Online Human Interaction Detection and Recognition With Multiple Cameras

Saeid Motiian, Farzad Siyahjani, *Student Member, IEEE*, Ranya Almohsen, and Gianfranco Doretto, *Member, IEEE*

***Abstract*— We address the problem of detecting and recognizing online the occurrence of human interactions as seen by a network of multiple cameras. We represent interactions by forming temporal trajectories, coupling together the body motion of each individual and their proximity relationships with others, and also sound whenever available. Such trajectories are modeled with kernel state-space (KSS) models. Their advantage is being suitable for the online interaction detection, recognition, and also for fusing information from multiple cameras, while enabling a fast implementation based on online recursive updates. For recognition, in order to compare interaction trajectories in the space of KSS models, we design so-called pairwise kernels with a special symmetry. For detection, we exploit the geometry of linear operators in Hilbert space, and extend to KSS models the concept of parity space, originally defined for linear models. For fusion, we combine KSS models with kernel construction and multiview learning techniques. We extensively evaluate the approach on four single view publicly available data sets, and we also introduce, and will make public, a new challenging human interactions data set that we have collected using a network of three cameras. The results show that the approach holds promise to become an effective building block for the analysis of real-time human behavior from multiple cameras.**

***Index Terms*— Detection, human behavior analysis, human interaction recognition, kernel parity space, multiple camera video surveillance, segmentation.**

I. Introduction

DETECTION and recognition of continuous activities from video are a core problem to address for enabling intelligent systems that can extract and manage content fully automatically. Recent years have seen a concentration of works revolving around the problem of recognizing single-person actions, as well as group activities [1]. On the other hand, the area of modeling the interactions between two people is still relatively unexplored. Only recently, more realistic interaction data sets [2], [3] have become available, and have triggered the development of more sophisticated approaches [4]–[8].

Besides recognizing interactions, their temporal detection, or segmentation, can support core tasks, such as more complex activity analysis [9], as well as motion capture data analysis and animation [10]. Despite a significant amount of research focusing on recognizing actions and activities, the problem of their time localization has received considerably less attention [3]. Even more so, if it has to be performed online, and enable the processing of video in real time, the main challenge is handling the complexity of the variability of data that represent activities, which is inherently multidimensional. Although some approaches have demonstrated some level of success in segmenting and recognizing activities [11]–[13], those have been developed to work offline, and only very recently new methods have been developed for detecting activities in a fully causal and online fashion [14], [15].

In this paper, we aim at developing a single modeling framework that is capable of detecting as well as recognizing human interactions in a causal, online fashion, even when multiple cameras are simultaneously monitoring the same area where the interaction is occurring. Such framework should also be fast to potentially become a building block for analyzing the behavior of a larger crowd, 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 body motion features. Concurrently, the tracking information of a pair of individuals allows the extraction of proxemics cues, which coupled with body motion cues form *interaction trajectories*.

We model interaction trajectories as the output of *kernel state-space (KSS) models*. This allows us to leverage the theories on reproducing kernel Hilbert spaces (RKHSs) [16], and on state-space models [17]. Exploiting the power of kernels allows a flexible and effective blending of heterogeneous high-dimensional features, which can be mapped onto a suitable Hilbert space where they can be easily modeled, even with linear models. Exploiting the theory on state-space models allows borrowing a number of well understood results about their estimation, and their power for doing analysis, recognition, and detection based on multidimensional temporal sequences.

We recognize human interactions by discriminating between KSS models. This requires designing kernels that satisfy certain properties. In particular: 1) they have to account for the geometry of the space where interaction trajectories are defined and 2) they have to satisfy certain symmetry properties induced by the fact that we are modeling people interactions. We address 1) and 2) by carefully exploiting kernel construction techniques, and by showing that kernels

Manuscript received December 21, 2015; revised April 5, 2016 and June 21, 2016; accepted August 22, 2016. Date of publication September 8, 2016; date of current version March 3, 2017. This work was supported by the National Institute of Justice under Grant NIJ-2010-DD-BX-0161 and Grant NIJ-2010-IJ-CX-K024. This paper was recommended by Associate Editor H. Yao.

The authors are with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV 26508 USA (e-mail: gianfranco.doretto@mail.wvu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2606998

for recognizing interaction trajectories should belong to a subcategory of the so-called *pairwise kernels*, which satisfy the *balanced* property. Those can not only boost performance, but also can significantly reduce the training time.

Using KSS models allows us to develop an online detection approach that copes with the high dimensions, as well as the complexity of the variability of interaction trajectories. In particular, we extend the notion of *parity space* (developed for detection based on linear models [18]), to be used with *kernel regression (KR)*, and KSS models, which are the Hilbert space counterparts of the linear versions. Rather than using Euclidean geometry to project data onto the parity space and reveal detection, we exploit the geometry of linear operators in Hilbert space, and derive closed form solutions for the computation of normalized test statistics, based solely on kernel evaluations. The framework is suitable to work online through the use of a temporal incremental window, and online parameter estimation techniques.

Since modeling human interactions from multiple surveillance camera views is a new subject, we introduce a new video data set recording several binary human interactions from three different views. In addition, we use it to extend and validate the detection and recognition techniques for fusing interaction trajectories acquired by multiple cameras, by leveraging kernel construction and multiview learning techniques [16], [19].

The rest of this paper is organized as follows. Section III describes how human interactions are represented by feature trajectories. Section IV describes the KSS models used for modeling interaction trajectories. Section V extends the parity space theory to KSS models to perform interaction detection. Section VI describes how KSS models are used for interaction recognition, and Section VII shows how to design kernels to perform this task. Finally, Section IX validates the approach on several data sets, including the one newly proposed, and Section X concludes this paper. A preliminary version of the content of this manuscript was previously published in [20] and [21].

II. Related Work

KSS models have been first introduced in [22] for dynamic texture recognition and subsequently have been used in [23] for action recognition. Compared with those works, we introduce a theoretically grounded framework that extends those models for temporal segmentation, and for modeling interactions. There is a body of work that exploits kernel-based methods to solve the two-sample test problem, and that applies this framework to the change point detection problem (see [24] and references therein). Those approaches work with either univariate temporal sequences or with small-dimensional sequences. However, in [15], the maximum mean discrepancy (MMD) distance [25] is used for the online temporal segmentation of actions. Mainly, our framework differs from theirs, because we can also account for the temporal correlation of sequences, and Section IX also shows that even the simpler KR model outperforms the MMD.

This paper also relates to the elegant framework introduced in [14]. Compared with them, we do not focus on the combined segmentation and early recognition. Instead, we offer a unified model to perform segmentation and recognition right at the conclusion of an interaction, given that for real-time analysis, the idea of early detection often has less importance outside of applications, such as affective computing. A related set of works is devoted to recognizing activities from partially observed videos. Reference [26] assumes that the unobserved part is at the end and focuses on activity prediction from its early stage. In [27], they assume that the unobserved part could be at any time in the video, and a sparse coding approach is used for predicting the activity from the observed video segments. In [28], instead, the unobserved part could be one person in a person–person interaction. Compared with those approaches, we do not consider unobserved video parts; instead, we aim at finding activity segments in a video by fusing multiple views observing the same occurrence of interactions.

In contrast to using partial information, methods, such as [29] and [30], propose frameworks that analyze the dynamics of the entire scene to recognize group activities while considering their correlation with individual actions and person–person interactions. In particular, in [30], inference is also combined with people tracking. In this paper, we do not analyze the global dynamic context and group activities, and focus on online processing and fusion from multiple views, and on binary people interactions that require the analysis of body motion that goes beyond tracking trajectory analysis.

Another body of work focuses on detecting 3D spatiotemporal subvolumes that might contain an activity. The approach aims at extending to time the sliding window approach for object detection in 2D images. Reference [31] finds 3D subvolumes with an extension of the deformable part-based model. Reference [32] uses multiple-instance learning on a support vector machine (SVM) to detect an activity inside 3D subvolumes, and [33] also introduces a search algorithm to detect optimal 3D subvolumes. These approaches can be computationally rather expensive, and focus more on localized type of activities, whereas our approach looks more at the potential to work in real time, with multiple camera views, and in an online fashion. To increase the temporal resolution of the analyzed activities (which include running and walking), and also the detection speed, other approaches, such as [34] and [35], developed search strategies aiming at finding a sequence of bounding boxes composing 2D $+\ t$ volumes. These approaches differ from ours, because they are not designed to work online, and to handle interactions imaged simultaneously from multiple cameras.

Finally, we mention that merely applying a generic action recognition approach to detect and recognize interactions is suboptimal, since interactions have a special symmetry as also shown in Section VII. Indeed, [2] provides a realistic interaction data set and proposes a matching function to measure similarity between two videos, and in [3], they compute two local and global descriptors using peoples head orientation.

III. Human Interaction Representation

Let us assume that $\{I_t\}_{t=1}^T$ is a video of length T, acquired by a multicamera system, which is depicting two or more people. We want to define a representation for describing

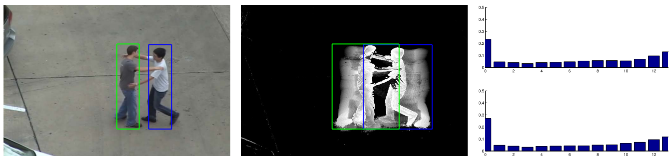


Fig. 1. From left to right: frame, motion image, and corresponding MHs of the green (top) and blue (bottom) boxes, for the UT-I data set [2].

an interaction between two individuals. At every time t, we assume that the tracking information of the ith person is available. Typically, this is obtained through the use of a multicamera multiperson tracker [36], which provides a tight bounding box indicating where the ith person is located in the image frame I_t. We consider three types of features for describing interactions. Those include *motion* (Section III-A), *proximity* (Section III-B), and *audio* features (Section III-D), whenever they are available. The representation that we are interested in has to be flexible enough to allow the use of all three types of features, and to use them in a causal fashion to enable the online detection and recognition of interactions. This is in contrast with traditional holistic representations based on the bag-of-words model [37], [38]. For this reason, we use a representation based on *feature trajectories*.

A. Motion Features

From each bounding box, two features are computed to describe the body motion. The first one is the *histogram of oriented optical flow* (HOOF) [23], $\mathbf{h}_{i,t}$, which 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 $\zeta - 1$ frames (where $\zeta > 1$). It requires the computation of the *motion*, or *frequency image* [39], $M_t \doteq \sum_{k=1}^{\zeta-1} \eta(I_t - I_{t-k})$, where $\eta(z) = 1$ if $|z| > \epsilon$, otherwise $\eta(z) = 0$. Here, ϵ is a threshold parameter to be set. Therefore, the MH 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, and fast to compute. Fig. 1 shows an example of motion image with the corresponding MH features.

Eventually, the ith 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, $\mathbf{h}_{i,t} \doteq [h_{i,t;1}, \ldots, h_{i,t;b}]^\top$, and made of c bins, $\mathbf{m}_{i,t} \doteq [m_{i,t;0}, m_{i,t;1}, \ldots, m_{i,t;c-1}]^\top$, where bin 0 accounts for the case of absence of motion.

B. Proximity Features

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. If this is not the case, one can characterize proximity by computing the distance in the image domain, normalized by the people size. Even if doing so is not view invariant, Section IX shows that it still significantly increases the classification accuracy. Other important cues include relative velocity and gaze direction between person i and j. We defer the use of those to future work.

C. Audio Features

Whenever audio is available, we represent it with mel-frequency cepstral coefficients (MFCCs) [40]. They are extracted from the MFC, which is a representation of the short-term power spectrum of sound. More precisely, the audio signal is first divided into frames using a windowing function, and the power spectrum of each frame is computed. Then, a mel filterbank is used to compute the signal energy in various frequency regions. Finally, the discrete cosine transform of the log-filtered energies is used to compute uncorrelated energy coefficients. From a video, we extract the audio signal, and we apply a sliding window with a step size corresponding to the video frame rate to compute the MFCCs, producing the audio feature trajectory $\{a_t\}_{t=1}^T$.

