Pairwise Kernels for Human Interaction Recognition

Saeid Motiian, Ke Feng, Harika Bharthavarapu, Sajid Sharlemin,
and Gianfranco Doretto

Lane Department of Computer Science and Electrical Engineering
West Virginia University, Morgantown, WV 26506, USA
{samotiian,kfeng,habharthavarapu,sasharlemin}@mix.wvu.edu,
gianfranco.doretto@mail.wvu.edu

Abstract. In this paper we model binary people interactions by forming temporal interaction trajectories, under the form of a time series, coupling together the body motion of each individual as well as their proximity relationships. Such trajectories are modeled with a non-linear dynamical system (NLDS). We develop a framework that entails the use of so-called pairwise kernels, able to compare interaction trajectories in the space of NLDS. To do so we address the problem of modeling the Riemannian structure of the trajectory space, and we also prove that kernels have to satisfy certain symmetry properties, which are peculiar of this interaction modeling framework. Experiment results show that this approach is quite promising, as it is able to match and improve state-of-the-art classification and retrieval accuracies on two human interaction datasets.

1 Introduction

Recognizing human interactions from video is an important step forward towards the long-term goal of performing scene understanding fully automatically. Recent years have seen a concentration of works revolving around the problem of recognizing single-person actions, as well as group activities (see [1] and references therein). On the other hand, the area of modeling the interactions between two people is still relatively unexplored. Only recently, more realistic interaction datasets [2,3] have become available, and triggered the development of more sophisticated approaches [4,5,6,7].

In this paper we aim at developing a modeling framework leading to an approach that is fast, and that could become a building block for analyzing the behavior of a larger crowd in a scene, monitored by a network of cameras. We make the assumption that people in the scene are been tracked. This allows to analyze the spatiotemporal volume around each person and to extract relevant motion features. At the same time, the tracking information of a pair of individuals enables the extraction of a set of proxemics cues, which coupled with the motion cues form *interaction trajectories*.

We model interaction trajectories as the output of non-linear dynamical systems (NLDS), and reduce the problem of recognizing human interactions to the problem of discriminating between NLDSs. This requires designing special kernels that satisfy certain properties. In particular, (a) they have to take into account the geometry of the space where the interaction trajectories are defined, and (b) they have to satisfy certain symmetry properties, which are induced by the fact that we are modeling people interactions. We address (a) and (b) by carefully exploiting kernel construction techniques,

and by clearly showing that kernels for recognizing interaction trajectories should belong to a subcategory of the so-called *pairwise kernels*, and in particular they should satisfy the *balanced* property. A positive side effect of this framework is that by using pairwise symmetric and balanced kernels not only one can boost performance, but also is possible to significantly reduce the training time, since there is no need to use a symmetric training dataset, which has double the size of a regular one.

In § 2 we describe how human interactions can be represented by interaction trajectories, and introduce a new efficient motion feature, called motion histogram. In § 3 we pose the human interaction recognition problem and identify the challenges it implies. In § 4 we explain how interaction trajectories are represented by NLDSs. In § 5 we explain how to design kernels for comparing interaction trajectories, while addressing the challenges outlined in § 3. § 6 shows classification, and retrieval experiments where several proposed kernels are tested, validating the framework from the theoretical perspective, as well as practical by achieving very promising results.

2 Representation of human interactions

Given a sequence of images $\{I_t\}_{t=1}^{T}$, depicting two, or more people, we are interested in defining a representation for describing a potential interaction between two individuals. At every frame the bounding box delimiting the region of each person is assumed to be given (e.g., through the use of a person tracker [8], as it is typically done in video surveillance settings). For the i-th person in the video sequence, at every time t the bounding box is used to extract features aiming at describing the body motion.

From each bounding box two features are computed. The first one is the *histogram of oriented optical flow* (HOOF) [9], $\mathbf{h}_{i,t}$. It captures the motion between two consecutive frames. In addition, we introduce a feature called *motion histogram* (MH), which summarizes the motion trajectory of the past $\tau - 1$ frames (where $\tau > 1$). It requires the computation of the *motion*, or *frequency image* [10], $M_t \doteq \sum_{k=1}^{\tau-1} \eta(I_t - I_{t-k})$, where $\eta(z) = 1$ if $|z| > \delta$, otherwise $\eta(z) = 0$. Here δ is a threshold parameter to be set. Therefore, the motion histogram of person i at frame t, $\mathbf{m}_{i,t}$, is computed by binning the motion image inside the bounding box of the person. Both histograms are scale and direction invariant, as well as fairly robust to background noise, besides being fast to compute. Figure 1 shows a couple of examples of motion images with the corresponding MH features.

Eventually, the i-th person is represented by the sequence of HOOF and MH features $\mathbf{h}_i \doteq \{\mathbf{h}_{i,t}\}_{t=1}^{T}$, and $\mathbf{m}_i \doteq \{\mathbf{m}_{i,t}\}_{t=1}^{T}$, respectively, where $\mathbf{h}_{i,t}$ and $\mathbf{m}_{i,t}$ are normalized histograms made of b bins, $\mathbf{h}_{i,t} \doteq [h_{i,t;1}, \cdots, h_{i,t;b}]^\top$, and made of τ bins, $\mathbf{m}_{i,t} \doteq [m_{i,t;0}, m_{i,t;1}, \cdots, m_{i,t;\tau-1}]^\top$, where bin 0 has been added to account for the case of absence of motion.

