in $\mathbb{R}^2$. A point and a direction vector in $\mathbb{R}^2$ define a line that 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 in the tangent space of the point, which together define a curve on the manifold via an 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 spaces 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 locally preserved. The tangent space is a vector space, and we use standard machine learning techniques to learn the classifiers on this space.

For the classification task, the approximations to the Riemannian distances computed on the ambient space should be as close to the true distances as possible. Since we approximate the distances (13) on the tangent space $T_{\mathbf{X}}$,

$$d^2(\mathbf{Y},\mathbf{Z}) \approx \|\mathrm{vec}_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{Z})) - \mathrm{vec}_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{Y}))\|_2^2, \tag{25}$$

it is a first-order approximation. The approximation error can be expressed in terms of the pairwise distances computed on the manifold and the tangent space

$$\begin{aligned}\epsilon = \sum_{i=1}^{N}\sum_{j=1}^{N} \Big(& d(\mathbf{X}_i,\mathbf{X}_j) \\ & - \left\|\mathrm{vec}_{\mathbf{X}}(\log_{\mathbf{X}}(\mathbf{X}_i)) - \mathrm{vec}_{\mathbf{X}}\big(\log_{\mathbf{X}}(\mathbf{X}_j)\big)\right\|_2 \Big)^2,\end{aligned} \tag{26}$$

which is equal to

$$\begin{aligned}\epsilon_{Sym_d^+} = \sum_{i=1}^{N}\sum_{j=1}^{N} \Big(& \left\|\log\Big(\mathbf{X}_i^{-\frac{1}{2}}\mathbf{X}_j\mathbf{X}_i^{-\frac{1}{2}}\Big)\right\|_F \\ & - \left\|\log\Big(\mathbf{X}^{-\frac{1}{2}}\mathbf{X}_i\mathbf{X}^{-\frac{1}{2}}\Big) - \log\Big(\mathbf{X}^{-\frac{1}{2}}\mathbf{X}_j\mathbf{X}^{-\frac{1}{2}}\Big)\right\|_F \Big)^2\end{aligned} \tag{27}$$

for the space of symmetric positive definite matrices, using (15) and (21).

The classifiers can be learned on the tangent space at any point $\mathbf{X}$ on the manifold. The best approximation, which preserves the pairwise distances, is achieved at the minimum of $\epsilon_{Sym_d^+}$. The error can be minimized with respect to $\mathbf{X}$, which gives the best tangent space to learn the classifier.

Since the mean of the points (22) is the minimizer of the sum of squared distances from the points in the set and the mapping preserves the structure of the manifold locally, it is also a good candidate for the minimizer of the error function (27). However, to our knowledge, a theoretical proof does not exist. For some special cases, it can be easily verified that the mean is the minimizer. Such a case arises when all the points lie on a geodesic curve, where the approximation error is zero for any point lying on the curve. Since the mean also lies on the geodesic curve, the approximation is perfect. Nevertheless, for a general set of points, we only have empirical validation based on simulations. We generated random points on Sym_d^+ many times with varying d. The approximation errors were measured on the tangent spaces at any of the points $T_{\mathbf{X}_{\mathbf{i=1\dots N}}}$ and at the mean $T_{\mathbf{X}_{\boldsymbol{\mu}}}$. In our simulations, the errors computed on the tangent spaces at the means were significantly lower than any other choice, and counterexamples were not observed. The simulations were also repeated for weighted sets of points, where the minimizers of the weighted approximation errors were achieved at the weighted means of the points.

Therefore, the weak learners are learned on the tangent space at the mean of the points. At each iteration, we compute the weighted mean of the points through (24), where the weights are adjusted through boosting. Then, we map the points to the tangent space at the weighted mean and learn a weak classifier on this vector space. Since the weights of the samples that are misclassified during the earlier stages of boosting increase, the weighted mean moves toward these points, and more accurate classifiers are learned for these points. The process is illustrated in Fig. 4. To evaluate a test example, the sample is projected to the tangent spaces at the computed weighted means and the weak learners are evaluated (Fig. 5). The approximation error is minimized by averaging over several weak learners.

4.2 LogitBoost on Riemannian Manifolds

We start with a brief description of the LogitBoost algorithm [16] 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

$$p(\mathbf{x}) = \frac{e^{F(\mathbf{x})}}{e^{F(\mathbf{x})} + e^{-F(\mathbf{x})}}, \qquad F(\mathbf{x}) = \frac{1}{2}\sum_{l=1}^{L} f_l(\mathbf{x}). \tag{28}$$

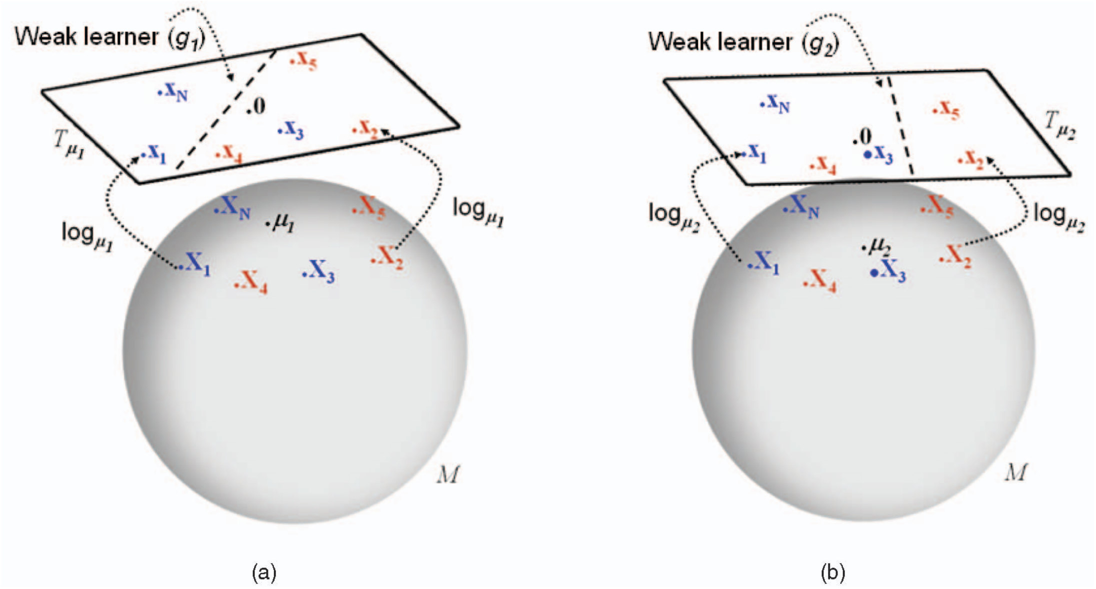


Fig. 4. Two iterations of boosting on a Riemannian manifold. The manifold is depicted with the surface of the sphere, and the plane is the tangent space at the mean. The samples are projected to tangent spaces at means via $\log_{\mu_l}$. The weak learners g_l are learned on the tangent spaces T_{μ_l}. (a) Sample $\mathbf{X}_3$ is misclassified; therefore its weight increases. (b) In the second iteration of boosting, the weighted mean moves toward $\mathbf{X}_3$.