D. Interaction Trajectories

Given the motion of persons i and j, described by $(\mathbf{h}_i, \mathbf{m}_i)$ and $(\mathbf{h}_j, \mathbf{m}_j)$, and their proximity described by $d_{ij} \doteq \{d_{ij,t}\}$, their combined feature trajectory is the temporal sequence $\mathbf{y}_{ij} \doteq \{\mathbf{y}_{ij,t}\}_{t=1}^T$, where $\mathbf{y}_{ij,t} \doteq [\mathbf{h}_{i,t}^\top, \mathbf{m}_{i,t}^\top, \mathbf{h}_{j,t}^\top, \mathbf{m}_{j,t}^\top, d_{ij,t}]^\top$. Whenever it is not necessary to indicate the person indices, in the rest of this paper, we simplify the notation $\{\mathbf{y}_{ij,t}\}_{t=1}^T$ to $\{\mathbf{y}_t\}$, and we refer to it as an *interaction trajectory*. Finally, when the audio features a_t are available, they are concatenated with the motion and proximity features to compose an *audio–video interaction trajectory* $\{[\mathbf{y}_{ij,t}^\top, a_t^\top]^\top\}_{t=1}^T$.

IV. Kernel Models for Interaction Trajectories

An interaction trajectory is a temporal sequence, assuming values in a non-Euclidean space with a specific structure. For instance, the HOOF and MH features are histograms. Therefore, trajectories could be mapped onto a new space, where they could be modeled more effectively by existing linear models. Here, we describe two models that include a mapping of the input trajectory by using kernels. Although the model of reference is described in Section IV-B, the model in Section IV-A serves as baseline, especially for introducing and comparing the online detection approach.

A. Kernel Regression Models

Given the interaction trajectory $\{\mathbf{y}_t\}$, assuming values in a space $\mathcal{S}$, which may not be Euclidean, let us consider the Mercer kernel $\kappa(\mathbf{y}_t, \mathbf{y}_t') = \langle \phi(\mathbf{y}_t), \phi(\mathbf{y}_t') \rangle$, where $\phi(\cdot)$ maps $\mathcal{S}$ onto $\mathcal{H}$, an RKHS [16]. We assume that $\{\mathbf{y}_t\}$ is mapped to a sequence $\{\phi(\mathbf{y}_t)\}$, which can be expressed with the *KR* model

$$\phi(\mathbf{y}_t) = C\mathbf{x}_t + w_t. \tag{1}$$

The quantity C may not be a matrix but a linear operator $C : \mathbb{R}^n \to \mathcal{H}$, acting on the regressor $\mathbf{x}_t \in \mathbb{R}^n$ at time t, to account that $\mathcal{H}$ could be an infinite dimensional space. Indeed, C can be represented as $C \doteq [c_1, \ldots, c_n]$ and $C\mathbf{x} \doteq \sum_{i=1}^n c_i x_i$, where $\mathbf{x} \doteq [x_1, \ldots, x_n]^\top$. The observation noise w_t is modeled as a zero-mean Gaussian process.

1) Parameter Estimation: The obvious choice to model the variability of $\{\mathbf{y}_t\}$ in feature space is to apply kernel PCA (KPCA) [16]. To this end, it is convenient to introduce the notation $\Phi \doteq [\phi(\mathbf{y}_1), \ldots, \phi(\mathbf{y}_T)]$, and the kernel matrix $K \doteq \Phi^\top \Phi$, where $[K]_{st} = \kappa(\mathbf{y}_s, \mathbf{y}_t)$. KPCA evaluates T kernel principal components out of a linear combination of the elements of ΦJ, where $J \doteq (I - 1/T\mathbf{e}\mathbf{e}^\top)$ is the so-called *centering projection matrix*, and $\mathbf{e} = [1, \ldots, 1]^\top \in \mathbb{R}^T$. The linear combination coefficients are computed from the eigendecomposition of K, after assuring that data in feature space have zero mean. This is done by computing $JKJ \doteq \alpha\Lambda\alpha^\top$, where $\Lambda \doteq \mathrm{diag}(\lambda_1, \ldots, \lambda_T)$ and α are the eigenvalue and eigenvector matrices. The set of orthonormal kernel principal components can be expressed as $\Phi J\alpha\Lambda^{-(1/2)}$. Assuming that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_T$, in order to model the highest amount of data variability in feature space with only $n < T$ components, it is well known that the first n have to be picked. If $\beta \doteq \alpha\Lambda_n^{-(1/2)}$ indicates the first n kernel principal component coefficients obtained by removing the columns of Λ after the first n, we set the observation operator of model (1) to

$$\hat{C} \doteq \Phi J\beta. \tag{2}$$

For the noise model, we notice that $\sigma^2 \doteq E[\langle w_t, w_t\rangle]$, and its sample estimation is available from the KPCA, which is given by $\hat{\sigma}^2 = 1/T\sum_{t=1}^{T} \|w_t\|^2 = 1/T\sum_{i=n+1}^{T} \lambda_i$.

B. Kernel State-Space Models

In the KR model, the samples of the temporal sequence $\{\mathbf{y}_t\}$ are independent identically distributed (i.i.d). When those are correlated, the regressor, or *state* temporal sequence $\{\mathbf{x}_t\}$, can be modeled by an autoregression, thus obtaining a *KSS* model, given by

$$\begin{cases} \mathbf{x}_{t+1} = A\mathbf{x}_t + \mathbf{v}_t \\ \phi(\mathbf{y}_t) = C\mathbf{x}_t + w_t \end{cases} \tag{3}$$

where $A \in \mathbb{R}^{n\times n}$ describes the dynamics of the state, and $\mathbf{v}_t$ is the system noise, which is zero-mean i.i.d. Gaussian distributed with covariance Q, and independent of w_t.

1) Parameter Estimation: It is possible to extend the procedure developed in [41] for estimating the parameters of KSS models (3). This is done by substituting the PCA applied to the temporal sequence with KPCA [16], as it is shown in [42]. In particular, [42] shows that the linear operator C can be estimated according to (2), whereas the estimation of A and Q is carried out as in [41]. In addition, our online implementation uses the online kernel PCA algorithm of [43], and the matrix A is recursively updated online as explained in [44].

V. Human Interaction Detection

Inspired by the successful concept of parity space, developed within the context of fault detection applications based on linear models [18], here, we extend the approach to be used with KR, as well as KSS models. The extension to KR models (Section V-A) provides a baseline approach for interaction detection, and helps with understanding the approach developed for KSS models in Section V-B. Subsequently, Section V-C applies the framework of Sections V-A and V-B to the problem of detecting human interactions online.

A. Detection With KR Models

We begin by introducing the concept of *kernel parity Hilbert space (KPHS)*, which is the subspace of $\mathcal{H}$ defined as $\mathcal{P} \doteq \{v \in \mathcal{H} | \langle c_i, v\rangle = 0, i = 1, \ldots, n\}$. We also indicate with $P_\mathcal{P}$ the operator that projects a vector $v \in \mathcal{H}$ onto $\mathcal{P}$, given by $P_\mathcal{P} v$, whereas $\xi_t \doteq P_\mathcal{P}\phi(\mathbf{y}_t)$ is called *kernel parity vector*.

If a temporal sequence $\{\mathbf{y}_t\}$ is made of i.i.d. samples and is modeled by (1), then for an input sample $\mathbf{y}_t$, the kernel parity vector ξ_t indicates in what direction and by how much the sample does not belong to the span of the $\{c_i\}$. In particular, since it is also true that $\xi_t = P_\mathcal{P} w_t$, ξ_t tells, in feature space $\mathcal{H}$, by how much the measurement noise has spilled into $\mathcal{P}$ in order for model (1) to hold. This fact is suggesting that if we knew the noise model, monitoring ξ_t would reveal whether the current sample $\mathbf{y}_t$ implies a noise model different than the one given, which means that model (1) should no longer hold.

Let us now consider the *residual error* $e_t \doteq \phi(\mathbf{y}_t) - C\hat{\mathbf{x}}_t$, where $\hat{\mathbf{x}}_t$ is the maximum likelihood estimation of the regressor, given the observation $\mathbf{y}_t$, and the model defined by κ and C. Under the hypothesis of the noise w_t being i.i.d. realizations from an uncorrelated stationary Gaussian process, which means that its autocorrelation function is given by $\sigma^2\delta$, where δ is a Dirac distribution defined over a suitable domain, the maximum likelihood estimation $\hat{\mathbf{x}}_t$ coincides with the least squares estimation

$$\hat{\mathbf{x}}_t = \arg\min_{\mathbf{x}} \|\phi(\mathbf{y}_t) - C\mathbf{x}\|^2. \tag{4}$$

Under the hypothesis expressed earlier, we now show how it is possible to connect the residual error to the kernel parity vector, and to define a rule to establish whether the sample $\mathbf{y}_t$ is in accordance with model (1). We start by plugging (2) into (4); we remove the mean of the model $(1/T)\Phi\mathbf{e}$, and after expanding the square, taking the derivative with respect to $\mathbf{x}$, and setting it to zero, one can obtain

$$\hat{\mathbf{x}}_t = \beta^\top J\breve{\kappa}(\mathbf{y}_t) \tag{5}$$

where $\breve{\kappa}(\mathbf{y}_t) \doteq (\tilde{\kappa}(\mathbf{y}_t) - 1/TK\mathbf{e})$ and $\tilde{\kappa}(\cdot) \doteq [\kappa(\mathbf{y}_1, \cdot), \ldots, \kappa(\mathbf{y}_T, \cdot)]^\top$. Moreover, by combining (4) and (5), we can see that $\min_{\mathbf{x}} \|\phi(\mathbf{y}_t) - 1/T\Phi\mathbf{e} - C\mathbf{x}\|^2 = \|P_{C^\perp}(\phi(\mathbf{y}_t) - 1/T\Phi\mathbf{e})\|^2$, where $P_{C^\perp}$ is the projection operator defined by

$$P_{C^\perp} = I - \Phi J\beta\beta^\top J\Phi^\top \tag{6}$$

where I here indicates the identity operator. Thus, we can say that $e_t = P_{C^\perp}(\phi(\mathbf{y}_t) - 1/T\Phi\mathbf{e})$, and by construction, $P_{C^\perp}$ represents an orthonormal projection onto the orthogonal complement of the span of the $\{c_i\}$, and therefore, it is equivalent to $P_\mathcal{P}$. In particular, we have $\|e_t\|^2 = \|\xi_t\|^2$. Note that everything so far has been derived under the hypothesis of w_t being an uncorrelated stationary Gaussian process, which is an idealized scenario. If, for instance, the noise is correlated, the autocorrelation function is not a Dirac delta, and the residual error should be estimated with generalized least squares. This means that this derivation and implementation become more complex, because the autocorrelation function needs to be estimated.

Finally, the fact that $\|e_t\|^2 = \|\xi_t\|^2$ suggests that the criterion for establishing whether or not a new sample $\mathbf{y}_t$ is in accordance with model (1) is to simply check if the *normalized residual error* $\|e_t\|^2/\sigma^2$ is lower or greater than a threshold ν, appropriately chosen. Also, note that through (6), the analytical expression of $\|e_t\|^2$, function only of the kernel κ, is readily available, and is given by

$$\begin{aligned}\|e_t\|^2 = \kappa(\mathbf{y}_t, \mathbf{y}_t) - \frac{1}{T}\mathbf{e}^\top(\tilde{\kappa}(\mathbf{y}_t) + \breve{\kappa}(\mathbf{y}_t)) \\ -\breve{\kappa}(\mathbf{y}_t)^\top J\beta\beta^\top J\breve{\kappa}(\mathbf{y}_t). \end{aligned} \tag{7}$$

B. Detection With KSS Models

In order to extend the idea of KPHS to KSS models, we first extend the definition of observability matrix, a well-known concept in the theory of linear dynamical systems (LDSs). In particular, we consider the linear operator $O_\tau : \mathbb{R}^n \to \mathcal{H}^\tau$, mapping $\mathbf{x}$ to $O_\tau \mathbf{x}$, where $O_\tau \doteq [C^\top, A^\top C^\top, \cdots A^{\tau-1^\top} C^\top]^\top$. Based on this, we extend the definition of KPHS into *KPHS of order τ (KPHS-τ)*, which is the subspace of $\mathcal{H}^\tau$ defined as $\mathcal{P}_\tau \doteq \{\upsilon \in \mathcal{H}^\tau | \upsilon^\top O_\tau = 0\}$.