In order to analyze the interaction between person i and person j, proxemics cues play an important discriminative role (e.g., person i cannot be hugging person j if they are far enough apart). That information here is captured by the Euclidean distance between the position $\mathbf{p}_{i,t}$ of person i, and the position $\mathbf{p}_{j,t}$ of person j, given by $d_{ij,t} \doteq \|\mathbf{p}_{i,t} - \mathbf{p}_{j,t}\|_2$. When the camera calibration is known and people tracking is performed on the ground-plane, the person position and velocity are readily available.

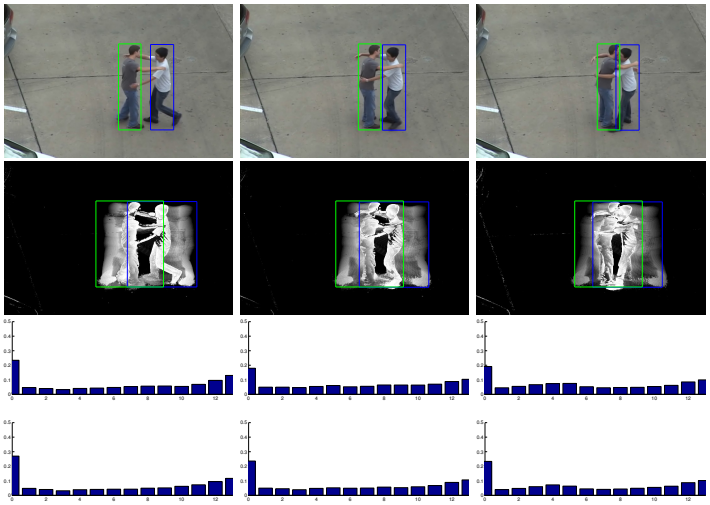

Fig. 1. From top to bottom: frames, motion images, and corresponding motion histograms of the green (top) and blue (bottom) boxes, for the UT-Interaction dataset [2].

If this is not the case, one can characterize proximity by computing the distance in the image domain, and performing a normalization based on the people size. Even if doing so is not view invariant, § 6 shows that this information still significantly increases the classification accuracy for the tested datasets. Other important cues include relative velocity and gaze direction between person i and j. We defer the use of those to future work.

Given the motion, described by $(\mathbf{h}_i, \mathbf{m}_i)$ and $(\mathbf{h}_j, \mathbf{m}_j)$, of person i and j, and their proximity described by $d_{ij} \doteq \{d_{ij,t}\}$, their *interaction trajectory* is the temporal sequence $\mathbf{y}_{ij} \doteq \{\mathbf{y}_{ij,t}\}_{t=1}^{T}$, where $\mathbf{y}_{ij,t} \doteq [\mathbf{h}_{i,t}^\top, \mathbf{m}_{i,t}^\top, \mathbf{h}_{j,t}^\top, \mathbf{m}_{j,t}^\top, d_{ij,t}]^\top$.

3 Recognizing human interactions

The interaction trajectory $\mathbf{y}_{ij}$ is a temporal sequence, and can be seen as a section of the realization of a stochastic process, which fully describes the dynamics of an interaction. Therefore, recognizing interactions is cast as a problem of recognizing stochastic processes. Under mild conditions, when $\mathbf{y}_{ij,t}$ is defined on an Euclidean space, it can be modeled as the output of a linear dynamical system (LDS), and several distances and kernels for comparing them have been proposed [11,12].

The problem we set out to address presents a couple of unique challenges. First, $\mathbf{y}_{ij,t}$ does not assume values in an Euclidean space but in a Riemannian manifold with a nontrivial structure, which is $\mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+$. In particular, $\mathbb{H}_b$ is the space of normalized histograms, which are probability mass functions satisfying the constraints $\sum_{k=1}^{b} h_{t;k} = 1$, and $h_{t;k} \geq 0$, $\forall i \in \{1, \cdots, b\}$; and similarly, $\mathbb{H}_\tau$ is the space of normalized histograms with τ bins.

The second challenge relates to the symmetry of the input feature space, which is peculiar to modeling interactions. In particular, a recognition schema entails the definition of a decision function $f : \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+ \to \mathbb{R}$, which will predict whether person i and j are engaging in a certain interaction (i.e., $f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) > 0$), or not (i.e., $f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) < 0$). Therefore, given that no person ordering is imposed a priori, the decision function is expected to be symmetric with respect to i and j, i.e.,

$$f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) = f(\mathbf{h}_j, \mathbf{m}_j, \mathbf{h}_i, \mathbf{m}_i, d_{ji}) \ . \tag{1}$$

In § 4 we summarize a general framework for modeling and comparing temporal sequences that do not assume values in an Euclidean space, which is based on kernelizing the output of an LDS, giving rise to a non-linear dynamical system (NLDS) representation. In § 5 we propose a family of so-called *pairwise kernels* that takes into account the Riemannian structure of the input feature space as well as its symmetry.

4 Modeling temporal sequences with kernel NLDSs

A stochastic process can be modeled as the output of a dynamical system. Therefore, one can compare processes by comparing dynamical system models. For second-order stationary processes assuming values in an Euclidean space, the model of choice is an LDS [13], and methods for comparing LDSs include geometric distances [14], algebraic kernels [12], and information theoretic metrics [15]. For stationary processes assuming values in a non-Euclidean space like $\mathbb{H}_b$, or the space of binary values, suitable extensions have been proposed in [9], and [16], respectively, which are based on the ideas summarized in this section.