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

$$-\sum_{i=1}^{N}[y_i log(p(\mathbf{x}_i)) + (1-y_i) log(1-p(\mathbf{x}_i))], \qquad (29)$$

through Newton iterations. At the core of the algorithm, LogitBoost fits a weighted least squares 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, where

$$z_i = \frac{y_i - p(\mathbf{x}_i)}{p(\mathbf{x}_i)(1-p(\mathbf{x}_i))}, \qquad w_i = p(\mathbf{x}_i)(1-p(\mathbf{x}_i)). \qquad (30)$$

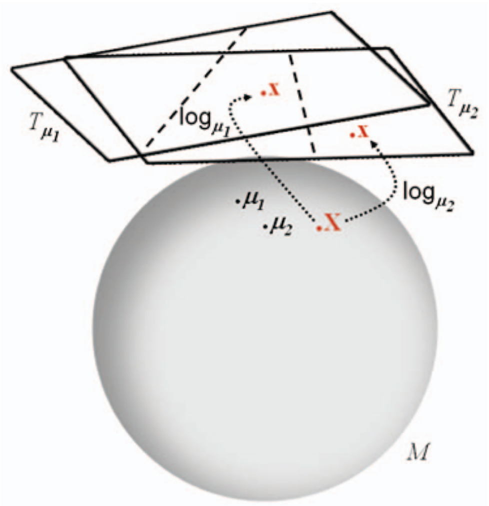


Fig. 5. Classification on a Riemannian manifold. The sample $\mathbf{X}$ is projected to the tangent spaces T_{μ_l} and the weak learners are evaluated.

The LogitBoost algorithm on Riemannian manifolds is similar to the original LogitBoost, except for a few 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 on the tangent space at the weighted mean of the points. We define the weak learners as

$$f_l(\mathbf{X}) = g_l(\mathrm{vec}_{\mu_l}(\log_{\mu_l}(\mathbf{X}))) \qquad (31)$$

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_{μ_l} (19) gives the orthonormal coordinates of the tangent vectors in T_{μ_l}.

The algorithm is presented in Fig. 6. The steps marked with $(^*)$ are the differences from the 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 domains of the functions are in $\mathbb{R}^m$.

5 Pedestrian Detection

In this section, we describe the utilization of the LogitBoost algorithm on Riemannian manifolds for the pedestrian detection problem. The domain of the classifier is the space of eight-dimensional symmetric positive definite matrices Sym_8^+. We combine $K = 30$ LogitBoost classifiers on Sym_8^+ with rejection cascade, as shown in Fig. 7. Weak learners $g_{k,l}$ are linear regression functions learned on the tangent space of Sym_8^+. The tangent space is a $m = 36$-dimensional vector space.

To avoid confusion with the previous section, we refer to the training set by $\{(R_i, y_i)\}_{i=1\dots N}$, where R_i are the image windows containing the background and pedestrians, and $y_i \in \{0, 1\}$ are the labels. A very large number of covariance descriptors can be computed from a single

```
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 through (24)
        $\boldsymbol{\mu}_l = \arg\min_{\mathbf{X}\in\mathcal{M}} \sum_{i=1}^{N} w_i d^2(\mathbf{X}_i, \mathbf{X})$ (*)
      – 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 (31) and $p(\mathbf{X}) \leftarrow \frac{e^{F(\mathbf{X})}}{e^{F(\mathbf{X})} + e^{-F(\mathbf{X})}}$
  • Store $F = \{\boldsymbol{\mu}_l, g_l\}_{l=1...L}$
  • Output the classifier
    sign$[F(\mathbf{X})]$ = sign$[\sum_{l=1}^{L} f_l(\mathbf{X})]$
```

Fig. 6. LogitBoost on Riemannian manifolds.

detection window R. Therefore, we do not have a single set of positive and negative features but several sets corresponding to each of the possible subwindows. Each weak learner is associated with a single subwindow of the detection window. Let $r_{k,l}$ be the subwindow associated with lth weak learner of cascade level k. The normalized covariance descriptor of the ith training sample associated with region $r_{k,l}$ is referred by $\mathbf{X}_{i,k,l} = \hat{\mathbf{C}}_{i,r_{k,l}}$. For simplicity, we use the shorthand notation

$$f_{k,l}(R_i) = f_{k,l}(\mathbf{X}_{i,k,l}) \tag{32}$$

for weak learners.

Let R_i^+ and R_i^- refer to the N_p positive and N_n negative samples in the training set, where $N = N_p + N_n$. While training the kth cascade level, we classify all the negative examples $\{R_i^-\}_{i=1...N_n}$ with the cascade of the previous $(k-1)$ LogitBoost classifiers. The samples that are correctly classified (samples classified as negative) are removed from the training set. Any window sampled from a negative image is a negative example; therefore, the cardinality of the negative set N_n is very large. During the training of each cascade level, we sample 10,000 negative examples.

The learning algorithm is slightly customized for the pedestrian detection task. We do not have a fixed number of weak learners L for each LogitBoost classifier k but a variable number L_k. Each cascade level is optimized to correctly detect at least 99.8 percent of the positive examples, while rejecting at least 35 percent of the negative examples. In addition, we enforce a margin constraint between the positive samples and the decision boundary. Let $p_k(R)$ be the learned probability function of a sample being positive at cascade level k, evaluated through (28). Let R_p be the positive example that has the $(0.998N_p)$th largest probability among all the positive examples. Let R_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 $p_k(R_p) - p_k(R_n) > margin$, where we set $margin = 0.2$. When the constraint is satisfied, the threshold (decision boundary) for cascade level k is stored as $thrd_k = F_k(R_n)$.

A test sample is classified as positive by cascade level k if $F_k(R) > thrd_k$ or, equivalently, $p_k(R) > p_k(R_n)$. With the proposed method, any of the positive training samples in the top 99.8 percentile have at least $margin$ more probability than the points on the decision boundary. The process continues with the training of the $(k+1)$th cascade level until $k = K$.

We incorporate a greedy feature selection method to produce a sparse set of classifiers focusing on important subwindows. At each boosting iteration l of the kth LogitBoost level, we sample 200 subwindows among all the possible subwindows and construct normalized covariance descriptors. We learn the weak classifiers representing each subwindow and add the best classifier that minimizes the negative binomial log likelihood (29) to the cascade level k. The procedure iterates with the training of the $(l+1)$th weak learner until the specified detection rates are satisfied.