Following the standard representation used in system identification [17], we define the vectors $\Phi^t_{t-\tau+1} \doteq [\phi(\mathbf{y}_{t-\tau+1})^\top, \ldots, \phi(\mathbf{y}_t)^\top]^\top$, $W^t_{t-\tau+1} \doteq [\omega^\top_{t-\tau+1}, \ldots, \omega^\top_t]^\top$, $\mathbf{V}^t_{t-\tau+1} \doteq [\mathbf{v}^\top_{t-\tau+1}, \ldots, \mathbf{v}^\top_t]^\top$, and the matrix

$$\tilde{O}_\tau \doteq \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ & \ddots & \cdots & \vdots \\ & & 0 & \vdots \\ O_{\tau-1} & \cdots & O_1 & 0 \end{bmatrix} \tag{8}$$

with which model (3) can be rewritten as

$$\Phi^t_{t-\tau+1} = O_\tau \mathbf{x}_{t-\tau+1} + \tilde{W}^t_{t-\tau+1} \tag{9}$$

where $\tilde{W}^t_{t-\tau+1} \doteq \tilde{O}_\tau \mathbf{V}^t_{t-\tau+1} + W^t_{t-\tau+1}$ is a zero-mean Gaussian process noise with autocorrelation matrix function $\tilde{O}_\tau I_\tau \otimes Q\tilde{O}_\tau^\top + I_\tau \otimes \sigma^2\delta$, and $\otimes$ indicates the Kronecker product.

As in Section V-A, $P_{\mathcal{P}_\tau}$ is the operator that projects a vector $\upsilon \in \mathcal{H}^\tau$ onto $\mathcal{P}_\tau$, given by $P_{\mathcal{P}_\tau}\upsilon$, whereas $\Xi^t_{t-\tau+1} \doteq P_{\mathcal{P}_\tau}\Phi^t_{t-\tau+1}$ is the kernel parity vector. According to the definition of KPHS-τ, it follows that $\Xi^t_{t-\tau+1} = P_{\mathcal{P}_\tau}\tilde{W}^t_{t-\tau+1}$, which shows that $\Xi^t_{t-\tau+1}$ is independent of the state $\mathbf{x}_{t-\tau+1}$, and it can be interpreted with respect to $\mathbf{y}_{t-\tau+1}, \ldots, \mathbf{y}_t$ exactly in the same way as ξ_t is interpreted with respect to $\mathbf{y}_t$.

We now shift the attention to the reconstruction error $E^t_{t-\tau+1} \doteq \Phi^t_{t-\tau+1} - O_\tau \hat{\mathbf{x}}_{t-\tau+1}$, where $\hat{\mathbf{x}}_{t-\tau+1}$ is the maximum likelihood estimation of $\mathbf{x}_{t-\tau+1}$, and we make the further simplifying assumption that the autocorrelation matrix function of $\tilde{W}^t_{t-\tau+1}$ is given by $I_\tau \otimes \sigma^2\delta$. This allows to compute $\hat{\mathbf{x}}_{t-\tau+1}$ with a simple least squares estimation. In particular, it is easy to show that

$$\hat{\mathbf{x}}_{t-\tau+1} = \left(\sum_{i=0}^{\tau-1} A^{i^\top} A^i\right)^{-1} \sum_{i=0}^{\tau-1} A^{\tau-1-i^\top}\beta^\top J\breve{\kappa}(\mathbf{y}_{t-i}). \tag{10}$$

From (10), one can compute the projection operator $P_{O_\tau^\perp}$, such that $E^t_{t-\tau+1} = P_{O_\tau^\perp}(\Phi^t_{t-\tau+1} - \mathbf{e}_\tau \otimes 1/T\Phi\mathbf{e})$, where $\mathbf{e}_\tau$ is a column vector with τ ones, which is given by

$$P_{O_\tau^\perp} \doteq I - O_\tau \left(\sum_{i=0}^{\tau-1} A^{i^\top} A^i\right)^{-1} O_\tau^\top. \tag{11}$$

By construction, $P_{O_\tau^\perp}$ represents an orthonormal projection onto the orthogonal complement of the span of the columns of O_τ, and therefore, it is equivalent to $P_{\mathcal{P}_\tau}$, and, in particular, $\|E^t_{t-\tau+1}\|^2 = \|\Xi^t_{t-\tau+1}\|^2$. As in Section V-A, this is true under the hypothesis of $\tilde{W}^t_{t-\tau+1}$ being an uncorrelated stationary Gaussian process, which is an idealized scenario.

Similarly to the KR model, the criterion for establishing whether or not the trajectory $\mathbf{y}_{t-\tau+1}, \ldots, \mathbf{y}_t$ is in accordance with model (3) is to simply check if the *normalized residual error* $\|E^t_{t-\tau+1}\|^2/\tau\sigma^2$ is lower or greater than a threshold ν, appropriately chosen. By using (11), the analytical expression of $\|E^t_{t-\tau+1}\|^2$, function only of the kernel κ, can be computed in closed form, and is given by

$$\begin{aligned}\|E^t_{t-\tau+1}\|^2 = \sum_{i=0}^{\tau-1}\kappa(\mathbf{y}_{t-i}, \mathbf{y}_{t-i}) - \frac{1}{T}\mathbf{e}^\top(\tilde{\kappa}(\mathbf{y}_{t-i}) + \breve{\kappa}(\mathbf{y}_{t-i})) \\ -\sum_{i=0}^{\tau-1}\breve{\kappa}(\mathbf{y}_{t-i})^\top J\beta A^{\tau-1-i}\left(\sum_{i=0}^{\tau-1} A^{i^\top} A^i\right)^{-1} \\ \cdot\sum_{i=0}^{\tau-1} A^{\tau-1-i^\top}\beta^\top J\breve{\kappa}(\mathbf{y}_{t-i}). \end{aligned} \tag{12}$$

As expected, we notice that when $\tau = 1$, (10)–(12) collapse to (5)–(7), respectively.

C. Online Human Interaction Detection

In order to detect human interactions online, we take the approach of deploying a temporal incremental window [10], [15], and sequentially detect segmentation cuts. Let us assume that in monitoring a temporal sequence $\{\mathbf{y}_t\}$, the last segmentation cut was observed at time $s < t$, where t is the current time. We want to test whether at time $t - \tau$, a new cut should be detected. To this end, either a KR model (1) or a KSS model (3) is estimated from the data in the *training time window* $[s + 1, \ldots, t - \tau]$ of length $T_t \doteq t - \tau - s$, i.e., $\mathbf{y}_{s+1}, \ldots, \mathbf{y}_{t-\tau}$ (see Fig. 2). A cut should be detected if the data observed in the subsequent *test time window* $[t - \tau + 1, \ldots, t]$, i.e., $\mathbf{y}_{t-\tau+1}, \ldots, \mathbf{y}_t$, and the model previously estimated does not fit "well enough."

The geometric framework introduced in Sections V-A and V-B tells us to project the test data onto the KPHS (KPHS for KR models or KPHS-τ for KSS models), and compare this projection with the noise model to decide whether data and model can fit. More formally, for the KR model one should compute the statistic

$$\varepsilon^{KR}_{t-\tau} \doteq \frac{1}{\tau\sigma^2}\sum_{i=0}^{\tau}\|e_{t-i}\|^2 \tag{13}$$

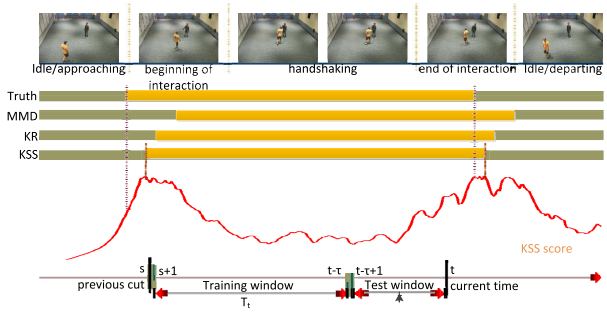


Fig. 2. From top to bottom: phases of a handshaking interaction; GT and estimated segmentation labels for the MMD, KR, and KSS models; KSS score (14) computed with a fixed length sliding window of length $\tau = 15$ for both training and testing; and notation for the incremental temporal window.

whereas for the KSS model, one should compute

$$\varepsilon_{t-\tau}^{\mathrm{KSS}} \doteq \frac{1}{\tau \sigma^2} \| E_{t-\tau+1}^{t} \|^2. \tag{14}$$

Finally, $\varepsilon_{t-\tau}^{KR}$ and $\varepsilon_{t-\tau}^{\mathrm{KSS}}$ can be used to test the hypotheses "yes cut," i.e., $\mathbf{H}_1$, versus "no cut," i.e., $\mathbf{H}_0$. In particular

$$\varepsilon_{t-\tau} \leq \nu \Rightarrow \mathbf{H}_0 \text{ is true}, \quad \varepsilon_{t-\tau} > \nu \Rightarrow \mathbf{H}_1 \text{ is true}. \tag{15}$$

If $\mathbf{H}_0$ is true, test (15) is repeated at time $t + \Delta t$. If $\mathbf{H}_1$ is true, the next test is performed at time $t + \tau$, with a training time window that restarts with length $T_{t+\tau} = \tau$.

VI. Human Interaction Recognition

Given a temporal sequence segment $\mathbf{y}_s, \ldots, \mathbf{y}_t$, obtained with the online segmentation of Section V-C, we want to identify the human interaction it might represent. Since interactions are characterized by a feature trajectory that is a temporally correlated sequence, we use a KSS model. Therefore, recognizing interactions entails comparing KSS models. If the trajectory space $\mathcal{S}$ was Euclidean, and the kernel used was linear, then the KSS degenerates to an LDS [17]. Methods for comparing LDSs include geometric distances [45], algebraic kernels [46], and information theoretic metrics [23]. When $\mathcal{S}$ is a non-Euclidean space, it is possible to compare KSS models through the use of Binet–Cauchy kernels [46]. In particular, [23] describes their use for action recognition when the input features are a temporal sequence of histograms, and [47] uses them for modeling and recognizing binary temporal sequences.

We note that a trajectory $\{\mathbf{y}_t\}$ does not assume values in an Euclidean space. Indeed, $\mathcal{S}$ is a Riemannian manifold with a nontrivial structure, which is $\mathbb{H}_b \times \mathbb{H}_c \times \mathbb{H}_b \times \mathbb{H}_c \times \mathbb{R}_+$. 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 $h_{t;k} \geq 0$, $\forall k \in \{1, \ldots, b\}$; and similarly, $\mathbb{H}_c$ is the space of normalized histograms with c bins. Therefore, we represent an interaction trajectory with a KSS model, where the kernel κ has to account for the geometric structure of $\mathcal{S}$. In addition, if the audio trajectory $\{a_t\}$ is also available, the kernel κ should be augmented accordingly.