Given a temporal sequence $\{\mathbf{y}_t\}_{t=1}^T$, assuming values in a non-Euclidean space $\mathcal{S}$, let us consider the Mercer kernel $K(\mathbf{y}_t, \mathbf{y}'_t) = \Phi(\mathbf{y}_t)^\top \Phi(\mathbf{y}'_t)$, where $\Phi(\cdot)$ is mapping $\mathcal{S}$ to $\mathcal{H}$, a Reproducing Kernel Hilbert Space (RKHS) [17]. We assume that $\{\mathbf{y}_t\}$ is mapped to a sequence $\{\Phi(\mathbf{y}_t)\}$, which can be modeled as the output of an LDS, given by

$$\begin{cases} \mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{v}_t \ , \\ \Phi(\mathbf{y}_t) = C\mathbf{x}_t + \mathbf{w}_t \ . \end{cases} \tag{2}$$

Here $\mathbf{x}_t \in \mathbb{R}^n$ is the state of the LDS at time t, $A \in \mathbb{R}^{n \times n}$ describes the dynamics of the state evolution, and the system noise $\mathbf{v}_t$ is zero-mean i.i.d. Gaussian distributed with appropriate covariance. In addition, (2) differs from a traditional LDS in that C may not be a matrix but a linear operator $C : \mathbb{R}^n \to \mathcal{H}$ to account that $\mathcal{H}$ could be an infinite dimensional space. The observation noise $\mathbf{w}_t$ is also modeled as a zero-mean i.i.d. Gaussian process with appropriate dimension and covariance, which is independent from $\mathbf{v}_t$.

It is possible to extend the procedure developed in [18] for estimating the parameters of LDSs to the case of NLDS models like (2). This is done by substituting the PCA applied to the temporal sequence with a Kernel PCA (KPCA) [17], as it is shown in [19]. In particular, given the sequence $\{\mathbf{y}_t\}$ and the kernel K, [19] shows how to estimate the matrix A, and a representation of the linear operator C, under the form of kernel principal components. The c-th component is defined by the map $\Phi(\cdot)$, and by the

KPCA weight vector $\alpha_c \doteq v_c/\sqrt{\lambda_c}$, where λ_c and v_c are the c-th largest eigenvalue and eigenvector of the kernel matrix between the zero-mean data in the high-dimensional space, computed as $(I - \frac{1}{T}\mathbf{e}\mathbf{e}^\top)K(I - \frac{1}{T}\mathbf{e}\mathbf{e}^\top)$, where $\mathbf{e} = [1, \cdots, 1]^\top \in \mathbb{R}^T$, and $[K]_{st} = K(\mathbf{y}_s, \mathbf{y}_t)$. It turns out that this representation of C is enough to define kernels for comparing NLDSs of the type in (2).

Recently, a family of Binet-Cauchy kernels for LDSs has been introduced in [12], and in [9] it has been extended for NLDSs like (2). In particular, the Binet-Cauchy trace kernel for NLDS is the expected value of an infinite series of weighted inner products between the outputs after embedding them into the high-dimensional (possibly infinite) space using the map $\Phi(\cdot)$. More precisely

$$K_{NLDS}(\{\mathbf{y}_t\}_{t=1}^{\infty}, \{\mathbf{y}'_t\}_{t=1}^{\infty}) \doteq E\left[\sum_{t=1}^{\infty} \lambda^t \Phi(\mathbf{y}_t)^\top \Phi(\mathbf{y}'_t)\right] = E\left[\sum_{t=1}^{\infty} \lambda^t K(\mathbf{y}_t, \mathbf{y}'_t)\right] , \tag{3}$$

where $0 < \lambda < 1$, and the expectation of the infinite sum of the inner products is taken w.r.t. the joint probability distribution of $\mathbf{v}_t$ and $\mathbf{w}_t$. The kernel (3) can be computed in closed form, and it requires the computation of the infinite sum $P = \sum_{t=1}^{\infty} \lambda^t (A^T)^\top F A'^\top$, where $F = \tilde{\alpha} S \tilde{\alpha}'$, and the columns of $\tilde{\alpha}$ and $\tilde{\alpha}'$ are the centered KPCA weight vectors of $\{\mathbf{y}_t\}$ and $\{\mathbf{y}'_t\}$, given by $\tilde{\alpha}_c = \alpha_c - \frac{\mathbf{e}^\top \alpha_c}{T}\mathbf{e}$, and $\tilde{\alpha}'_d = \alpha'_d - \frac{\mathbf{e}^\top \alpha'_d}{T'}\mathbf{e}$, respectively. S instead is such that $[S]_{st} = K(\mathbf{y}_s, \mathbf{y}'_t)$, where $s \in \{1, \cdots, T\}$, and $t \in \{1, \cdots, T'\}$. If $\lambda \|A\| \|A'\| < 1$, where $\|\cdot\|$ is a matrix norm, then P can be computed by solving the corresponding Sylvester equation $P = \lambda A^\top P A' + F$.

Given P, kernel (3) can be computed in closed form provided that the covariances of the system noise, the observation noise, and the initial state are available. On the other hand, as [9] points out, for recognition of phenomena that are assumed to be made by one or multiple cycles of a temporal sequence, we want to use a kernel that is independent from the initial state and the noise processes. Therefore, the original kernel (3) is simplified to K^{σ}_{NLDS}, which is a kernel only on the dynamics of the NLDS, and is given by the maximum singular value of P, i.e.,

$$K^{\sigma}_{NLDS} = \max \sigma(P) \ . \tag{4}$$

For more details about the estimation of the NLDS model parameters, and about the derivation of kernel (4) the reader is referred to [18,19,9].