The negative sample set is not well characterized for detection tasks. Therefore, while projecting the points to the tangent space, we compute the weighted mean of only the positive samples. Although it rarely happens, if some of the features are fully correlated, there will be singularities in the covariance descriptor. We ignore those cases by adding

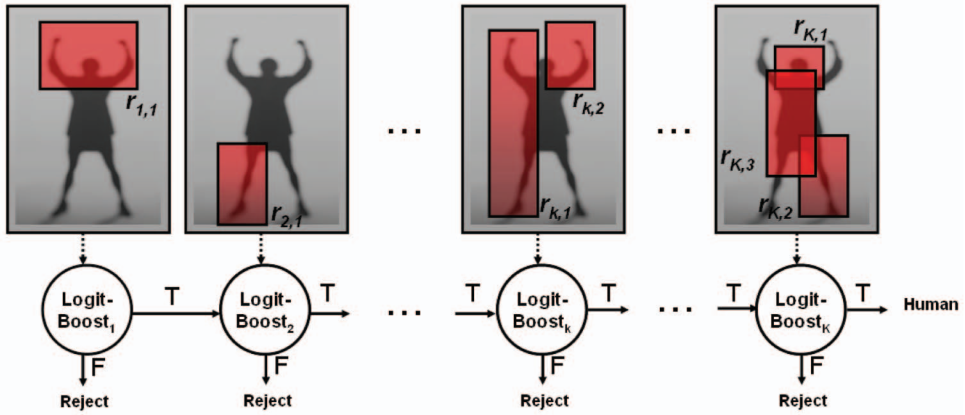


Fig. 7. Cascade of LogitBoost classifiers. The kth LogitBoost classifier selects normalized covariance descriptors of subwindows $r_{k,i}$.

```
Input: Training set $\{(R_i, y_i)\}_{i=1...N}$, $y_i \in \{0,1\}$
• Repeat for $k = 1...K$
  – Classify negative examples $\{R_i^-\}_{i=1...N_n}$ with the cascade of $(k-1)$ classifiers and remove samples which are correctly (negative) classified
  – Start with weights $w_i = 1/N$, $i = 1...N$, $F_k(R) = 0$ and $p_k(R_i) = \frac{1}{2}$, let $l = 1$
  – Repeat while $p_k(R_p) - p_k(R_n) < margin$
    ∗ Compute the response values and weights
      $z_i = \frac{y_i - p_k(R_i)}{p_k(R_i)(1-p_k(R_i))}$, $w_i = p_k(R_i)(1-p_k(R_i))$
    ∗ Sample $\{r_{k,t}\}_{t=1...200}$ subwindows and construct normalized covariance descriptors
      $\mathbf{X}_{i,k,t} = \hat{\mathbf{C}}_{i,r_{k,t}}$
    ∗ Repeat for $t = 1...200$
      · Compute the weighted mean of the positive samples $\{X^+_{i,k,t}\}_{i=1...N_p}$ through (24)
        $\boldsymbol{\mu}_{k,t} = \arg\min_{\mathbf{X}\in Sym_8^+} \sum_{i=1}^{N_p} w_i d^2(\mathbf{X}^+_{i,k,t}, \mathbf{X})$
      · Map the data points to the tangent space at $\boldsymbol{\mu}_{k,t}$
        $\mathbf{x}_{i,k,t} = \mathrm{vec}_{\boldsymbol{\mu}_t}(\log_{\boldsymbol{\mu}_t}(\mathbf{X}_{i,k,t}))$
      · Fit function $g_{k,t}(\mathbf{x})$ by weighted least-squares regression of $z_i$ to $\mathbf{x}_{i,k,t}$ using weights $w_i$
    ∗ Update $F_k(R) \leftarrow F_k(R) + \frac{1}{2} f_{k,l}(R)$, where $f_{k,l}$ is the best classifier among $\{f_{k,t}\}_{t=1...200}$ which minimizes the negative binomial log-likelihood (29) and $p_k(R) \leftarrow \frac{e^{F_k(R)}}{e^{F_k(R)}+e^{-F_k(R)}}$
    ∗ Sort positive and negative samples according to descending probabilities and find samples at the decision boundaries
      $R_p = (0.998N_p)$-th $R^+$, $R_n = (0.35N_n)$-th $R^-$
    ∗ $l = l + 1$
  – Store $F_k = \{(r_{k,l}, \boldsymbol{\mu}_{k,l}, g_{k,l})\}_{l=1...L_k}$ and $thrd_k = F_k(R_n)$
```

Fig. 8. Pedestrian detection with cascade of LogitBoost classifiers on Sym_8^+.

a very small identity matrix to the covariance. The pedestrian detection with cascade of LogitBoost classifiers on Sym_8^+ is given in Fig. 8.

The learning algorithm produces a set of K LogitBoost classifiers, which are composed of L_k triplets,

$$F_k = \left\{(r_{k,l}, \boldsymbol{\mu}_{k,l}, g_{k,l})\right\}_{l=1...L_k} \text{ and } thrd_k, \tag{33}$$

where $r_{k,l}$ is the selected subwindow, $\boldsymbol{\mu}_{k,l}$ is the mean, and $g_{k,l}$ is the learned regression function of the lth weak learner of the kth cascade. To evaluate a *test region* R with kth classifier, the normalized covariance descriptors constructed from regions $r_{k,l}$ are projected to tangent spaces $T_{\boldsymbol{\mu}_{k,l}}$, and the features are evaluated with $g_{k,l}$,

$$\begin{aligned} &\mathrm{sign}[F_k(R) - thrd_k] \\ &= \mathrm{sign}\left[\sum_{l=1}^{L_k} g_{k,l}\Big(\mathrm{vec}_{\boldsymbol{\mu}_{k,l}}\Big(\log_{\boldsymbol{\mu}_{k,l}}\Big(\hat{\mathbf{C}}_{r_{k,l}}\Big)\Big)\Big) - thrd_k\right]. \end{aligned} \tag{34}$$

The initial levels of the cascade are learned on relatively easy examples; thus, there are very few weak classifiers in these levels. Due to the cascade structure, only a few are evaluated for most of the test samples, which produce a very efficient solution.