Given the kernel κ, KSS models can be compared with a Binet–Cauchy kernel [46]. In particular, the Binet–Cauchy trace kernel is the expected value of an infinite series of weighted inner products between the outputs after embedding them into the feature space using the map $\phi(\cdot)$. More precisely

$$\begin{aligned} K_{\mathrm{KSS}}\left(\{\mathbf{y}_t\}_{t=1}^{\infty}, \{\mathbf{y}'_t\}_{t=1}^{\infty}\right) &\doteq E\left[\sum_{t=1}^{\infty} \lambda^t \phi(\mathbf{y}_t)^\top \phi(\mathbf{y}'_t)\right] \\ &= E\left[\sum_{t=1}^{\infty} \lambda^t \kappa(\mathbf{y}_t, \mathbf{y}'_t)\right] \end{aligned} \tag{16}$$

where $0 < \lambda < 1$, and the expectation of the infinite sum of the inner products is taken with respect to the joint probability distribution of $\mathbf{v}_t$ and ω_t. The kernel (16) can be computed in closed form, and it requires the computation of the infinite sum $P = \sum_{t=1}^{\infty} \lambda^t (A^t)^\top F A'^t$, where $F = \alpha J S J \alpha'$, J is a centering matrix of suitable dimensions, and α and α' are the KPCA weight vectors of $\{\mathbf{y}_t\}$ and $\{\mathbf{y}'_t\}$, respectively. S instead is such that $[S]_{st} = \kappa(\mathbf{y}_s, \mathbf{y}'_t)$, where $s \in \{1, \ldots, T\}$ and $t \in \{1, \ldots, T'\}$. 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 (16) 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 [23] 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 of the initial state and the noise processes. Therefore, the original kernel (16) is simplified to $K_{\mathrm{KSS}}^{\sigma}$, which is a kernel only on the dynamics of the KSS model, and is given by the maximum singular value of P, that is

$$K_{\mathrm{KSS}}^{\sigma} = \max \sigma(P). \tag{17}$$

For more details about the derivation of kernel (17), the reader is referred to [23]. In the sequel, we use the notation K_{KSS} to indicate $K_{\mathrm{KSS}}^{\sigma}$ to reduce the notation clutter. Besides (17), in some experiments, we have also used a radial basis function (RBF) kernel with distance derived from the kernel K_{KSS}, given by

$$K_{\mathrm{GKSS}}(\boldsymbol{Y}, \boldsymbol{Y}') = e^{-\eta(K_{\mathrm{KSS}}(\boldsymbol{Y},\boldsymbol{Y}) + K_{\mathrm{KSS}}(\boldsymbol{Y}',\boldsymbol{Y}') - 2K_{\mathrm{KSS}}(\boldsymbol{Y},\boldsymbol{Y}'))} \tag{18}$$

where $\boldsymbol{Y} = [\mathbf{y}_s, \ldots, \mathbf{y}_t]$. K_{KSS} and K_{GKSS} are used to train a multiclass classifier of the libSVM [48].

VII. Kernels for Interaction Recognition

In this section, we design the kernel κ, which is used by the recognition framework in Section VI. The guiding criterions for designing κ are the fact that the effectiveness of representing an interaction trajectory with a KSS model (3) is heavily affected by how ϕ maps the input space $\mathcal{S}$ onto the RKHS $\mathcal{H}$, and it should consider the geometry of $\mathcal{S}$ as much as possible. Moreover, whenever audio is present, κ should be augmented accordingly. The second important guiding criterion arises from the following observation. We notice that a recognition schema entails learning a decision function $f : \mathcal{S}^T \to \mathbb{R}$, which predicts 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

TABLE I
SUMMARY OF THE KERNEL HIERARCHY SYMBOLS

Symbol	Definition	Symbol	Definition
K_{KSS} and K_{GKSS}	Kernels for trajectories.	κ_H	Pairwise kernel on the histogram space.
$\kappa(\mathbf{Y}_t, \mathbf{Y}'_t)$	Kernel for trajectory samples with N views at time t.	κ_H^T and κ_H^D	Tensor and direct sum versions of κ_H.
$\kappa(\mathbf{y}_t, \mathbf{y}'_t)$	Kernel for trajectory samples at time t.	κ_H^{TL} and κ_H^{DS}	Balanced versions of κ_H^T and κ_H^D, respectively.
κ_d	Gaussian RBF kernel for proximity features.	k_h and k_m	Kernels for histogram features.
κ_a	Gaussian RBF kernel for audio features.	k_S	Geodesic kernel for histogram features.

person ordering is imposed *a priori*, the decision function is expected to be symmetric with respect to i and j, that is

$$f(\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j, d_{ij}) = f(\mathbf{h}_j, \mathbf{m}_j, \mathbf{h}_i, \mathbf{m}_i, d_{ji}). \quad (19)$$

In Section VII-A, we propose a family of so-called *pairwise kernels*, summarized in Table I, that account for the geometry of $\mathcal{S}$, as well as the symmetry of the decision function.

A. Pairwise Kernels for Interaction Trajectories

Since the input space $\mathcal{S} \doteq \mathbb{H}_b \times \mathbb{H}_c \times \mathbb{H}_b \times \mathbb{H}_c \times \mathbb{R}_+$ is non-Euclidean, defining κ to be a linear kernel would clearly be suboptimal. A typical approach for improving the map to an RKHS is to use a generic, top-performing nonlinear kernel, such as the Gaussian RBF kernel with Euclidean distance. However, in this way, we do not take advantage of the 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 best of our knowledge, defining a distance on $\mathcal{S}$ is still an open problem, although for $\mathbb{H}_b$ (or $\mathbb{H}_c$) alone a theoretical solution exists, which is the *Fisher–Rao* metric [49]. Therefore, whenever it cannot be done otherwise, we advocate the use of kernel construction techniques [16], which consider the fact that $\mathcal{S}$ is given by the Cartesian product of subspaces. This way allows to concentrate on each subspace separately, and exploits 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 $\mathbb{H}_b \times \mathbb{H}_c \times \mathbb{H}_b \times \mathbb{H}_c$, and $\mathbb{R}_+$. Therefore, we can design a kernel for histograms κ_H on the first subspace, and a kernel κ_d for people distances on the second subspace. κ_H and κ_d can then be combined by computing their *tensor product (TP) kernel* [16], leading to

$$\begin{aligned}\kappa &\doteq (\kappa_H \otimes \kappa_d)\big(\mathbf{y}_{ij}, \mathbf{y}'_{ij}\big)\\ &= \kappa_H\big((\mathbf{h}_i, \mathbf{m}_i, \mathbf{h}_j, \mathbf{m}_j), (\mathbf{h}'_i, \mathbf{m}'_i, \mathbf{h}'_j, \mathbf{m}'_j)\big)\kappa_d\big(d_{ij}, d'_{ij}\big) \quad (20)\end{aligned}$$

where we have dropped the time subscript t to lighten the notation. Intuitively, a kernel defines similarity in an input space. Kernel (20) 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 pointed out in Section III-B.

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

$$\kappa_d(d_{ij}, d'_{ij}) \doteq \exp\big(-\gamma\,\big|d_{ij} - d'_{ij}\big|^2\big). \quad (21)$$

For kernel κ_H, we note that it is a so-called *pairwise kernel* [50], because it is such that $\kappa_H : (\mathcal{X}_H \times \mathcal{X}_H) \times (\mathcal{X}_H \times \mathcal{X}_H) \to \mathbb{R}$, where $\mathcal{X}_H \doteq \mathbb{H}_b \times \mathbb{H}_c$, 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 semi-definite implies κ_H to have the following *symmetry* property:

$$\kappa_H((a, b), (a', b')) = \kappa_H((a', b'), (a, b)) \quad (22)$$

for all $a, b, a', b' \in \mathcal{X}_H$. By using kernel construction techniques based on direct sum (DS) and TP 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 κ_H:

$$\kappa_H^D = (k_H \oplus k_H)(a, b, a', b') = k_H(a, a') + k_H(b, b') \quad (23)$$
$$\kappa_H^T = (k_H \otimes k_H)(a, b, a', b') = k_H(a, a')k_H(b, b') \quad (24)$$

which obviously satisfy the symmetry property (22).

We now verify whether by using the kernels defined in (23) and (24), it is possible to construct decision functions for interaction trajectories, which are supposed to satisfy the symmetry property (19). We plan to learn decision functions f with an SVM that exploits the general kernel (16). Therefore, they will assume the form

$$\begin{aligned}&f(\{a_{i,t}, a_{j,t}, d_{ij,t}\})\\ &\doteq \sum_{u,v} \alpha_{uv}\ell_{uv} K_{\text{KSS}}\big(\{a_{i,t}, a_{j,t}, d_{ij,t}\}, \big\{a'_{u,t}, a'_{v,t}, d'_{uv,t}\big\}\big) + b_f\end{aligned} \quad (25)$$

where α_{uv}, ℓ_{uv}, and b_f are the usual SVM parameters [16], 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, (25) tells us that the symmetry property (19) imposes that

$$\begin{aligned}&K_{\text{KSS}}\big(\{a_{i,t}, a_{j,t}, d_{ij,t}\}, \big\{a'_{u,t}, a'_{v,t}, d'_{uv,t}\big\}\big)\\ &\quad = K_{\text{KSS}}\big(\{a_{j,t}, a_{i,t}, d_{ji,t}\}, \big\{a'_{u,t}, a'_{v,t}, d'_{uv,t}\big\}\big)\end{aligned} \quad (26)$$

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, (26) induces a symmetry property on the kernel (20) through (16), which is given by

$$\begin{aligned}&\kappa\big((a_{i,t}, a_{j,t}, d_{ij,t}), \big(a'_{u,t}, a'_{v,t}, d'_{uv,t}\big)\big)\\ &\quad = \kappa\big((a_{j,t}, a_{i,t}, d_{ji,t}), \big(a'_{u,t}, a'_{v,t}, d'_{uv,t}\big)\big)\end{aligned} \quad (27)$$

and finally, since $d_{ij,t} = d_{ji,t}$ and $d_{uv,t} = d_{vu,t}$, (27) imposes on κ_H the relationship

$$\kappa_H((a_{i,t}, a_{j,t}), (a'_{u,t}, a'_{v,t})) = \kappa_H\big((a_{j,t}, a_{i,t}), \big(a'_{u,t}, a'_{v,t}\big)\big) \quad (28)$$

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 (28) is different than the symmetry relationship (22), and kernels that satisfy (28) are called *balanced* [50].

Unfortunately, the pairwise kernels κ_H^D and κ_H^T, defined in (23) and (24), are symmetric but not balanced. Therefore, we propose to use two kernels that have been proved to have good theoretical properties [50], in that they guarantee minimal loss of information, and can be thought of as the balanced versions of κ_H^D and κ_H^T. They are defined as

$$\begin{aligned}\kappa_H^{\mathrm{DS}}((a,b),(a',b')) &= \kappa_H^{SD}((a,b),(a',b')) \\ &\quad + \kappa_H^{ML}((a,b),(a',b')) \qquad (29)\\ \kappa_H^{\mathrm{TL}}((a,b),(a',b')) &= \frac{1}{2}(k_H(a,a')k_H(b,b') \\ &\quad + k_H(a,b')k_H(b,a')) \qquad (30)\end{aligned}$$

where

$$\begin{aligned}\kappa_H^{SD}((a,b),(a',b')) &= \frac{1}{2}(k_H(a,a') + k_H(a,b') \\ &\quad + k_H(b,a') + k_H(b,b')) \qquad (31)\\ \kappa_H^{ML}((a,b),(a',b')) &= \frac{1}{4}(k_H(a,a') - k_H(a,b') \\ &\quad - k_H(b,a') + k_H(b,b'))^2 \qquad (32)\end{aligned}$$

κ_H^{TL} is called *tensor learning pairwise kernel* [51], whereas κ_H^{DS} is called *DS pairwise kernel* [50].

Finally, we are left with the task of designing k_H, which is defined on the space $(\mathbb{H}_b \times \mathbb{H}_c) \times (\mathbb{H}_b \times \mathbb{H}_c)$. 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 toward establishing similarity, we apply the TP rule to further decompose k_H into $k_h : \mathbb{H}_b \times \mathbb{H}_b \to \mathbb{R}$ and $k_m : \mathbb{H}_c \times \mathbb{H}_c \to \mathbb{R}$, producing

$$k_H\big((\mathbf{h}_{i,t}, \mathbf{m}_{i,t}), (\mathbf{h}'_{i,t}, \mathbf{m}'_{i,t})\big) = k_h\big(\mathbf{h}_{i,t}, \mathbf{h}'_{i,t}\big) k_m\big(\mathbf{m}_{i,t}, \mathbf{m}'_{i,t}\big). \qquad (33)$$

Both k_h and k_m are kernels for comparing histograms. As shown in [23], several options are available for kernels in this domain, and one with an excellent compromise between performance and speed is given by the Mercer kernel

$$k_S(\mathbf{h}_1, \mathbf{h}_2) = \sum_{k=1}^{b} \sqrt{h_{1,k} h_{2,k}} \qquad (34)$$

which is derived by considering 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 c bins, respectively.