5 Pairwise kernels for recognizing interaction trajectories

In this section we intend to use the framework described in § 4 to model, compare, classify, and rank interaction trajectories. § 3 has pointed out that trajectories live in a non-Euclidean space with a special symmetry. At the same time, the effectiveness of modeling them with (2) depends upon how well the kernel $K(\mathbf{y}_{ij,t}, \mathbf{y}'_{ij,t})$ is able to "map" the non-Euclidean input feature space $\mathcal{S}$ to a RKHS $\mathcal{H}$. Therefore, here we propose a few strategies for designing the kernel K.

CLASSIFICATION ACCURACY - SET 1						
Kernel/Class	Hug	Kick	Push	Punch	Hand Shake	AVG
	No Proximity					
k_S	75.00	75.00	46.15	33.33	75.00	60.65
$K_H^{TL}(k_S)$	83.33	75.00	61.53	41.66	91.66	70.49
$K_H^{DS}(k_S)$	83.33	75.00	38.46	33.33	91.66	63.93
$K_H^{TL}(k_h k_m)$	83.33	83.33	84.61	8.33	100	70.49
$K_H^{DS}(k_h k_m)$	83.33	75.00	38.46	33.33	91.66	63.93
	With Proximity					
RBF	100	100	76.92	50.00	83.33	81.96
$k_S k_d$	83.33	83.33	61.53	41.66	83.33	70.49
$K_H^{TL}(k_S)k_d$	91.66	83.33	76.92	91.66	100	88.52
$K_H^{DS}(k_S)k_d$	100	83.33	69.23	50	91.66	78.68
$K_H^{TL}(k_h k_m)k_d$	100	100	69.23	91.66	100	**91.80**
$K_H^{DS}(k_h k_m)k_d$	100	83.33	69.23	66.66	91.66	81.96

CLASSIFICATION ACCURACY - SET 2						
Kernel/Class	Hug	Kick	Push	Punch	Hand Shake	AVG
	No Proximity					
k_S	72.72	36.36	37.5	16.66	87.5	51.51
$K_H^{TL}(k_S)$	54.54	54.54	62.5	16.66	87.5	57.57
$K_H^{DS}(k_S)$	54.54	54.54	25.00	8.33	93.75	48.48
$K_H^{TL}(k_h k_m)$	72.72	63.63	50.00	50.00	62.50	59.09
$K_H^{DS}(k_h k_m)$	45.45	27.27	43.75	16.16	87.50	46.96
	With Proximity					
RBF	100	45.45	87.50	41.66	81.25	72.72
$k_S k_d$	100	54.54	81.25	41.66	83.33	72.72
$K_H^{TL}(k_S)k_d$	81.81	72.72	50.00	16.16	75.00	59.09
$K_H^{DS}(k_S)k_d$	90.90	27.27	50.00	33.33	93.75	60.60
$K_H^{TL}(k_h k_m)k_d$	100	72.72	87.50	75.00	100	**87.87**
$K_H^{DS}(k_h k_m)k_d$	100	36.36	87.50	41.66	100	75.75

Table 1. Classification accuracy for the UT-Interaction dataset [2]. For Set 1 MH features are computed with $\tau = 14$, and $\delta = 2$; HOOF features are computed with $b = 18$; NLDS order is set to $n = 8$. For Set 2 MH features are computed with $\tau = 22$, and $\delta = 5$; HOOF features are computed with $b = 24$; NLDS order is set to $n = 10$.

Since the input feature space $\mathcal{S} \doteq \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{R}_+$ is a non-Euclidean space which is a Riemannian manifold, defining K to be a linear kernel would clearly be sub-optimal. A typical approach for improving the map to a RKHS is to use a generic, top-performing non-linear kernel, such as the Gaussian radial basis function (RBF) kernel with Euclidean distance. However, in this way we do not take advantage of the known Riemannian structure of $\mathcal{S}$. One way to do so is to replace the Euclidean distance with a proper distance for the manifold $\mathcal{S}$. Unfortunately, to the authors' knowledge defining a distance on $\mathcal{S}$ is still an open problem, although for $\mathbb{H}_b$ (or $\mathbb{H}_\tau$) alone a theoretical solution exists, which is the *Fisher-Rao* metric [20]. Therefore, whenever it cannot be done otherwise, we advocate the use of kernel construction techniques [17], which take into account the fact that $\mathcal{S}$ is given by the Cartesian product of subspaces. This way allows to concentrate on each subspace separately, and exploit the known subspace geometry to the full extent.

To start, we notice that the input feature space $\mathcal{S}$ is given by the Cartesian product of the subspaces $\mathbb{H}_b \times \mathbb{H}_\tau \times \mathbb{H}_b \times \mathbb{H}_\tau$, and $\mathbb{R}_+$. Therefore, we can focus on designing a kernel for histograms K_H on the first subspace, and a kernel K_d for people distances on the second subspace. K_H and K_d can then be combined by computing their *tensor product kernel* [17], leading to

$$K \doteq (K_H \otimes K_d)(\mathbf{y}_{ij}, \mathbf{y}'_{ij}) = K_H((\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j), (\mathbf{h}'_i, \mathbf{m}'_i, \mathbf{h}'_j, \mathbf{m}'_j))K_d(d_{ij}, d'_{ij}) \,, \tag{5}$$

where we have dropped the time subscript t to lighten the notation. Intuitively, a kernel defines similarity in an input space. Kernel (5) yields a high value only if the instances in each subspace have high similarity with the corresponding instances in the same subspace. This is desirable because the classification of interactions should be based on the similarity across not only the motion features, but also the proximity cues, as it is explained in § 2.