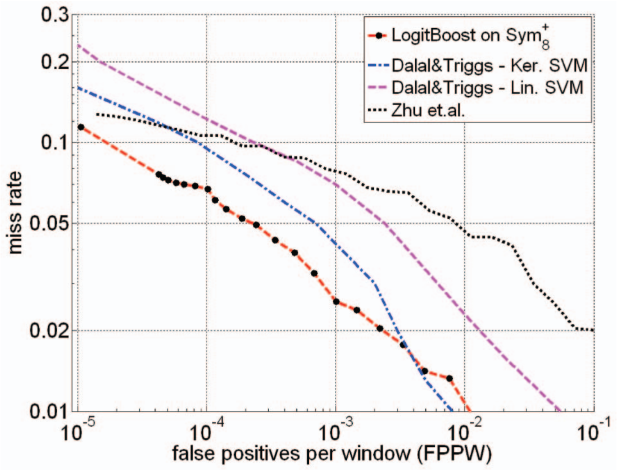


Fig. 9. Comparison with the methods of Dalal and Triggs [9] and Zhu et al. [53] on the INRIA data set. The curves for other approaches are generated from the respective papers. See text for details.

6 Experiments

We conduct experiments on two challenging data sets, INRIA and DaimlerChrysler. We compare the performance of our method with the best results published on the given data sets. In addition, we present several detection examples for crowded scenes.

6.1 Experiments on the INRIA Data Set

The INRIA pedestrian data set [9] contains 1,774 pedestrian annotations (3,548 with reflections) and 1,671 person-free images. The pedestrian annotations were scaled into a fixed size of 64 × 128 windows, which include a margin of 16 pixels around the pedestrians. The data set was divided into two, where 2,416 pedestrian annotations and 1,218 person-free images were selected as the training set, and 1,132 pedestrian annotations and 453 person-free images were selected as the test set. Detection on the INRIA pedestrian data set is challenging since it includes subjects with a wide range of variations in pose, clothing, illumination, background, and partial occlusions.

In the *first experiment*, we compare our results with [9] and [53]. Although it has been noted that kernel SVM is computationally expensive, we consider both the linear and kernel SVM method in [9]. In [53], a cascade of AdaBoost classifiers was trained using HOG features, and two different results were reported based on the normalization of the descriptors. Here, we consider only the best performing result, the L_2 norm.

In Fig. 9, we plot the detection error trade-off curves on a log-log scale. The y-axis corresponds to the miss rate $\frac{\text{FalseNeg}}{\text{FalseNeg+TruePos}}$, and the x-axis corresponds to false positives per window (FPPW) $\frac{\text{FalsePos}}{\text{TrueNeg+FalsePos}}$. 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 the cascade of all 30 levels. The markers between the two extremes correspond to a cascade of between 11 and 30 levels.

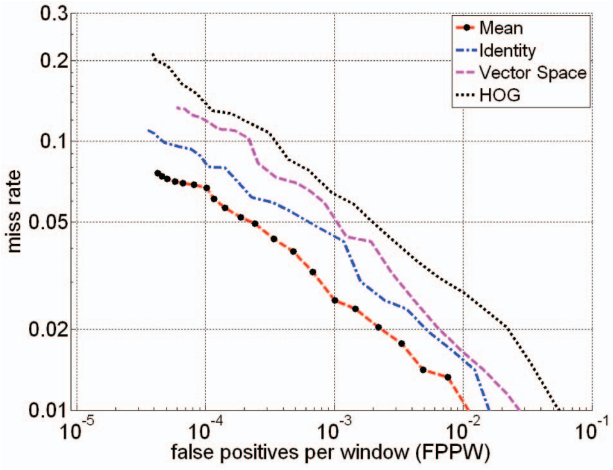


Fig. 10. Detection rates of different approaches for our method on the INRIA data set. See text for details.

To generate the result at 10^{-5} FPPW (leftmost marker), we shifted the decision boundaries of all the cascade levels $thrd_k$ to produce fewer false positives at the cost of higher miss rates. We see that at almost all the false-positive rates, our miss rates are significantly lower than the other approaches. The closest result to our method is the kernel SVM classifier in [9], which requires kernel evaluation at 1,024-dimensional space to classify a single detection window. If we consider 10^{-4} as an acceptable FPPW, our miss rate is 6.8 percent, where the second best result is 9.3 percent.

Since the method removes samples that were rejected by the previous levels of cascade, during the training of the last levels, only a very small number of negative samples, on the 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 the introduction of more negative images. In our method, 25 percent of false positives originated from a single image, which contained a flower texture, where the training set did not include a similar example.

In the *second experiment*, we consider an empirical validation of the presented classification algorithm on Riemannian manifolds. In Fig. 10, we present the detection error trade-off curves for the following four different approaches:

- We apply 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.
- 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 (vector space) 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. Notice that our LogitBoost implementation utilizing HOG descriptors has 3 percent more miss rate at 10^{-4} FPPW than [53], which trains an AdaBoost classifier. The performance is significantly degraded beyond this point.

In the *third experiment*, we examine the sensitivity of the covariance and HOG descriptors to translation and scaling of the target windows relative to the original position. The performance of the HOG descriptors is tested with our implementation. The false-positive rates of both classifiers are fixed at 10^{-4} FPPW. The translation sensitivity is presented in Fig. 11, where we observe that the covariance descriptors are less sensitive to small translations of the target windows. The detection rate is almost constant for ± 6 pixel translation, which approximately corresponds to 10 percent translation in the x-axis and 5 percent translation in the y-axis of the target window (64×128).

The scale sensitivity of both methods are presented in Fig. 12, where we again observe that the covariance

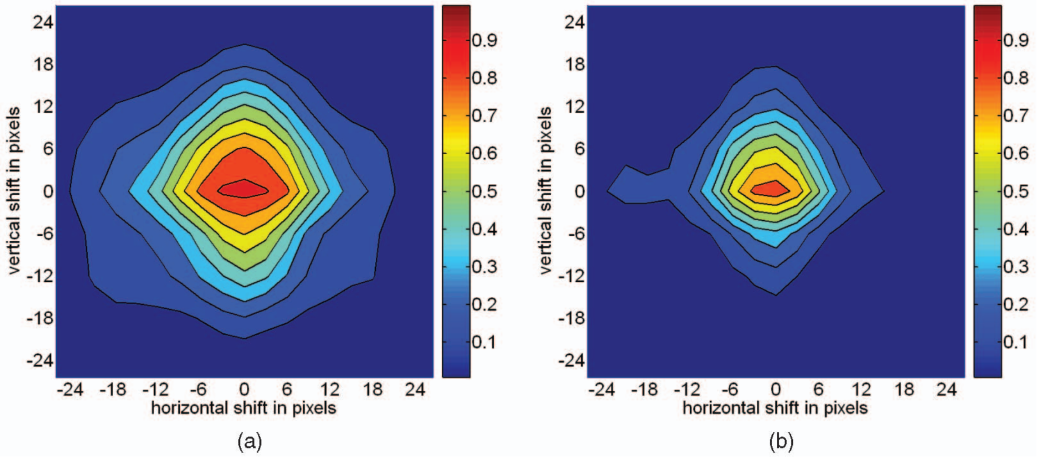


Fig. 11. Translation sensitivity. (a) Covariance descriptors. (b) HOG descriptors. Covariance descriptors have a larger region of support.