B. Kernels for Audio–Video Interaction Trajectories

When the interaction trajectory is augmented with audio, we modify κ in (20) by adding an audio kernel κ_a as another factor. We have used the linear kernel, as well as the Gaussian RBF kernel

$$\kappa_a(a_t, a'_t) \doteq \exp\big(-\gamma \,||a_t - a'_t||^2\big). \qquad (35)$$

VIII. Extension to Multiple Cameras

We now assume that N cameras are simultaneously tracking person i and person j and each of them provide interaction trajectories $\{\mathbf{y}_t^{(1)}\}, \cdots, \{\mathbf{y}_t^{(N)}\}$. We consider three different ways to perform the fusion of this information for detecting and recognizing interactions. Since all trajectories represent the same interaction as it appears from different viewpoints, we make the assumption that each trajectory should have been originated from the same state trajectory $\{\mathbf{x}_t\}$.

The first fusion approach consists in combining the trajectories by forming the new trajectory $\{\mathbf{Y}_t\} \doteq \{[\mathbf{y}_t^{(1)\top}, \ldots, \mathbf{y}_t^{(N)\top}]^\top\}$. At this point, we simply need to define the map ϕ, to directly apply the detection and recognition approaches explained in Sections V and VI. In particular, since a trajectory point at time t is defined in the space $\mathcal{S}^N$, in order to account for the geometry of this space, we define ϕ by choosing the kernel κ in (16) to be the TP kernel of the respective kernels (20) of each camera view, that is

$$\kappa(\mathbf{Y}_t, \mathbf{Y}'_t) = \prod_{\upsilon=1}^{N} \kappa\big(\mathbf{y}_t^{(i)}, \mathbf{y}_t^{(i)'}\big). \qquad (36)$$

We indicate this as the *TP* fusion approach.

The second approach consists in applying the map ϕ as defined in Section VII to each of the views. This approach is equivalent to using the detection and recognition methods with a kernel κ in (16) that is the sum of the respective kernels (20) of each camera view, that is

$$\kappa(\mathbf{Y}_t, \mathbf{Y}'_t) = \sum_{\upsilon=1}^{N} \kappa\big(\mathbf{y}_t^{(i)}, \mathbf{y}_t^{(i)'}\big). \qquad (37)$$

We indicate this as the DS fusion approach.

The third method is a modification of the previous. The DS method is based on a feature map induced by (37), and then uses KPCA to estimate the state trajectory $\{\mathbf{x}_t\}$. Another approach instead, which is common in multiview learning [52], is to estimate a state trajectory $\{\mathbf{x}_t\}$, such that the correlation between the interaction trajectories from all the views is the highest possible. The statistical tool that allows to perform this projection is the *kernel multiset canonical correlation analysis (KMCCA)* [19]. While KPCA produces a single projection matrix $\beta = \alpha\Lambda^{-(1/2)}$ (as explained in Section IV-A, where we do not consider the truncation to the first n components for the moment), that is, then applied to each data set in feature space (i.e., $\Phi^{(1)}, \ldots, \Phi^{(N)}$), KMCCA produces N projection matrices $\beta^{(1)}, \ldots, \beta^{(N)}$, one for each of the camera views. Therefore, in order to apply the proposed detection and recognition methods, it is sufficient to compute the "equivalent" β that should be used, and all the rest remains the same. In particular, since it has to be that

$$\begin{bmatrix} \Phi^{(1)} J \beta^{(1)} \\ \vdots \\ \Phi^{(N)} J \beta^{(N)} \end{bmatrix} = \begin{bmatrix} \Phi^{(1)} \\ \vdots \\ \Phi^{(N)} \end{bmatrix} J \tilde{\beta} \qquad (38)$$

it follows that

$$\tilde{\beta} = J^{-1} \left(\sum_{\upsilon=1}^{N} K^{(\upsilon)} \right)^{-1} \sum_{\upsilon=1}^{N} K^{(\upsilon)} J \beta^{(\upsilon)}. \qquad (39)$$

Finally, since we require the "equivalent" β to have orthogonal columns, we decompose $\tilde{\beta}$ with an SVD, such that $\tilde{\beta} = U_{\tilde{\beta}} S_{\tilde{\beta}} V_{\tilde{\beta}}$, and we set $\beta = U_{\tilde{\beta}} S_{\tilde{\beta},n}$, where $S_{\tilde{\beta},n}$ is obtained from $S_{\tilde{\beta}}$ after removing the columns after the first n. We refer to this as the *KMCCA* fusion approach. Note that the fusion task that we have considered is different form the cross-view human activity recognition task [53]. There the goal is to improve recognition performance from a single camera view that was "unseen" during training time. Here, instead, the goal is to improve performance from the simultaneous presence of multiple camera views at testing time.

IX. Experiments

We have tested our approaches on four state-of-the-art data sets and two newly collected data sets: the UT-Interaction (UT-I) data set [2], the TV Human Interaction (TVHI) data set [3], the BIT-Interaction (BIT-I) data set [6], the SBU-Kinect-Interaction (SBU) data set [54], the Human Activities Under Surveillance—Person Interaction (HAUS-PI) data set, and the Multiple Views HAUS-PI (MVHAUS-PI) data set. Since the BIT-I, the TVHI, and the SBU data sets include tightly segmented videos, which only contain the active part of the interactions, we use them to test only the recognition but not the detection algorithms. We briefly introduce the data sets and then describe the recognition and detection results obtained from the single view data sets, and we finish with the results on recognition and detection from the new multiple camera human interaction data set MVHAUS-PI.

UT-I: The UT-I data set [2] is a single camera view data set containing 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 data set is divided into Set 1 and Set 2, each consisting of ten 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. To have people tracking information, since the ground-truth (GT) annotation of the data set was not providing that, we annotated the data with the VATIC tool [55]. The left frame of Fig. 1 shows the bounding boxes obtained with this process. Also, the right frame 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.

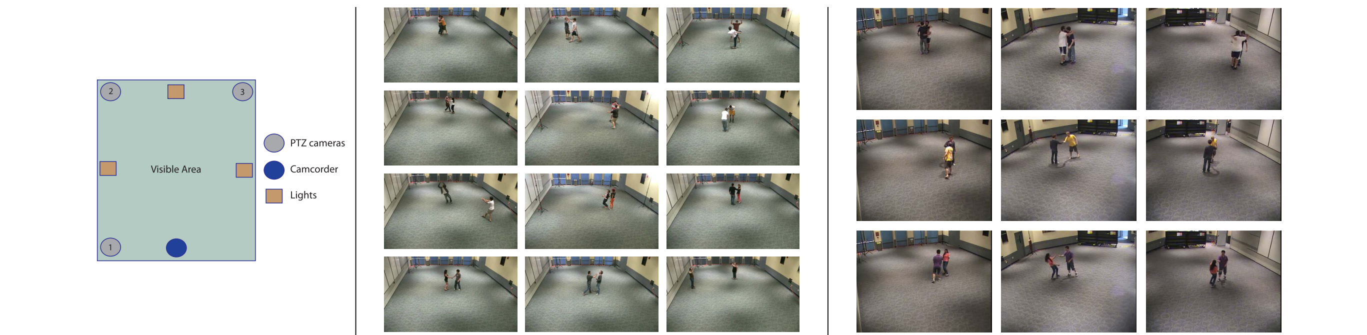
TVHI: The TVHI data set [3] is a single camera view data set with videos from five different classes: *Hand Shakes*, *High-Fives*, *Hugs*, *Kisses*, and *Negative* examples. The length of the videos ranges from 30 to 600 frames. There is a great degree of variation among the videos as they are compiled from different TV shows, which makes this data set very challenging. As people tracking information, we used the GT annotations available 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 with a width double the original annotations, in order to analyze the motion in a region surrounding each person. Note that, similar to [3], some of the original videos were not considered due to their very limited length, or due to sharp view point changes during the interaction.

BIT-I: The BIT-I data set [6] is a single camera view data set with videos from eight classes of human interactions: *Bow*, *Boxing*, *Handshake*, *High-Five*, *Hug*, *Kick*, *Pat*, and *Push*. Each class includes 50 videos. All people in the scene are annotated by bounding boxes using the VATIC tool, and the features are computed as explained for the UT-I data set. For each class, there are large variations of people's appearance, scales, illumination condition, background, and viewpoints. Moreover, some interactions are partially occluded.

SBU: The SBU data set [54] includes RGB and depth video sequences of eight human interaction classes: *Approaching*, *Departing*, *Kicking*, *Punching*, *Pushing*, *Hugging*, *ShakingHands*, and *Exchanging*. Videos are recorded using the Microsoft Kinect and the 3D joint locations are also available.

HAUS-PI and MVHAUS-PI: We collected the HAUS-PI data set and the MVHAUS-PI. The data set comprises videos of 16 person interaction classes: *Handshaking (HS), Hugging (HG), High-Fiving (HF), Kicking (KI), Punching (PC), Pushing (PS), Slapping (SL), Bowing (BO), Waving (WA), Starring (SR), Getting Up (GU), Contraband Exchange (CE), Shooting (SH), Stabbing (SB), Talking (TA), and Patting (PA).* The number of video samples per class is approximately 45. A sketch of the collection site is shown in Fig. 3(a), where a camcorder, with 1280×720 of progressive pixel resolution, and three surveillance cameras, with 640×480 of interlaced pixel resolution, were used to record the activities in the visible area. We calibrated all the cameras, meaning that the projection matrix for every video is available, which provides a metric mapping between the video image plane and the 3D ground plane of the visible area. We developed a semiautomated tool for annotating all the videos, which provides the tracking bounding boxes of the people in the scene. Those can be used in conjunction with the projection matrices to compute 3D people's trajectories, and they are also used for computing the HOOF and MH features as it is done for the UT-I data set. In particular, the camcorder videos were downsampled to a resolution of 640×360, and the surveillance camera videos were deinterlaced down to a resolution of 320×240. In addition, the annotation tool allows to indicate the time intervals when an interaction is occurring, and we have used it also to augment the UT-I data set with temporal GT annotations. The participants in the data collection were allowed to enter the scene from any direction, and the interactions are recorded with a very high viewpoint change variation.

The camcorder videos constitute the HAUS-PI data set, which is a single view data set. In particular, we have annotated the videos of 12 classes, and we have used them for testing the single view detection and recognition algorithms. Fig. 3(b) shows a frame from each of those classes. The set of videos from the three surveillance cameras are time synchronized, and constitute the MVHAUS-PI data set. For three different interactions, Fig. 3(c) shows the synchronized frames taken from the three cameras. To the best of our knowledge, this is the first video data set for person interactions captured from multiple view points by a surveillance camera network. In this paper, we have annotated all the 16 classes, and we have used the data set for testing the multiple camera views



(a) (b) (c)

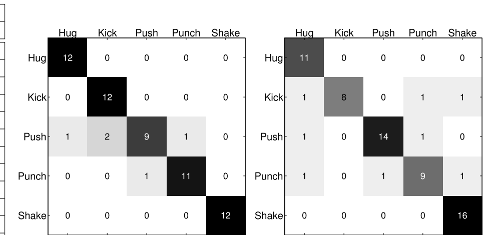
Fig. 3. HAUS-PI collection. (a) Floor plant of the collection site with locations of the cameras and lights used to record 16 different person interactions. The dimension of the visible area is approximately 30 ft × 30 ft. (b) One frame per each of the 12 annotated classes of the HAUS-PI data set. From top-left to bottom-right: *Hugging, High-Five, Kicking, Pushing, Slapping, Bowing, Shooting, Stabbing, Patting, Handshake, Punching*, and *Waving*. (c) Each row shows three time synchronized frames from the three surveillance cameras used to collect the MVHAUS-PI data set. From top to bottom: interaction are: *Hugging, Handshaking*, and *Punching*. From left to right: we see: View 1, View 2, and View 3.