CLASSIFICATION ACCURACY						
Kernel/Class	HS	HF	HG	KS	NG	AVG
	No Proximity					
k_S	33.33	51.72	36.36	30.43	52.00	40.77
$K_H^{TL}(k_S)$	19.05	58.62	18.18	47.83	60.00	40.74
$K_H^{DS}(k_S)$	0	62.07	31.82	30.43	88.00	42.46
$K_H^{TL}(k_h k_m)$	38.10	44.83	31.82	30.43	44.00	37.84
$K_H^{DS}(k_h k_m)$	9.52	41.38	31.82	43.48	68.00	38.84
	With Proximity					
RBF	4.76	65.52	59.09	73.91	60.00	52.66
$k_S k_d$	19.05	62.07	86.36	73.91	56.00	59.48
$K_H^{TL}(k_S)k_d$	19.05	79.31	81.82	65.22	64.00	61.88
$K_H^{DS}(k_S)k_d$	23.81	51.72	90.91	78.26	64.00	61.74
$K_H^{TL}(k_h k_m)k_d$	28.57	79.31	86.36	65.22	64.00	**64.69**
$K_H^{DS}(k_h k_m)k_d$	38.10	51.72	81.82	73.91	72.00	63.51

RETRIEVAL PRECISION					
Kernel/Class	HS	HF	HG	KS	AVG
	No Proximity				
k_S	0.239	0.316	0.293	0.402	0.314
$K_H^{TL}(k_S)$	0.208	0.335	0.265	0.498	0.330
$K_H^{DS}(k_S)$	0.199	0.300	0.267	0.424	0.300
$K_H^{TL}(k_h k_m)$	0.267	0.264	0.277	0.523	0.330
$K_H^{DS}(k_h k_m)$	0.222	0.296	0.263	0.422	0.302
	With Proximity				
$k_S k_d$	0.310	0.319	0.559	0.541	0.427
$K_H^{TL}(k_S)k_d$	0.334	0.333	0.485	0.482	0.412
$K_H^{DS}(k_S)k_d$	0.339	0.351	0.538	0.483	0.424
$K_H^{TL}(k_h k_m)k_d$	0.342	0.357	0.551	0.525	0.439
$K_H^{DS}(k_h k_m)k_d$	0.351	0.338	0.554	0.540	**0.440**

Table 2. Classification accuracy, and video retrieval average precision for the TVHI dataset [3]. MH features are computed with $\tau = 5$, and $\delta = 3$; HOOF features are computed with $b = 10$; NLDS order is set to $n = 10$.

For kernel K_d we observe that d_{ij} belongs to $\mathbb{R}_+$, and therefore we simply chose a Gaussian RBF kernel, given by

$$K_d(d_{ij}, d'_{ij}) \doteq \exp(-\gamma |d_{ij} - d'_{ij}|^2) \ . \tag{6}$$

For kernel K_H, we note that it is a so-called *pairwise kernel* [21], because it is such that $K_H : (\mathcal{X}_H \times \mathcal{X}_H) \times (\mathcal{X}_H \times \mathcal{X}_H) \to \mathbb{R}$, where $\mathcal{X}_H \doteq \mathbb{H}_b \times \mathbb{H}_\tau$, and it could be used to support *pairwise classification*, which aims at deciding whether the examples of a pair $(a, b) \in \mathcal{X}_H \times \mathcal{X}_H$ belong to the same class or not. The requirement of being positive semidefinite implies that K_H satisfies the following *symmetry* property

$$K_H((a, b), (a', b')) = K_H((a', b'), (a, b)) \ , \tag{7}$$

for all $a, b, a', b' \in \mathcal{X}_H$. By using kernel construction techniques based on direct sum and tensor product of kernels, given the kernel $k_H : \mathcal{X}_H \times \mathcal{X}_H \to \mathbb{R}$, one can build the following pairwise versions of K_H

$$K_H^D = (k_H \oplus k_H)(a, b, a', b') = k_H(a, a') + k_H(b, b') \ , \tag{8}$$
$$K_H^T = (k_H \otimes k_H)(a, b, a', b') = k_H(a, a')k_H(b, b') \ , \tag{9}$$

which obviously satisfy the symmetric property.

We now verify whether by using the kernels defined in (8) and (9) it is possible to construct decision functions for interaction trajectories, which are supposed to satisfy the symmetry property (1). We plan to learn decision functions f with a SVM that exploits the general kernel (3). Therefore, they will assume the form

$$f(\{a_{i,t}, a_{j,t}, d_{ij,t}\}) \doteq \sum_{u,v} \alpha_{uv} \ell_{uv} K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) + \beta \ , \tag{10}$$

where α_{uv}, ℓ_{uv}, and β are the usual SVM parameters [17], and $a_{i,t} = (\mathbf{h}_{i,t}, \mathbf{m}_{i,t}) \in \mathcal{X}_H$, and $a_{j,t} = (\mathbf{h}_{j,t}, \mathbf{m}_{j,t}) \in \mathcal{X}_H$. More importantly, (10) tells us that the symmetry property (1) imposes that

$$\begin{aligned} K_{NLDS}(\{a_{i,t}, a_{j,t}, d_{ij,t}\},\{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) = \\ K_{NLDS}(\{a_{j,t}, a_{i,t}, d_{ji,t}\}, \{a'_{u,t}, a'_{v,t}, d'_{uv,t}\}) \ , \end{aligned} \tag{11}$$