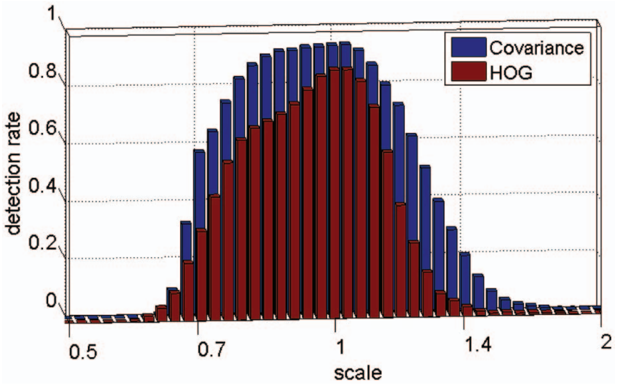


Fig. 12. Scale sensitivity. The x-axis is plotted at log scale. Covariance descriptors are less sensitive to small scale changes.

descriptors are less sensitive to small scalings of the target windows. For covariance descriptors, the detection rates are almost constant for ±20 percent scalings and gradually decrease beyond.

For detection applications, there is a trade-off between the low and high sensitivity of the detector with respect to small transformations. While detecting objects in novel scenes, objects are searched through the transformation space. A detector with invariance to small transformations has the advantage of reducing the size of the search space, whereas the space should be searched more densely with a high-sensitivity detector. Besides, a less sensitive detector is more desirable when the target objects have high variability and training data is not perfectly aligned. On the other hand, a highly sensitive detector can better localize the targets. In Figs. 11 and 12, we see that the detection rates for our approach are smooth and symmetric functions with respect to both translation and scale, having a peak at the original location of the target. Therefore, with a simple maxima search, we can accurately localize the targets. Please see Section 6.3 for more details.

In Fig. 13, 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 levels reject 90 percent of the negative examples. On the average, our method requires the evaluation of 8.45 covariance descriptors per negative detection window, whereas on the average, 15.62 HOG evaluations were required in [53].

6.2 Experiments on the DaimlerChrysler Data Set

The DaimlerChrysler data set [31] contains 4,000 pedestrian (24,000 with reflections and small shifts) and 25,000 nonpedestrian annotations. As opposed to the INRIA data set, nonpedestrian annotations were selected by a preprocessing step from the negative samples, which match a pedestrian shape template based on the average Chamfer distance score. Both annotations were scaled into a fixed size of 18 × 36 windows, and pedestrian annotations include a margin of 2 pixels around. The data set was organized into three training and two test sets, each of them having 4,800 positive and 5,000 negative examples. The small size of the windows, combined with a carefully arranged negative set, makes detection on the DaimlerChrysler data set extremely challenging. In addition, 3,600 person-free images with varying sizes between 360 × 288 and 640 × 480 were also supplied.

In [31], an experimental study was described, comparing three different feature descriptors and various classification techniques. The compared feature descriptors were the PCA coefficients, Haar wavelets, and local receptive fields (LRFs), which are the output of the hidden layer of a specially designed feed-forward NN. The connections of the neurons in the hidden layer of the NN were restricted to local regions of the image, and the hidden layers were divided into branches, with all the neurons sharing the same set of weights. Although several other classification methods were also considered, the best detection performances among all the different features were achieved utilizing SVMs.

In the *first experiment*, we compare our method with the best results for each descriptor in [31]. The same training configuration is prepared by selecting two out of three training sets. Since the number of nonpedestrian annotations was very limited for the training of our method, we adapted the training parameters. A cascade of $K = 15$ LogitBoost classifiers on Sym_8^+ is learned, where each level is optimized to detect at least 99.75 percent of the positive examples, while rejecting at least 25 percent negative samples.

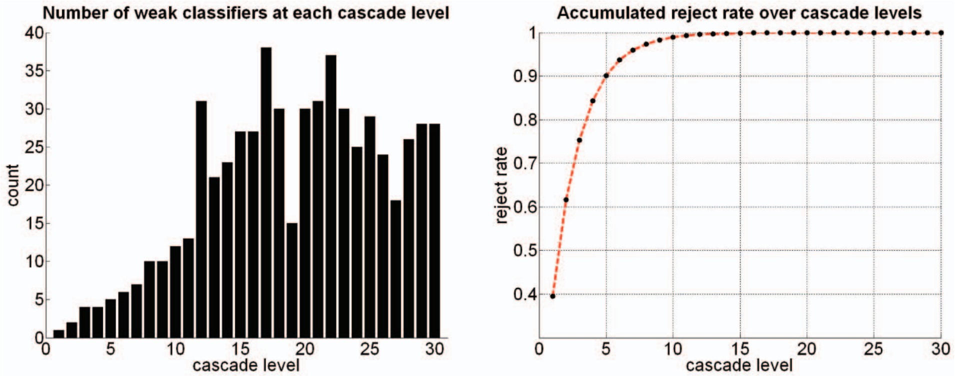


Fig. 13. The number of weak classifiers at each cascade level and the accumulated rejection rate over the cascade levels. See text for details.

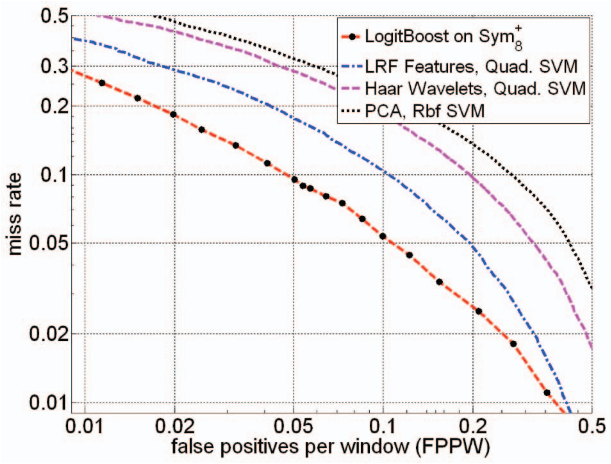


Fig. 14. Comparison with [31] on the DaimlerChrysler data set. The curves for other approaches are generated from the original paper. See text for details.

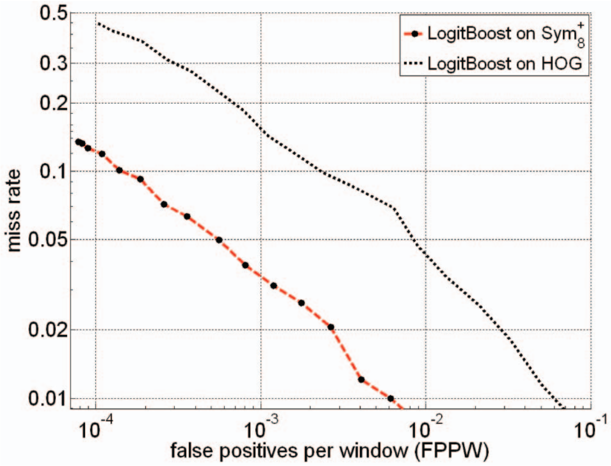


Fig. 15. Comparison of covariance and HOG descriptors on the DaimlerChrysler data set. See text for details.