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
$\kappa_H^{TL}(k_S)$	83.33	75.00	61.53	41.66	91.66	70.49
$\kappa_H^{DS}(k_S)$	83.33	75.00	38.46	33.33	91.66	63.93
$\kappa_H^{TL}(k_h k_m)$	83.33	83.33	84.61	8.33	100	70.49
$\kappa_H^{DS}(k_h k_m)$	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
$\kappa_H^{TL}(k_S)k_d$	91.66	83.33	76.92	91.66	100	88.52
$\kappa_H^{DS}(k_S)k_d$	100	83.33	69.23	50	91.66	78.68
$\kappa_H^{TL}(k_h k_m)k_d$	100	100	69.23	91.66	100	**91.80**
$\kappa_H^{DS}(k_h k_m)k_d$	100	83.33	69.23	66.66	91.66	81.96

(a)

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
$\kappa_H^{TL}(k_S)$	54.54	54.54	62.5	16.66	87.5	57.57
$\kappa_H^{DS}(k_S)$	54.54	54.54	25.00	8.33	93.75	48.48
$\kappa_H^{TL}(k_h k_m)$	72.72	63.63	50.00	50.00	62.50	59.09
$\kappa_H^{DS}(k_h k_m)$	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
$\kappa_H^{TL}(k_S)k_d$	81.81	72.72	50.00	16.16	75.00	59.09
$\kappa_H^{DS}(k_S)k_d$	90.90	27.27	50.00	33.33	93.75	60.60
$\kappa_H^{TL}(k_h k_m)k_d$	100	72.72	87.50	75.00	100	**87.87**
$\kappa_H^{DS}(k_h k_m)k_d$	100	36.36	87.50	41.66	100	75.75

(b)



(c) (d)

Fig. 4. Recognition on UT-I. Recognition accuracy on (a) Set 1 and (b) Set 2, respectively. For Set 1, MH features are computed with $\tau = 14$ and $\delta = 2$; HOOF features are computed with $b = 18$; KSS order is set to $n = 8$. For Set 2, MH features are computed with $\tau = 22$ and $\delta = 5$; HOOF features are computed with $b = 24$; KSS order is set to $n = 10$. Confusion matrices for the best performance on (c) Set 1 and (d) Set 2 with K_{KSS} kernel.

detection and recognition algorithms. The HAUS-PI and the MVHAUS-PI data sets are publicly available.

Model Selection: The order of models (1) and (3) was searched in the range $\{8, 10\}$, γ of the Gaussian RBF kernels in the range $\{10^0, 10^{-1}, 10^{-2}\}$, the number of bins of the HOOF and MH features in the range $\{5, \ldots, 22\}$, and τ in the range $\{10, 20, 40\}$. λ in (16) was set to 0.9.

A. Single Camera View Results

1) Kernel Comparison for Recognition: We tested the kernels proposed in Section VII-A to classify interactions using the UT-I, and the TVHI data sets. Different choices of κ_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 κ_d kernel (21). 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 $\mathbb{H}_{2b+2c}$, it is possible to test the following choices for κ_H: 1) k_S, which is the geodesic kernel (34); 2) κ_H^{TL} (30), where k_H is a geodesic kernel, indicated with $\kappa_H^{\text{TL}}(k_S)$; 3) κ_H^{DS} (29), where k_H is a geodesic kernel, indicated with $\kappa_H^{\text{DS}}(k_S)$; 4) κ_H^{TL} (30), where k_H is the TP kernel (33), indicated with $\kappa_H^{\text{TL}}(k_h k_m)$; and 5) κ_H^{DS} (29), where k_H is the TP kernel (33), indicated with $\kappa_H^{\text{DS}}(k_h k_m)$. Finally, to better show the advantage of using pairwise kernels, for kernel κ (20), we also tested a baseline Gaussian RBF kernel with Euclidean distance.

The kernels described earlier were used, in conjunction with kernel (17), to train the multiclass classifier of the libSVM [48] with leave-one-out cross-validation. The tables in Fig. 4(a) and (b) show the classification accuracy for the UT-I data set, whereas the table in Fig. 5(a) shows the classification results for the TVHI data set. From them, we can draw a number of considerations. First, as pointed out in Section VII-A, using the baseline RBF kernel leads to reduced performance, highlighting the need for using pairwise kernels for human interaction recognition. Second, using the tensor learning pairwise kernel $\kappa_H^{\text{TL}}(k_h k_m)$, rather than the DS pairwise kernel $\kappa_H^{\text{DS}}(k_h k_m)$, usually leads to higher performance. Third, we have verified the importance of designing kernels by considering the structure of the input feature space in the way that different kernels rank in terms of performance (e.g., replacing k_S with $k_h k_m$). Fourth, we have verified the importance in incorporating proximity information for discriminating between interactions.

For the TVHI data set, we have also performed a video retrieval experiment. In particular, we have converted the proposed kernels in pairwise distances, where the kernel K_{KSS} is normalized to 1 when $\{\mathbf{y}_t\} = \{\mathbf{y}'_t\}$, by computing $\tilde{K}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) \doteq K_{\text{KSS}}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\})/F$ where $F = (K_{\text{KSS}}(\{\mathbf{y}_t\}, \{\mathbf{y}_t\})K_{\text{KSS}}(\{\mathbf{y}'_t\}, \{\mathbf{y}'_t\}))^{1/2}$, and the distance

CLASSIFICATION ACCURACY

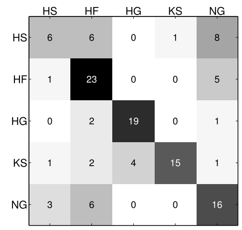
Kernel \ Class	HS	HF	HG	KS	NG	AVG
No Proximity						
k_S	33.33	51.72	36.36	30.43	52.00	40.77
$\kappa_H^{TL}(k_S)$	19.05	58.62	18.18	47.83	60.00	40.74
$\kappa_H^{DS}(k_S)$	0	62.07	31.82	30.43	88.00	42.46
$\kappa_H^{TL}(k_h k_m)$	38.10	44.83	31.82	30.43	44.00	37.84
$\kappa_H^{DS}(k_h k_m)$	9.52	41.38	31.82	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
$\kappa_H^{TL}(k_S)k_d$	19.05	79.31	81.82	65.22	64.00	61.88
$\kappa_H^{DS}(k_S)k_d$	23.81	51.72	90.91	78.26	64.00	61.74
$\kappa_H^{TL}(k_h k_m)k_d$	28.57	79.31	86.36	65.22	64.00	**64.69**
$\kappa_H^{DS}(k_h k_m)k_d$	38.10	51.72	81.82	73.91	72.00	63.51

(a)

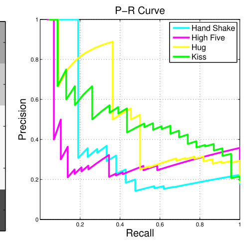
RETRIEVAL PRECISION

Kernel \ Class	HS	HF	HG	KS	NG	AVG
No Proximity						
k_S	0.239	0.316	0.293	0.402	-	0.314
$\kappa_H^{TL}(k_S)$	0.208	0.335	0.265	0.498	-	0.330
$\kappa_H^{DS}(k_S)$	0.199	0.300	0.267	0.424	-	0.300
$\kappa_H^{TL}(k_h k_m)$	0.267	0.264	0.277	0.523	-	0.330
$\kappa_H^{DS}(k_h k_m)$	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
$\kappa_H^{TL}(k_S)k_d$	0.334	0.333	0.485	0.482	-	0.412
$\kappa_H^{DS}(k_S)k_d$	0.339	0.351	0.538	0.483	-	0.424
$\kappa_H^{TL}(k_h k_m)k_d$	0.342	0.357	0.551	0.525	-	0.439
$\kappa_H^{DS}(k_h k_m)k_d$	0.351	0.338	0.554	0.540	-	**0.440**

(b)



(c)



(d)

Fig. 5. Recognition and retrieval on TVHI. (a) Recognition accuracy. (b) Video retrieval average precision. For both (a) and (b), MH features are computed with $\tau = 5$ and $\delta = 3$; HOOF features are computed with $b = 10$; KSS order is set to $n = 10$. (c) Confusion matrix for the best performance. (d) Per-class precision-recall curves for the best performance. (c) and (d) Use the K_{KSS} kernel.

TABLE II

RECOGNITION ACCURACY ON THE UT-I DATA SET FOR THE K_{KSS} KERNEL, USING PROXIMITY AND DIFFERENT MOTION FEATURES, INCLUDING ONLY MHs, ONLY HOOF FEATURES, AND BOTH. MOTION FEATURES ARE COMPUTED AS INDICATED IN FIG. 4

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

TABLE III

RECOGNITION ACCURACY FOR THE K_{KSS} AND THE K_{GKSS} KERNELS ON DIFFERENT DATA SETS AND COMPARISON WITH THE STATE OF THE ART

	UT-I SET 1	UT-I SET 2	TVHI	BIT-I	SBU
Patron-Perez et al. [3]	84	86	48.9	-	-
Yu et al. [8]	93.33	91.70	-	-	-
Kong et al. [7]	91.67 (set 1 and 2 combined)		-	90.63	-
Yun et al. [54]	-	-	-	-	80.20
K_{KSS} (17)	91.80	87.87	64.69	91.00	81.21
K_{GKSS} (18)	95.08	89.39	64.16	92.21	78.36

TABLE IV

AUDIO–VIDEO RECOGNITION ON TVHI. RECOGNITION ACCURACY USING THE AUDIO FEATURES ALONE AND THE COMBINED AUDIO AND VIDEO FEATURES

	Audio Linear	Audio RBF (35)	Audio and Video $\kappa_v \kappa_a$	Audio and Video MKL
K_{KSS} (17)	45.11	44.16	64.61	-
K_{GKSS} (18)	45.83	48.33	65.20	**72.00**
BoW [58]	-	-	50.86	

between two interaction trajectories becomes $d(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\}) \doteq 2(1 - \tilde{K}(\{\mathbf{y}_t\}, \{\mathbf{y}'_t\})$. Fig. 5(b) and (d) shows the retrieval precision and the per-class precision-recall curves, as defined in [56]. It can be seen that even with such a simple approach, the results are comparable to those in [3]. We expect that by using the proposed kernels in a "learning to rank" approach [57], the retrieval precision would undergo a substantial increase.

For various kernels, Table II shows how classification performance on the UT-I data set 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 shows that the proposed MH features capture motion history information that is as discriminative as the information captured by the HOOF. Also, the two information sources are only partially correlated, given the significant boost in classification accuracy in the latter case.

In the rest of this paper, we use the pairwise kernel indicated as $\kappa_H^{\text{TL}}(k_h k_m)k_d$ for video trajectories, since it consistently provides the best performance. Although a direct comparison against the state of the art is rather difficult, given the differences in the representation, features used, the use of tracking, and so on, Table III reports a summary of recently reported classification accuracies, where we also added the results of our approach on the BIT-I and the SBU data sets. Our approach performs comparably and often better than recent methods [3]–[5], [7], [8], which is promising. Figs. 4(c) and (d) and 5(c) show the confusion matrices corresponding to the best classification accuracy obtained with the K_{KSS} kernel.

2) Recognition With Audio–Video Trajectories: Since the TVHI data set includes also the audio track, we have used it for recognition in conjunction with the video. Besides the video trajectories, we have extracted the audio trajectories as explained in Section III-D. We have conducted recognition experiments with only audio trajectories, and with the combined audio–video trajectories. Audio trajectories alone are modeled with a KSS model with a kernel κ_a that is either linear or RBF, and is indicated with K_{KSS} in Table IV. In addition, we have also used the K_{GKSS} kernel (18).