for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$, and $d_{ij,t}, d'_{uv,t} \in \mathbb{R}_+$. In turn, (11) induces a symmetry property on the kernel (5) through (3), which is given by

$$K((a_{i,t}, a_{j,t}, d_{ij,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) = K((a_{j,t}, a_{i,t}, d_{ji,t}), (a'_{u,t}, a'_{v,t}, d'_{uv,t})) \ , \tag{12}$$

and finally, since $d_{ij,t} = d_{ji,t}$ and $d_{uv,t} = d_{vu,t}$, (12) imposes on K_H the following relationship

$$K_H((a_{i,t}, a_{j,t}), (a'_{u,t}, a'_{v,t})) = K_H((a_{j,t}, a_{i,t}), (a'_{u,t}, a'_{v,t})) \ , \tag{13}$$

to be valid for all $a_{i,t}, a_{j,t}, a'_{u,t}, a'_{v,t} \in \mathcal{X}_H$. Note that the relationship (13) is different than the symmetry relationship (7), and kernels that satisfy (13) are called *balanced* [21].

Unfortunately, the pairwise kernels K_H^D, and K_H^T, defined in (8) and (9), are symmetric but not balanced. Therefore, we propose to test two kernels that have been proved to have good theoretical properties [21], in that they guarantee minimal loss of information, and can be thought of as the balanced versions of K_H^D, and K_H^T. They are defined as follows

$$K_H^{DS}((a,b),(a',b')) = K_H^{SD}((a,b),(a',b')) + K_H^{ML}((a,b),(a',b')) \ , \tag{14}$$

$$K_H^{TL}((a,b),(a',b')) = \frac{1}{2}(k_H(a,a')k_H(b,b') + k_H(a,b')k_H(b,a')) \ , \tag{15}$$

where

$$K_H^{SD}((a,b),(a',b')) = \frac{1}{2}(k_H(a,a') + k_H(a,b') + k_H(b,a') + k_H(b,b')) \ , \tag{16}$$

$$K_H^{ML}((a,b),(a',b')) = \frac{1}{4}(k_H(a,a') - k_H(a,b') - k_H(b,a') + k_H(b,b'))^2 \ . \tag{17}$$

In particular, K_H^{TL} is called *tensor learning pairwise kernel* [22], whereas K_H^{DS} is called *direct sum pairwise kernel* [21].

Finally, we are left with the task of designing k_H, which is defined on the space $(\mathbb{H}_b \times \mathbb{H}_\tau) \times (\mathbb{H}_b \times \mathbb{H}_\tau)$. Since it is not required to be balanced, and both features, $\mathbf{h}_{i,t}$ and $\mathbf{m}_{i,t}$, should concur at the same time towards establishing similarity, we apply the tensor product rule to further decompose k_H into two kernels, $k_h : \mathbb{H}_b \times \mathbb{H}_b \to \mathbb{R}$ and $k_m : \mathbb{H}_\tau \times \mathbb{H}_\tau \to \mathbb{R}$, producing

$$k_H((\mathbf{h}_{i,t}, \mathbf{m}_{i,t}), (\mathbf{h}'_{i,t}, \mathbf{m}'_{i,t})) = k_h(\mathbf{h}_{i,t}, \mathbf{h}'_{i,t})k_m(\mathbf{m}_{i,t}, \mathbf{m}'_{i,t}) \ . \tag{18}$$

Both k_h and k_m are kernels for comparing histograms. There are several options for kernels in this domain, as it is outlined in [9], where it has been shown that an excellent

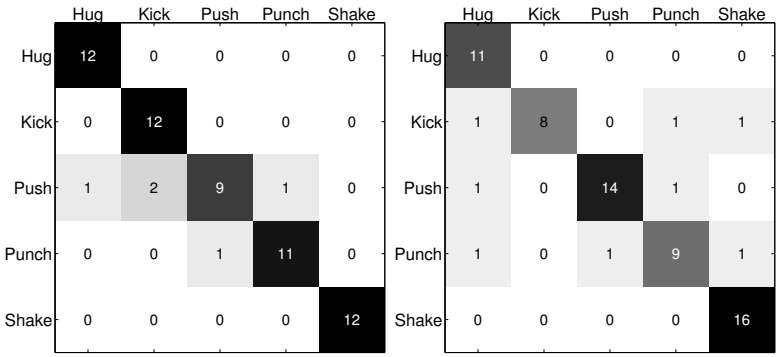

Fig. 2. Confusion matrices for the UT-Interaction dataset: Set 1 (left), and Set 2 (right).

compromise between performance and speed is given by the following Mercer kernel

$$k_S(\mathbf{h}_1, \mathbf{h}_2) = \sum_{k=1}^{b} \sqrt{h_{1,k} h_{2,k}} \ , \tag{19}$$

which is derived by taking into account that $\mathbb{H}_b$ is diffeomorphic to a subset of the hypersphere $\mathbb{S}^{b-1}$. We refer to this as the *geodesic kernel*. Both k_h and k_m are picked to be geodesic kernels for histograms with b and τ bins, respectively.

6 Experiments

We have tested our approach on two state-of-the-art human interaction datasets: the UT-Interaction dataset [2], and the TV Human Interaction (TVHI) dataset [3]. The first one contains videos of six interaction classes: *hand shake*, *hug*, *kick*, *point*, *punch*, and *push*. We have excluded the *point* class because it is representative of a single person action. The dataset is divided into Set 1 and Set 2, each consisting of 10 videos. Set 1 videos have mostly a static background, and Set 2 videos have some background motion, with some small camera motion, which makes Set 2 slightly more challenging than Set 1. In our model we assume to have people tracking information, and since the ground-truth annotation of the dataset was not providing that, we annotated the dataset with the VATIC tool [23]. The top-row of Figure 1 shows the bounding boxes obtained with this process. Also, the second row shows the same boxes with a width that is three times of the original. Those wider boxes were used to compute the MH and the HOOF features. In particular, the motion images are computed with respect to the L channel of the Lab color space, and the HOOF features are based on the optical flow computed with the OpenCV library.