In Fig. 14, we plot the detection error trade-off curves. The cascade of 15 LogitBoost classifiers produced an FPPW rate of 0.05. The detection rates with lower FPPW are generated by shifting the decision boundaries of all the cascade levels gradually, until $\mathrm{FPPW} = 0.01$. We see that our approach has significantly lower miss rates at all the false-positive rates. This experiment should not be confused with the experiments on the INRIA data set, where much lower FPPW rates were observed. Here, the negative set consists of hard examples selected by a preprocessing step.

In the *second experiment*, we set up a different test configuration on the DaimlerChrysler data set. The 3,600 person-free images are divided into two, where 2,400 images are selected as the negative training set, and 1,200 images are selected for the negative test set. For both the covariance descriptors and the HOG descriptors, we trained a cascade of $K = 25$ classifiers. We observed that the object sizes were too small for HOG descriptors to separate among positive and negative examples at the later levels of cascade. The classifiers trained utilizing HOG descriptors failed to achieve the specified detection (99.8 percent) and rejection rates (35.0 percent). We stopped adding weak learners to a cascade level after reaching $L_k = 100$. The detection error trade-off curves are given in Fig. 15, where we see that the covariance descriptors significantly outperform HOG descriptors.

6.3 Detection Examples

Since the sizes of the pedestrians in novel scenes are not known a priori, the images are searched at multiple scales. There are two searching strategies. The first strategy is to scale the detection window and apply the classifier at multiple scales. The second strategy is to scale the image and apply the classifier at the original scale. In covariance representation, we utilized gradient-based features, which are scale dependent. Therefore, evaluating the classifier at the original scale (second strategy) produces the optimal result. However, in practice, up to scales of 2x, we observed that the detection rates were almost the same, whereas in more extreme scale changes, the performance of the first strategy degraded. The drawback of the second strategy is a slightly increased search time, since the method requires computation of the filters and the integral representation at multiple scales.

Utilizing the classifier trained on the INRIA data set, we generated several detection examples for crowded scenes with pedestrians having variable illumination, appearance, pose, and partial occlusion. The results are shown in Fig. 16. The images are searched at five scales using the first strategy, starting with the original window size of 64 × 128 and two smaller and two larger scales of ratio 1.2. The white dots are all the detection results, and we filtered them with adaptive bandwidth mean shift filtering [6] with bandwidth $1/10$th of the window width and height. Black dots show the modes, and ellipses are generated by averaging the detection window sizes converging to the modes.

6.4 Computational Complexity

The training of the classifiers took two days on a Pentium D 2.80-Ghz processor with 2.00 Gbytes of RAM with a C++ implementation, which is a reasonable time to train a cascade model. The computation of the tensor of integral images requires $O(d^2WH)$ arithmetic operations, which approximately takes 0.1 second for a 320 × 240 image. The computation of the normalized covariance descriptor of an arbitrary region requires $O(d^2)$ operations using the integral structures and is invariant of the region size.

The most computationally expensive operation during the classification is the eigenvalue decomposition to compute the logarithm of a symmetric matrix, which requires $O(d^3)$ arithmetic operations. Given a test image, on the average, the method can search around 3,000 detection windows per second, which approximately corresponds to 3 seconds for a dense scan of a 320 × 240 image, 3 pixel jumps vertically and horizontally.

7 Conclusion

We presented a new approach for the pedestrian detection problem utilizing covariance matrices as object descriptors and a novel learning algorithm on the Riemannian