Audio–video trajectories are used for recognition in three different ways. First, we augment the K_{KSS} kernel (17) with the base kernel κ that is the TP of the video kernel κ_v, and the audio kernel κ_a. Second, just as for the audio case, we have also used an RBF kernel based on the distance derived from the K_{KSS} kernel. Finally, we also treat the audio and video trajectories independently, and compare them with the respective K_{GKSS} kernels, and use multiple kernel learning (MKL) to do classification based on audio and video information. Table IV shows that the simple TP augmentation does not improve the recognition rate based on video trajectories only [see Fig. 5(a), where $\kappa_v = \kappa_H^{\text{TL}}(k_h k_m)k_d$)] and that using the RBF kernel gives a small improvement. A more substantial improvement instead comes from using MKL on the two distinct audio and video trajectories. Finally, in Table IV, we also report the results on audio and video human interaction recognition

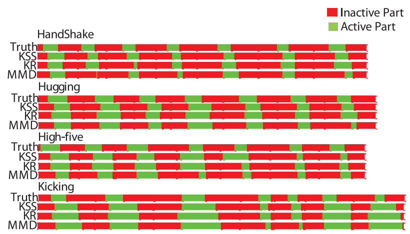


Fig. 6. Temporal detection for HAUS-PI data set. Detection results using the MMD, the KR, and the KSS models in comparison with the GT annotations, for seven sequences randomly selected from four classes. The active parts indicate when an interaction is happening.

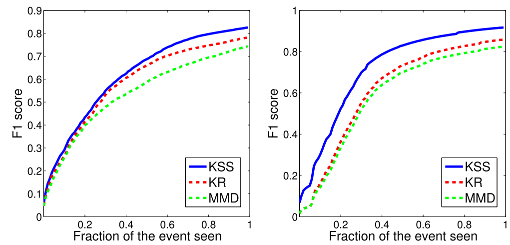


Fig. 7. Single camera view F1 scores. F1 score curves for HAUS-PI (left) and UT-I (right) data sets. Larger values of the F1 score for a given fraction of the interaction indicate better localization of the ongoing event.

computed by [58], where we note that the two approaches are significantly different, since the first one relies on a bag-of-word model, whereas ours use different features and relies on tracking information, and are not directly comparable.

3) Recognition With 3D Joint Trajectories: Inspired by [54], we extend our method to work with the 3D joint positions provided by the SBU data set. We compute *joint distance* features as follows. At time t, we compute $\mathbf{j}_{i,t}$, the set of distances between every two joints of skeleton i, and $\mathbf{j}_{ij,t}$, the set of distances between joints of skeletons i and j. We also compute *joint motion* features as follows. At time t, we compute $\mathbf{k}_{i,t}$, the set of distances between joints of skeleton i at time t and at time $t-1$, and we compute $\mathbf{k}_{ij,t}$, the set of distances between joints of skeleton i at time t and skeleton j at time $t-1$. Therefore, *3D joint interaction trajectories* are defined by $\mathbf{y}_{ij,t} \doteq [\mathbf{j}_{i,t}^\top, \mathbf{k}_{i,t}^\top, \mathbf{j}_{j,t}^\top, \mathbf{k}_{j,t}^\top, \mathbf{j}_{ij,t}^\top, \mathbf{k}_{ij,t}^\top, \mathbf{k}_{ji,t}^\top]^\top$. We followed the experimental setup of [54] and we used $\kappa_H^{\mathrm{TL}}(k_{B_\mathbf{j}} k_{B_\mathbf{k}} k_P)$ as kernel, where $k_{B_\mathbf{j}}$, $k_{B_\mathbf{k}}$, and k_P are Gaussian RBF kernels with Euclidean distance, acting on the body shape features (i.e., $\mathbf{j}_{i,t}$, $\mathbf{j}_{j,t}$), on the body motion features (i.e., $\mathbf{k}_{i,t}$, $\mathbf{k}_{j,t}$), and on the proxemic features (i.e., $\mathbf{j}_{ij,t}$, $\mathbf{k}_{ij,t}$, $\mathbf{k}_{ji,t}$), respectively. Table III shows that our method compares favorably against [54].

4) Detection Results: For the approach to work online, we have implemented the online kernel PCA algorithm in [43], and the matrix A was recursively updated online as explained in [44]. Computationally, the bottleneck is the extraction of the motion features, but their GPU implementation can work in real time. The rest of the algorithm, currently implemented in MATLAB, runs at more than 30 frames/s on a high-end workstation.

For the detection evaluation, we mostly follow the protocol introduced in [14]. We use the UT-I, and the HAUS-PI data sets, and we compare three methods: the approach described in [15], indicated as MMD, the KR model, (13), and the KSS model, (14). Fig. 6 shows detection results using the MMD, the KR, and the KSS models in comparison with the GT annotations, for seven sequences randomly selected from the 12 annotated classes of the HAUS-PI data set. The active parts (in green box) indicate when an interaction is happening. Fig. 7 shows the F1 score, which indicates how well an interaction is localized on the temporal axis. As expected, the KSS is clearly the best performer on both the UT-I as well as the HAUS-PI data sets. Also, even though the KR model works under the same assumptions as the MMD,

TABLE V

RI FOR HAUS-PI AND UT-I. THIS IS A MEASURE OF THE SIMILARITY BETWEEN TWO DATA CLUSTERINGS. WE COMPUTED THE RI OF THE INTERACTION SEGMENTATIONS AGAINST THE GT LABELS. A HIGHER RI MEANS BETTER INTERACTION LOCALIZATION

RI	HAUS-PI	UT-I
MMD	0.70	0.68
KR	0.71	0.69
KSS	0.72	0.72

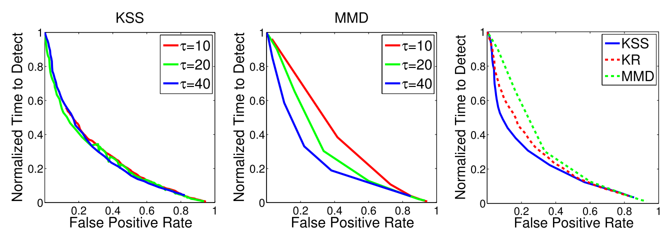


Fig. 8. AMOC curves for the HAUS-PI. Sensitivity of the normalized time to detection with respect to the length τ of the test time window, for the KSS model (left) and for the MMD model (center). Right: comparison between the KSS, KR, and MMD models.

we do observe better performance. Another measure for the localization accuracy is given by the Rand index (RI), which is included in Table V. Even according to this measure, we obtain a similar comparison between the models.

Fig. 8 shows the normalized time to detection (see [14] for definition), which indicates the timeliness with which the beginning of an interaction is identified. In particular, the left and the center plot clearly show that the KSS approach is much less sensitive to the length τ of the test time window. A similar plot was obtained with the KR model, but was omitted for luck of space. Fig. 8 on the right instead shows that the KSS model compares favorably against the others, and so is about the KR versus the MMD model.

Table VI shows the recognition accuracy on the HAUS-PI after the interaction detection with the MMD, the KR, and the KSS models, and also with the GT segmentation annotation. The detection based on the KSS model (here also used for recognition) and the GT annotations lead to very similar recognition accuracies. We also computed the recognition accuracies for the UT-I data set and obtained 95.08% for Set 1, and 89.39% for Set 2, by using the KSS model for detection and recognition with the RBF kernel (18).

TABLE VI

CLASSIFICATION ACCURACY FOR THE HAUS-PI DATA SET OBTAINED AFTER THE TEMPORAL DETECTION

MMD	KSS	KR	GT
50.87	54.15	53.75	54.09

TABLE VII

RECOGNITION ACCURACY AND RI OF MVHAUS-PI DATA SET FOR THE K_{KSS} KERNEL AND FOR DIFFERENT SCENARIOS. GT STANDS FOR ACCURACY USING GT SEGMENTATION AND DI STANDS FOR ACCURACY USING DETECTED INTERACTIONS

MVHAUS-PI Accuracy and Rand Index				
Scenario	Details	GT	DI	RI
Single View	View 1	36.74	33.55	0.60
	View 2	34.35	32.11	0.58
	View 3	38.66	34.50	0.61
Two Views	*TP* with views 1 and 2	47.44	46.33	0.65
	TP with views 1 and 3	47.28	46.17	0.65
	TP with views 2 and 3	44.73	43.93	0.63
Three Views	*TP* with all views	60.38	58.47	0.76
	DS with all views	51.44	49.68	0.71
	KMCCA with all views	58.14	57.34	0.73

B. Multiple Camera Views Results

Here, we have used only the MVHAUS-PI data set. For interaction recognition, we have used the temporal segmentation provided as ground truth (indicated as GT in Table VII), and also the one computed with the KSS model (indicated as DI in Table VII). We have considered three scenarios, called *Single View*, *Two Views*, and *Three Views*. The Single View scenario uses the data coming from each camera view separately, and performs joint interaction detection and recognition as described in Sections V and VI. The Two Views scenario considers every pair of camera views, and uses the *TP* fusion approach, (36), for detection and recognition. The Three Views scenario considers all the camera views, and we use each of the fusion approaches for detection and recognition explained in Section VIII, namely, the TP approach, the DS, (37), and the *KMCCA* approach, (39).

From Table VII, we confirm that adding camera views increases the recognition accuracy. In addition, also, the detection accuracy increases. Indeed, a better detection should be responsible for a portion of the recognition improvement. In addition, the recognition with the automatic detection (DI) reaches, on average, 91.26% of the accuracy obtained with the GT segmentation in the Single View scenario, 97.83% in the Two Views scenario, and 97.37% in the Three Views scenario. We also note that the recognition accuracy of the Single View scenario is much lower than the accuracy of the HAUS-PI. This is mainly due to the higher number of classes (16 instead of 12), but also due to the higher resolution of the HAUS-PI data set, and a different number of videos per class, since the interactions in the MVHAUS-PI are required to be visible from all the camera views.

Fig. 9 shows the confusion matrices for the best Single View scenario (View 3), the best Two Views scenario (Views 1 and 2), and the best Three Views fusion (using TP). In particular, we note that although using the KMCCA greatly improves upon the simple DS fusion, it is the simple TP fusion that provides the best recognition accuracy, as well as the best RI for detection (see Table VII). Finally, Fig. 10 shows the AMOC curves and F1 scores for the Three Views scenario, where the three approaches are compared, confirming that the KMCCA greatly improves upon the DS, but the TP approach improves even further, despite its simplicity.

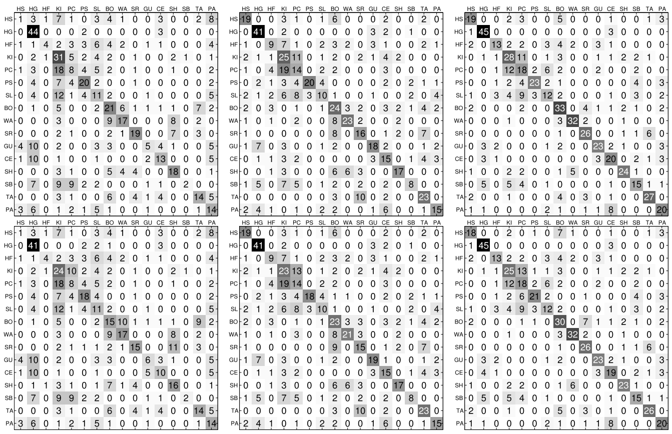


Fig. 9. MVHAUS-PI data set. Top row: confusion matrices obtained with the K_{KSS} kernel using detected interactions for the best Single View scenario (View 3, left), Two Views scenario (Views 1 and 2, middle), and Three Views fusion (TP, right). Bottom row: corresponding matrices obtained using GT segmentation.

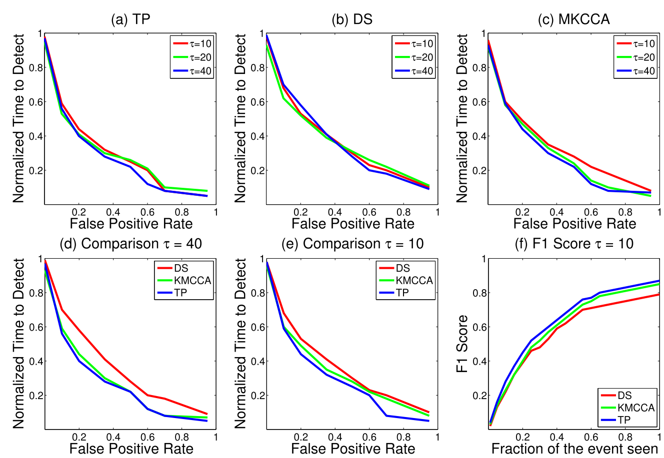


Fig. 10. MVHAUS-PI—AMOC and F1 scores for three views. (a)–(c) AMOC curves for *TP*, DS, and *MKCCA* fusion, respectively. (d) and (e) comparison between different fusion approaches for $\tau = 40$ and $\tau = 10$. (f) F1 score for different approaches with $\tau = 10$. The curves show that *TP* works slightly better than the others.