The TVHI dataset has videos from 5 different classes: *hand-shakes*, *high-fives*, *hugs*, *kisses*, and *negative* examples. The length of the videos range from 30 to 600 frames. There is a great degree of variation among the videos as they are compiled from different TV shows, which makes this dataset very challenging. As people tracking information

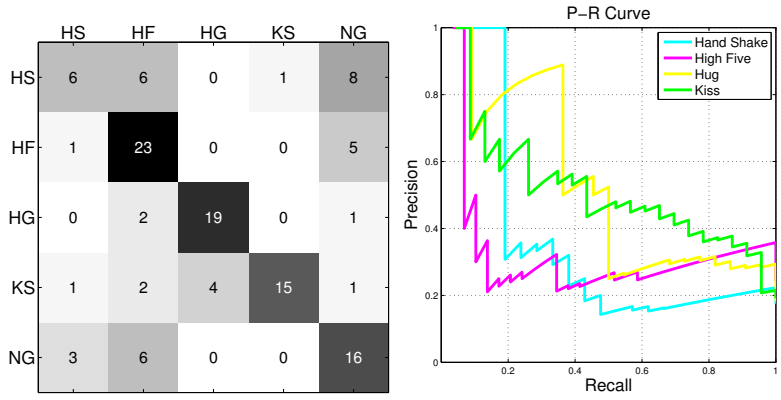

Fig. 3. Confusion matrix (left), and per-class precision-recall curves (right) for the TVHI dataset.

we were able to use the ground-truth annotations made available along with the videos, consisting of bounding boxes framing the upper bodies of all the actors in the scene. Our analysis was limited to the bounding boxes corresponding to the people interacting, and the features were extracted from boxes having a width that was double the original annotations, in order to analyze the motion in a region surrounding each person. Note that, similarly to [3], some of the original videos where not considered due to their very limited length, or due to sharp view point changes during the interaction.

We tested the kernels proposed in § 5 for classification. Different choices of K_H are evaluated, where for each of them we consider the case of interaction trajectories with or without proximity cues. Presence or absence of this information is well marked on the tables, and also on the table kernel labels, by the presence or absence of the k_d kernel (6). Since the input features $(\mathbf{h}_{i,t}, \mathbf{m}_{i,t}, \mathbf{h}_{j,t}, \mathbf{m}_{j,t})$ live in a subspace of $\mathbb{H}_{2b+2\tau}$, it is possible to test the following choices for K_H: (a) k_S, which is the geodesic kernel (19); (b) K_H^{TL} (15), where k_H is a geodesic kernel, indicated with $K_H^{TL}(k_S)$; (c) K_H^{DS} (14), where k_H is a geodesic kernel, indicated with $K_H^{DS}(k_S)$; (d) K_H^{TL} (15), where k_H is the tensor product kernel (18), indicated with $K_H^{TL}(k_h k_m)$; (e) K_H^{DS} (14), where k_H is the tensor product kernel (18), indicated with $K_H^{DS}(k_h k_m)$. Finally, for kernel K (5) we also tested a Gaussian RBF kernel with Euclidean distance.

The kernels described above were used, in conjunction with kernel (4), to train the multi-class classifier of the libSVM [24] with leave-one-out cross-validation. Table 1 shows the classification accuracy for the UT-Interaction dataset, whereas Table 2 shows the classification results for the TVHI dataset. From them we can draw a number of considerations. First, as pointed out in § 5 using an RBF kernel with Euclidean distance leads to suboptimal results. Second, we have experienced a higher degree of good performance consistency for the tensor learning pairwise kernel $K_H^{TL}(k_h k_m)$, versus the direct sum pairwise kernel $K_H^{DS}(k_h k_m)$. Third, we have verified the importace of designing kernels by taking into account the structure of the input feature space in the way that different kernels rank in terms of performance. Fourth, we have verified the

SET 1				SET 2			
Kernel/Feature	MH	HOOF	Both	Kernel/Feature	MH	HOOF	Both
$k_S k_d$	65.57	68.85	70.49	$k_S k_d$	60.60	56.06	72.72
$K_H^{TL}(k_S)k_d$	68.85	73.77	88.52	$K_H^{TL}(k_S)k_d$	50.00	54.55	59.90
$K_H^{DS}(k_S)k_d$	70.49	70.49	78.68	$K_H^{DS}(k_S)k_d$	53.03	54.55	60.60
$K_H^{TL}(k_h k_m)k_d$	-	-	**91.80**	$K_H^{TL}(k_h k_m)k_d$	-	-	**87.87**
$K_H^{DS}(k_h k_m)k_d$	-	-	81.96	$K_H^{DS}(k_h k_m)k_d$	-	-	75.75

Table 3. Classification accuracy for the UT-Interaction dataset obtained using proximity and different motion features, including only motion histograms (MH), only HOOF features, and both. Motion features are computed as indicated in Table 1.

importance in incorporating proximity information for discriminating between interactions. Fifth, the best classification accuracy is on-par or better than recently reported results [3,4,5,6,7], indicating that the approach is promising. Figure 2 and Figure 3 show the confusion matrices corresponding to the best classification accuracy.