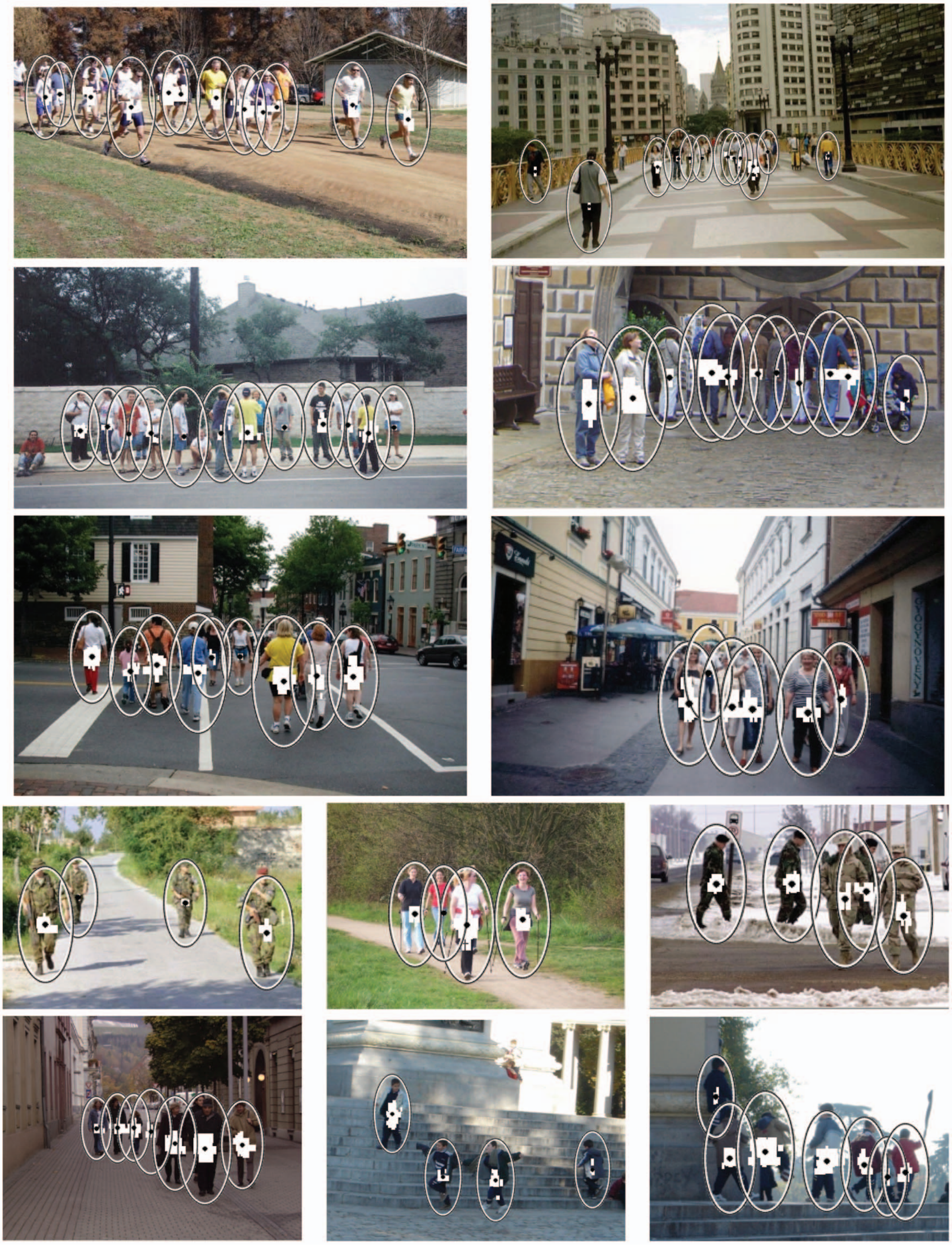


Fig. 16. Detection examples. The classifier is trained on the INRIA data set. White dots show all the detection results. Black dots are the modes generated by mean shift smoothing, and the ellipses are average detection window sizes. There are extremely few false positives and negatives.

manifolds. The superior performance of the proposed approach is shown on the INRIA and DaimlerChrysler human data sets with detailed comparisons to the existing methods.

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. In addition, the approach can be combined with any boosting method. During our experiments, we implemented LogitBoost, GentleBoost, and AdaBoost classifiers on Riemannian manifolds using LDA, decision stumps, and linear SVMs as weak learners. The results of the methods were comparative. Due to its simplicity (training time and ease of implementation) and slightly better performance, we presented the LogitBoost algorithm.

In the future, we are planning to extend our research in two directions. First, the method will be generalized for the multiclass object detection task, where instead of a binary classifier, a multiclass learning scheme will be investigated on Riemannian manifolds. Second, we are planning to test our algorithm on different Riemannian manifolds that commonly occur in computer vision problems, such as rigid motion estimation on a special euclidean group $SE(n)$ or shape space descriptors on the Grassmann manifold $G_{n,k}$.

Acknowledgments

The authors would like to thank Mitsubishi Electric Research Labs, Cambridge, Massachusetts, for supporting this study and Navneet Dalal, Bill Triggs, Qiang Zhu, Shai Avidan, Stefan Munder, and Dariu Gavrila for providing the data sets and the results of their experiments.

References

[1] S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," *Proc. European Conf. Computer Vision (ECCV '03),* pp. 113-127, 2003.

[2] E. Begelfor and M. Werman, "Affine Invariance Revisited," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '06),* vol. 2, pp. 2087-2094, 2006.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 4, pp. 509-522, Apr. 2002.

[4] A. Berg, T. Berg, and J. Malik, "Shape Matching and Object Recognition Using Low Distortion Correspondence," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '05)*, pp. 26-33, 2005.

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

[6] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, pp. 603-619, 2002.

[7] C. Cortes and V. Vapnik, "Support Vector Networks," *Machine Learning,* vol. 20, pp. 273-297, 1995.

[8] N. Dalal, "Finding People in Images and Videos," PhD dissertation, Inst. Nat'l Polytechnique de Grenoble, July 2006.

[9] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '05),* vol. 1, pp. 886-893, 2005.

[10] G. Dorkó and C. Schmid, "Selection of Scale-Invariant Parts for Object Class Recognition," *Proc. Ninth Int'l Conf. Computer Vision (ICCV '03),* pp. 634-640, 2003.

[11] P.F. Felzenszwalb and D.P. Huttenlocher, "Pictorial Structures for Object Recognition," *Int'l J. Computer Vision,* vol. 61, no. 1, pp. 55-79, 2005.

[12] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '03),* pp. 264-271, 2003.

[13] P.T. Fletcher and S. Joshi, "Riemannian Geometry for the Statistical Analysis of Diffusion Tensor Data," *Signal Processing,* vol. 87, no. 2, pp. 250-262, 2007.

[14] W. Förstner and E. Gülch, "A Fast Operator for Detection and Precise Location of Distinct Points, Corners and Centres of Circular Features," *Proc. Intercommission Conf. Fast Processing of Photogrammetric Data,* pp. 281-305, 1987.

[15] W. Förstner and B. Moonen, "A Metric for Covariance Matrices," technical report, Dept. of Geodesy and Geoinformatics, Stuttgart Univ., 1999.

[16] J. Friedman, T. Hastie, and R. Tibshirani, "Additive Logistic Regression: A Statistical View of Boosting," *Annals of Statistics,* vol. 28, no. 2, pp. 337-407, 2000.

[17] D. Gavrila and V. Philomin, "Real-Time Object Detection for Smart Vehicles," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '99),* pp. 87-93, 1999.

[18] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Fourth Alvey Vision Conf.,* pp. 147-151, 1988.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation,* second ed. Prentice Hall, 1998.

[20] S. Ioffe and D.A. Forsyth, "Probabilistic Methods for Finding People," *Int'l J. Computer Vision,* vol. 43, no. 1, pp. 45-68, 2001.

[21] T. Kadir and M. Brady, "Scale, Saliency and Image Description," *Int'l J. Computer Vision,* vol. 45, no. 2, pp. 83-105, 2001.

[22] H. Karcher, "Riemannian Center of Mass and Mollifier Smoothing," *Comm. Pure and Applied Math.,* vol. 30, pp. 509-541, 1977.

[23] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian Detection in Crowded Scenes," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '05),* vol. 1, pp. 878-885, 2005.

[24] Y. Li, Y. Tsin, Y. Genc, and T. Kanade, "Object Detection Using 2D Spatial Ordering Constraints," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '05),* pp. 711-718, 2005.

[25] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[26] K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple Object Class Detection with a Generative Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '06),* vol. 1, pp. 26-36, 2006.

[27] K. Mikolajczyk and C. Schmid, "Indexing Based on Scale Invariant Interest Points," *Proc. Eighth Int'l Conf. Computer Vision (ICCV '01),* pp. 525-531, 2001.

[28] K. Mikolajczyk, C. Schmid, and A. Zisserman, "Human Detection Based on a Probabilistic Assembly of Robust Part Detectors," *Proc. Eighth European Conf. Computer Vision (ECCV '04),* vol. 1, pp. 69-81, 2004.

[29] M. Moakher, "A Differential Geometric Approach to the Geometric Mean of Symmetric Positive-Definite Matrices," *SIAM J. Matrix Analysis and Applications,* vol. 26, pp. 735-747, 2005.

[30] A. Mohan, C. Papageorgiou, and T. Poggio, "Example-Based Object Detection in Images by Components," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 4, pp. 349-360, Apr. 2001.