X. CONCLUSION

In this paper, we have introduced a modeling framework for the online detection and recognition of human interactions from multiple cameras. We have made extensive use of kernel methods to optimize the recognition performance based on balanced pairwise kernels and we have extended established detection techniques for their use with KSS models. In addition, we have introduced a new data set for testing the proposed framework on data acquired from multiple views. Our current implementation shows that results are comparable with or better than the state of the art (which are available only for single view approaches, since there are no other approaches for multiple camera views) and that the speed of the algorithms should allow the implementation in real time of the proposed approach. Note, however, that there are several important issues that have not been investigated. For instance, we have not studied to what extent the approach might be able to scale with the number of people in the scene, and with the number of cameras in the network. Another issue pertains the decentralization of the computational operations. And one more is about the robustness against tracking errors, or the use of fragmented tracks. While we believe that the proposed framework is very promising, and could become an important part of a system for the analysis in real time of human behavior, we are also aware that the issues mentioned earlier need to be investigated in future research.

References

[1] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, Apr. 2011, Art. no. 16.

[2] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *Proc. ICCV*, 2009, pp. 1593–1600.

[3] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman, "Structured learning of human interactions in TV shows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2441–2453, Dec. 2012.

[4] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *Proc. ICCV*, 2011, pp. 778–785.

[5] U. Gaur, Y. Zhu, B. Song, and A. Roy-Chowdhury, "A 'string of feature graphs' model for recognition of complex activities in natural videos," in *Proc. ICCV*, 2011, pp. 2595–2602.

[6] Y. Kong, Y. Jia, and Y. Fu, "Learning human interaction by interactive phrases," in *Proc. ECCV*, 2012, pp. 300–313.

[7] Y. Kong, Y. Jia, and Y. Fu, "Interactive phrases: Semantic descriptions for human interaction recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 9, pp. 1775–1788, Sep. 2014.

[8] G. Yu, J. Yuan, and Z. Liu, "Propagative Hough voting for human activity recognition," in *Proc. ECCV*, 2012, pp. 693–706.

[9] F. Zhou, F. D. L. Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 582–596, Mar. 2013.

[10] J. Barbič, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins, and N. S. Pollard, "Segmenting motion capture data into distinct behaviors," in *Proc. Graph. Interface*, 2004, pp. 185–194.

[11] Z. Wang, J. Wang, J. Xiao, K.-H. Lin, and T. Huang, "Substructure and boundary modeling for continuous action recognition," in *Proc. CVPR*, 2012, pp. 1330–1337.

[12] S. Satkin and M. Hebert, "Modeling the temporal extent of actions," in *Proc. ECCV*, 2010, pp. 536–548.

[13] S. M. Oh, J. M. Rehg, T. Balch, and F. Dellaert, "Learning and inferring motion patterns using parametric segmental switching linear dynamic systems," *Int. J. Comput. Vis.*, vol. 77, nos. 1–3, pp. 103–124, 2008.

[14] M. Hoai and F. De la Torre, "Max-margin early event detectors," in *Proc. CVPR*, 2012, pp. 2863–2870.

[15] D. Gong, G. Medioni, S. Zhu, and X. Zhao, "Kernelized temporal cut for online temporal segmentation and recognition," in *Proc. ECCV*, 2012, pp. 229–243.

[16] B. Schölkopf and A. J. Smola, *Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[17] L. Ljung, *System Identification: Theory for the User*, 2nd ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 1999.

[18] E. Y. Chow and A. S. Willsky, "Analytical redundancy and the design of robust failure detection systems," *IEEE Trans. Autom. Control*, vol. 29, no. 7, pp. 603–614, Jul. 1984.

[19] J. Rupnik and J. Shawe-Taylor, "Multi-view canonical correlation analysis," in *Proc. SiKDD*, 2010, pp. 1–4.

[20] S. Motiian, K. Feng, H. Bharthavarapu, S. Sharlemin, and G. Doretto, "Pairwise kernels for human interaction recognition," in *Advances in Visual Computing*, vol. 8034. Berlin, Germany: Springer, 2013, pp. 210–221.

[21] F. Siyahjani, S. Motiian, H. Bharthavarapu, S. Sharlemin, and G. Doretto, "Online geometric human interaction segmentation and recognition," in *Proc. IEEE ICME*, Jul. 2014, pp. 1–6.

[22] A. B. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," in *Proc. CVPR*, 2005, pp. 846–851.

[23] R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, "Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions," in *Proc. CVPR*, 2009, pp. 1932–1939.

[24] Z. Harchaoui, F. Bach, O. Cappe, and E. Moulines, "Kernel-based methods for hypothesis testing: A unified view," *IEEE Signal Process. Mag.*, vol. 30, no. 4, pp. 87–97, Jul. 2013.

[25] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola, "A kernel method for the two-sample-problem," in *Proc. NIPS*, 2007, pp. 513–520.

[26] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in *Proc. ECCV*, 2014, pp. 596–611.

[27] Y. Cao *et al.*, "Recognize human activities from partially observed videos," in *Proc. IEEE CVPR*, Jun. 2013, pp. 2658–2665.

[28] D.-A. Huang and K. M. Kitani, "Action-reaction: Forecasting the dynamics of human interaction," in *Proc. ECCV*, 2014, pp. 489–504.

[29] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori, "Discriminative latent models for recognizing contextual group activities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 8, pp. 1549–1562, Aug. 2012.

[30] W. Choi and S. Savarese, "Understanding collective activities of people from videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1242–1257, Jun. 2014.

[31] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *Proc. IEEE CVPR*, Jun. 2013, pp. 2642–2649.

[32] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang, "Action detection in complex scenes with spatial and temporal ambiguities," in *Proc. IEEE ICCV*, Sep./Oct. 2009, pp. 128–135.

[33] J. Yuan, Z. Lin, and Y. Wu, "Discriminative video pattern search for efficient action detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 9, pp. 1728–1743, Sep. 2011.

[34] M. Jain, J. van Gemert, H. Jègou, P. Bouthemy, and C. G. M. Snoek, "Action localization with tubelets from motion," in *Proc. IEEE CVPR*, Jun. 2014, pp. 740–747.

[35] D. Tran, J. Yuan, and D. Forsyth, "Video event detection: From sub volume localization to spatiotemporal path search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 2, pp. 404–416, Feb. 2014.

[36] M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proc. CVPR*, 2008, pp. 1–8.

[37] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. CVPR*, 2011, pp. 3169–3176.

[38] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. ICCV*, 2013, pp. 3551–3558.

[39] G. Doretto, T. Sebastian, P. Tu, and J. Rittscher, "Appearance-based person reidentification in camera networks: Problem overview and current approaches," *J. Ambient Intell. Humanized Comput.*, vol. 2, no. 2, pp. 127–151, 2011.

[40] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 28, no. 4, pp. 357–366, Aug. 1980.

[41] G. Doretto, A. Chiuso, Y. N. Wu, and S. Soatto, "Dynamic textures," *Int. J. Comput. Vis.*, vol. 51, no. 2, pp. 91–109, 2003.

[42] A. B. Chan and N. Vasconcelos, "Classifying video with kernel dynamic textures," in *Proc. CVPR*, 2007, pp. 1–6.

[43] L. Hoegaerts, L. De Lathauwer, I. Goethals, J. A. K. Suykens, J. Vandewalle, and B. De Moor, "Efficiently updating and tracking the dominant kernel principal components," *Neural Netw.*, vol. 20, no. 2, pp. 220–229, 2007.

[44] S. J. Kim, G. Doretto, J. Rittscher, P. Tu, N. Krahnstoever, and M. Pollefeys, "A model change detection approach to dynamic scene modeling," in *Proc. AVSS*, Sep. 2009, pp. 490–495.

[45] K. De Cock and B. De Moor, "Subspace angles between ARMA models," *Syst. Control Lett.*, vol. 46, no. 4, pp. 265–270, 2002.

[46] S. V. N. Vishwanathan, A. J. Smola, and R. Vidal, "Binet–Cauchy kernels on dynamical systems and its application to the analysis of dynamic scenes," *Int. J. Comput. Vis.*, vol. 73, no. 1, pp. 95–119, 2007.

[47] W. Li and N. Vasconcelos, "Recognizing activities by attribute dynamics," in *Proc. NIPS*, 2012, pp. 1106–1114.

[48] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[49] A. Srivastava, I. Jermyn, and S. Joshi, "Riemannian analysis of probability density functions with applications in vision," in *Proc. CVPR*, 2007, pp. 1–8.

[50] C. Brunner, A. Fischer, K. Luig, and T. Thies, "Pairwise support vector machines and their application to large scale problems," *J. Mach. Learn. Res.*, vol. 13, pp. 2279–2292, Jan. 2012.

[51] A. Ben-Hur and W. S. Noble, "Kernel methods for predicting protein–protein interactions," *Bioinformatics*, vol. 21, no. 1, pp. i38–i46, Jun. 2005.

[52] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, "Canonical correlation analysis: An overview with application to learning methods," *Neural Comput.*, vol. 16, no. 12, pp. 2639–2664, 2004.

[53] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *Proc. IEEE ICCV*, Oct. 2007, pp. 1–7.

[54] K. Yun, J. Honorio, D. Chattopadhyay, T. L. Berg, and D. Samaras, "Two-person interaction detection using body-pose features and multiple instance learning," in *Proc. IEEE CVPRW*, Jun. 2012, pp. 28–35.

[55] C. Vondrick, D. Patterson, and D. Ramanan, "Efficiently scaling up crowdsourced video annotation," *Int. J. Comput. Vis.*, vol. 101, no. 1, pp. 184–204, 2013.

[56] S. Büttcher, C. L. A. Clarke, and G. V. Cormack, *Information Retrieval*. Cambridge, MA, USA: MIT Press, 2010.

[57] Z. Cao, T. Qin, T.-Y. Liu, M.-F. Tsai, and H. Li, "Learning to rank: From pairwise approach to listwise approach," in *Proc. ICML*, 2007, pp. 129–136.

[58] M. J. Marín-Jiménez, R. Muñoz-Salinas, E. Yeguas-Bolivar, and N. P. de la Blanca, "Human interaction categorization by using audio-visual cues," *Mach. Vis. Appl.*, vol. 25, no. 1, pp. 71–84, Jan. 2014.

Saeid Motiian received the B.Sc. degree in electrical engineering from Iran University of Science and Technology, Tehran, Iran, in 2009 and the M.Sc. degree from the University of Tehran, Tehran, in 2011. He is currently working toward the Ph.D. degree in electrical engineering with West Virginia University, Morgantown, WV, USA.

His research interests include modeling, detection, and recognition of human activities.

Farzad Siyahjani (S'12) received the B.Sc. degree in electrical engineering from Sahand University, Tabriz, Iran, in 2008 and the M.Sc. degree in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2010. He is currently working toward the Ph.D. degree in electrical engineering with West Virginia University, Morgantown, WV, USA.

He is collaborating with SleepIQ Labs, San Jose, CA. His research interests include machine learning, artificial intelligence, signal and image processing, and pattern recognition.

Ranya Almohsen received the B.Sc. degree in computer science from King Saud University, Riyadh, Saudi Arabia, and the M.S. degree in computer science from West Virginia University, Morgantown, WV, USA, in 2014, where she is currently working toward the Ph.D. degree in computer science.

Her research interests include computer vision, with a focus on human behavior analysis and person reidentification.

Gianfranco Doretto (M'06) received the Ph.D. degree in computer science from University of California at Los Angeles, Los Angeles, CA, USA, in 2005.

He was a Lead Scientist with GE Global Research, Garching bei München, Germany, until 2010. He is an Associate Professor with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA. His research interests include computer vision, with a focus on statistical modeling for video analysis.