For the TVHI dataset we have also performed a video retrieval experiment. In particular, we have converted the proposed kernels in pairwise distances, where the kernel K_{NLDS} is normalized to 1 when $\{\mathbf{y}_t\} = \{\mathbf{y}'_t\}$, by computing $\tilde{K}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) \doteq K_{NLDS}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) / \sqrt{K_{NLDS}(\{\mathbf{y}_t\}, \{\mathbf{y}_t\}) K_{NLDS}(\{\mathbf{y}'_t\}, \{\mathbf{y}'_t\})}$, and the distance between two interaction trajectories becomes $d(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) \doteq 2(1 - \tilde{K}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\})$. Table 2 and Figure 3 show the retrieval precision and the per-class precision-recall curves, as defined in [25]. It can be seen that even with such a simple approach, the results are comparable to the ones in [3]. We expect that by using the proposed kernels in a "learning to rank" approach [26], the retrieval precision would undergo a substantial increase.

Finally, for the various kernels Table 3 shows how classification performance is affected in three cases, namely when only the MH features are used, only the HOOF features are used, and when both are used. It can be seen that the proposed motion histogram features are capturing valuable motion history information, which is as discriminative as the one captured by the HOOF. Also, it is uncorrelated to the information captured by the HOOF, given the significant boost in classification accuracy.

Acknowledgments

This work was supported in part by grant 2010-DD-BX-0161, awarded by the National Institute of Justice, Office of Justice Programs, US Department of Justice.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: A review. ACM Comput. Surv. **43** (2011) 16 1
2. Ryoo, M., Aggarwal, J.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: ICCV. (2009) 1593–1600 1, 3, 6, 9

3. Patron-Perez, A., Marszalek, M., Reid, I., Zisserman, A.: Structured learning of human interactions in TV shows. IEEE TPAMI **34** (2012) 2441–2453 1, 7, 9, 10, 11
4. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV. (2011) 778–785 1, 11
5. Gaur, U., Zhu, Y., Song, B., Roy-Chowdhury, A.: A "string of feature graphs" model for recognition of complex activities in natural videos. In: Computer Vision (ICCV), 2011 IEEE International Conference on. (2011) 2595–2602 1, 11
6. Kong, Y., Jia, Y., Fu, Y.: Learning human interaction by interactive phrases. In: ECCV. (2012) 300–313 1, 11
7. Yu, G., Yuan, J., Liu, Z.: Propagative hough voting for human activity recognition. In: ECCV. (2012) 693–706 1, 11
8. Andriluka, M., Roth, S., Schiele, B.: People-tracking-by-detection and people-detection-by-tracking. In: CVPR. (2008) 1–8 2
9. Chaudhry, R., Ravichandran, A., Hager, G., Vidal, R.: Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In: CVPR. (2009) 1932–1939 2, 4, 5, 8
10. Doretto, G., Sebastian, T., Tu, P., Rittscher, J.: Appearance-based person reidentification in camera networks: problem overview and current approaches. Journal of Ambient Intelligence and Humanized Computing **2** (2011) 127–151 2
11. Saisan, P., Doretto, G., Wu, Y.N., Soatto, S.: Dynamic texture recognition. In: CVPR. Volume 2., Kauai, Hawaii, USA (2001) 58–63 3
12. Vishwanathan, S., Smola, A., Vidal, R.: Binet-cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes. IJCV **73** (2007) 95–119 3, 4, 5
13. Ljung, L.: System identification: theory for the user. 2nd edn. Prentice-Hall, Inc. (1999) 4
14. De Cock, K., De Moor, B.: Subspace angles and distances between ARMA models. In: MTNS. (2000) 4
15. Chan, A., Vasconcelos, N.: Probabilistic kernels for the classification of auto-regressive visual processes. In: CVPR. Volume 1. (2005) 846–851 4
16. Li, W., Vasconcelos, N.: Recognizing activities by attribute dynamics. In: NIPS. (2012) 4
17. Schölkopf, B., Smola, A.: Learning with kernels: SVM, regularization, optimization, and beyond. The MIT press (2002) 4, 6, 8
18. Doretto, G., Chiuso, A., Wu, Y.N., Soatto, S.: Dynamic textures. IJCV **51** (2003) 91–109 4, 5
19. Chan, A.B., Vasconcelos, N.: Classifying video with kernel dynamic textures. In: CVPR. (2007) 1–6 4, 5
20. Srivastava, A., Jermyn, I., Joshi, S.: Riemannian analysis of probability density functions with applications in vision. In: CVPR. (2007) 1–8 6
21. Brunner, C., Fischer, A., Luig, K., Thies, T.: Pairwise support vector machines and their application to large scale problems. JMLR **13** (2012) 2279–2292 7, 8
22. Ben-Hur, A., Noble, W.S.: Kernel methods for predicting protein-protein interactions. Bioinformatics **21 Suppl 1** (2005) i38–i46 8
23. Vondrick, C., Patterson, D., Ramanan, D.: Efficiently scaling up crowdsourced video annotation. IJCV (2012) 1–21 10.1007/s11263-012-0564-1. 9
24. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology **2** (2011) 27:1–27:27 10
25. Buettcher, S., Clarke, C.L.A., Cormack, G.V.: Information Retrieval. The MIT Press (2010) 11
26. Cao, Z., Qin, T., Liu, T.Y., Tsai, M.F., Li, H.: Learning to rank: from pairwise approach to listwise approach. In: ICML. (2007) 129–136 11