[31] S. Munder and D.M. Gavrila, "An Experimental Study on Pedestrian Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, pp. 1863-1868, 2006.

[32] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer, "Weak Hypotheses and Boosting for Generic Object Detection and Recognition," *Proc. Eighth European Conf. Computer Vision (ECCV '04),* pp. 71-84, 2004.

[33] A. Opelt, A. Pinz, and A. Zisserman, "Incremental Learning of Object Detectors Using a Visual Shape Alphabet," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '06),* vol. 1, pp. 3-10, 2006.

[34] P. Papageorgiou and T. Poggio, "A Trainable System for Object Detection," *Int'l J. Computer Vision,* vol. 38, no. 1, pp. 15-33, 2000.

[35] X. Pennec, P. Fillard, and N. Ayache, "A Riemannian Framework for Tensor Computing," *Int'l J. Computer Vision,* vol. 66, no. 1, pp. 41-66, 2006.

[36] F. Porikli, "Integral Histogram: A Fast Way to Extract Histograms in Cartesian Spaces," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '05),* vol. 1, pp. 829-836, 2005.

[37] F. Porikli, O. Tuzel, and P. Meer, "Covariance Tracking Using Model Update Based on Lie Algebra," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '06),* vol. 1, pp. 728-735, 2006.

[38] R. Ronfard, C. Schmid, and B. Triggs, "Learning to Parse Pictures of People," *Proc. Seventh European Conf. Computer Vision (ECCV '02),* vol. 4, pp. 700-714, 2002.

[39] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, pp. 22-38, 1998.

[40] P. Sabzmeydani and G. Mori, "Detecting Pedestrians by Learning Shapelet Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR),* 2007.

[41] R.E. Schapire, "The Boosting Approach to Machine Learning, an Overview," *Proc. MSRI Workshop Nonlinear Estimation and Classification,* 2002.

[42] P. Simard, L. Bottou, P. Haffner, and Y. LeCun, "Boxlets: A Fast Convolution Algorithm for Signal Processing and Neural Networks," *Proc. Conf. Advances in Neural Information Processing Systems II,* pp. 571-577, 1998.

[43] L. Sirovitch and M. Kirby, "Low-Dimensional Procedure for the Characterization of Human Faces," *J. Optical Soc. of America,* vol. 2, pp. 519-524, 1987.

[44] R. Subbarao and P. Meer, "Nonlinear Mean Shift for Clustering over Analytic Manifolds," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '06),* vol. 1, pp. 1168-1175, 2006.

[45] K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, pp. 39-51, 1998.

[46] A. Torralba, K. Murphy, and W. Freeman, "Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '04),* pp. 762-769, 2004.

[47] M. Turk and A. Pentland, *Face Recognition Using Eigenfaces,* pp. 586-591, 1991.

[48] O. Tuzel, F. Porikli, and P. Meer, "Region Covariance: A Fast Descriptor for Detection and Classification," *Proc. Ninth European Conf. Computer Vision (ECCV '06),* vol. 2, pp. 589-600, 2006.

[49] O. Tuzel, R. Subbarao, and P. Meer, "Simultaneous Multiple 3D Motion Estimation via Mode Finding on Lie Groups," *Proc. 10th Int'l Conf. Computer Vision (ICCV '05),* vol. 1, pp. 18-25, 2005.

[50] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '01),* vol. 1, pp. 511-518, 2001.

[51] P. Viola, M. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '03),* vol. 1, pp. 734-741, 2003.

[52] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. Sixth European Conf. Computer Vision (ECCV '00),* pp. 18-32, 2000.

[53] Q. Zhu, S. Avidan, M.C. Yeh, and K.T. Cheng, "Fast Human Detection Using a Cascade of Histograms of Oriented Gradients," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR '06),* vol. 2, pp. 1491-1498, 2006.

Oncel Tuzel received the BS and the MS degrees in computer engineering from the Middle East Technical University, Ankara, Turkey, in 1999 and 2002. He is currently a PhD student in the Department of Computer Science, Rutgers University, Piscataway, New Jersey. His research interests are in computer vision, computer graphics, machine learning, and statistical pattern recognition. He has coauthored more than 10 papers in the area of computer vision and holds several patents awarded or pending. He was on the program committee of several international conferences such as CVPR, ICCV, and ECCV. He received the best paper runner-up award at the IEEE Computer Vision and Pattern Recognition Conference in 2007. He is a student member of the IEEE and the IEEE Computer Society.

Fatih Porikli received the PhD degree in video object segmentation from Polytechnic University, Brooklyn, New York. He is a principal research scientist and project manager at Mitsubishi Electric Research Labs (MERL), Cambridge, Massachusetts. Before joining MERL in 2000, he developed aerial image analysis applications at Hughes Research Laboratories, California, in 1999 and 3D-stereoscopic systems at AT&T Research Laboratories, New Jersey, in 1997. His research concentrated on computer vision, online learning and classification, robust optimization, multimedia processing, and video mining with many commercial applications ranging from surveillance to medical to intelligent transportation systems. He received the R&D 100 Scientist of the Year Award in 2006. He won the best paper runner up award at the 2007 IEEE International Conference on Computer Vision and Pattern Recognition. He authored more than 70 technical publications and has applied for more than 40 patents. He is serving as an associate editor for the *Journal of Machine Vision Applications*, *EURASIP Journal on Image and Video Processing*, and *Journal of Real-Time Imaging*. He is a senior member of the IEEE, the ACM, and the SPIE—The International Society for Optical Engineering.

Peter Meer received the Dipl Engn degree in electrical engineering from the Bucharest Polytechnic Institute, Romania, in 1971, and the DSc degree in electrical engineering from the Technion, Israel Institute of Technology, Haifa, Israel, in 1986. From 1971 to 1979, he was with the Computer Research Institute, Cluj, Romania, working on the R&D of digital hardware. Between 1986 and 1990, he was an assistant research scientist at the Center for Automation Research, University of Maryland, College Park. In 1991, he joined the Department of Electrical and Computer Engineering, Rutgers University, Piscataway, New Jersey, where he is currently a professor. He has held visiting appointments in Japan, Korea, Sweden, Israel, and France and was on the organizing committees of numerous international workshops and conferences. He was an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence* between 1998 and 2002, a member of the editorial board of *Pattern Recognition* between 1990 and 2005, and a guest editor of *Computer Vision and Image Understanding* for a special issue on robust statistical techniques in image understanding in 2000. He is the coauthor of an award-winning paper in pattern recognition in 1989, the best student paper in 1999, the best paper in 2000, and the runner-up paper in 2007 of the IEEE Conference on Computer Vision and Pattern Recognition. His research interest is in the application of modern statistical methods to image understanding problems. He is a senior member of the IEEE.