MoCap multichannel time series representation and relevance analysis by kernel adaptive filtering and multikernel learning oriented to action recognition tasks

J. D. Pulgarin-Giraldo[1,2](✉), A. M. Alvarez-Meza[3], S. Van Vaerenbergh[4], I. Santamaría[4], and G. Castellanos-Dominguez[2]

[1] G-BIO Research Group, Universidad Autónoma de Occidente, Cali, Colombia.
[2] Signal Processing and Recognition Group, Universidad Nacional de Colombia, Manizales, Colombia.
jdpulgaring@unal.edu.co
[3] Faculty of Engineering, Universidad Tecnológica de Pereira, Pereira, Colombia.
[4] Dept. of Communications Engineering, University of Cantabria, Santander, Spain.

Abstract. A framework based in kernel adaptive filters and multikernel learning for MoCap multichannel data is presented. In this sense, kernel adaptive filters are used to encode the dynamic of each channel. Then, a model for each time series is constructed with a codebook and latent functions estimated by KRLS tracker algorithm. These independent channel representations assemble similarity between multiple realizations in a RKHS thanks to Maximum Mean Discrepancy criterion. Later on, a kernel alignment algorithm is used to assemble multiple channels in a unique kernel that relates all realizations. Supervised classification over this kernel shows a good assembling for different actions realizations. Moreover, relevance estimated by the kernel alignment highlights the most significant channels in action realizations. Results show that our methodology easily constructs a good representation for MoCAp multiple channel data, and results agree with the findings made in the biomechanical analysis for our kind of data: tennis stroke's records.

Keywords: Multichannel data, kernel adaptive filters, maximum mean discrepancy, center kernel alignment.

1 Introduction

Human action recognition from Motion Capture(MoCap) data is a well-established area in pattern recognition [1, 2]. To date, the main efforts are directed at creating a sufficiently robust dynamic model of human movement accomplished under a priori given actions. All of these models have been validated in predicting the movement and/or action recognition through a certain (di)similarity measure, often in a Reproducing Kernel Hilbert Space (RKHS) due to the nonlinear dynamics in biomechanic action generation. However, the models are mostly oriented to classify with a high accuracy the executed action rather than to observe the relevance of each channel. Relevance studies in human action recognition are oriented to highlight the most useful features in feature extraction, but the body segments and articulations were the sensors are placed are not studied. For this reason, in addition to predict and classify, these techniques must define in some way the relevance of each channel in the task.

Kernel adaptive filters (KAFs) are nonlinear adaptive filters based on the framework of kernel methods [3]. They are frequently used in problems of nonlinear time series prediction. Besides, KAFs not only are a well-established methodology in time series prediction but also provide compact *dictionary* or *codebook*. This is an interesting property since at the end the stored elements will be the most representative of the time series for the current task, which is one-step ahead prediction. This property avoids segmentation stages to obtain a suitable data analysis and overpass well-established methods of feature extraction in human action recognition [4, 5].

On the other hand, Hilbert space embeddings are a recent trend in kernel methods that map distributions into infinite-dimensional feature spaces using kernels, such that comparisons and manipulations of these distributions can be performed using standard feature space operations like inner products or projections [6]. Moreover, an existing framework for analyzing distributions and comparing distributions called Maximum Mean Discrepancy (MMD) [7] allows us to evaluate expectations over functions in the unit ball of a RKHS. However, multiple probability distributions analysis or even joint probability distributions analysis based on Hilbert space embeddings are not explored.

Regarding the combination of multiple dynamic models by kernel methods, the most widely studied have been those built by convex combinations of a finite set of base kernels [8, 9, 1]. In classification and regression tasks, the classical uniform combination solution has been improved thanks to the Centered Kernel Alignment (CKA) algorithm proposed by Cortes [10]. This one uses a similarity measure between kernels or kernels matrices to measure the similarity of each base kernel with a target kernel obtained from output labels. CKA is efficient and easy to implement and, additionally, the weights of the combinations provide the relevance of each base kernel. If the base kernels are constructed independently from each channel in a MoCap multichannel time series, these weights somehow reveal the most important channels involved in an action execution.

Here, a methodology for MoCap multichannel data representation is presented without data segmentation. This methodology is oriented to assemble appropriately all channels in a Reproducing Kernel Hilbert Space. This embedding allows aligning kernels with a target kernel with label classes in supervised classification. This kernel alignment procedure not only provides good classification accuracy but also reveals the most significant channels in the action execution associated with the kernel target.

2 Theoretical framework

We assume a scenario in which a set of J time series $\mathbf{x}_j[t]$ are obtained from sensor measurements, with $j = 1, \ldots, J$. For each time series, T time steps are available, i.e. $t = 1, \ldots, T$. We collect the entire set of measurements in the matrix $\boldsymbol{X} \in \mathbb{R}^{J \times T}$, which contains the J time series as its rows,

$$\boldsymbol{X} = \begin{bmatrix} x_1[1] & x_1[2] & \ldots & x_1[T] \\ x_2[1] & x_2[2] & \ldots & x_2[T] \\ \vdots & \vdots & \ddots & \vdots \\ x_J[1] & x_J[2] & \ldots & x_J[T] \end{bmatrix} \tag{1}$$

We further assume that multiple such sets are available. The n-th set is represented as $\boldsymbol{X}^n$, with $n = 1, \ldots, N$, and to indicate that a time series belongs to a particular set n we use the notation $\mathbf{x}_j^n[t]$.

Our goal is to develop a similarity measure between different such multichannel time series and to perform different types of analyses with this measure. To this end, we will first represent each individual time series $\mathbf{x}_j^n[t]$ as a compact model $\mathcal{M}_j^n$, and then we will define a similarity measure between sets of these models.

2.1 Dynamical channel model encoded by kernel adaptive filtering

With the aim of properly modeling each individual time series $\mathbf{x}_j^n[t]$, we will represent its dynamic behavior through kernel adaptive filters (KAFs) [3]. For the sake of clarity, in the following we will omit the superscript n until section Section 2.3.

In kernel methods, the Representer Theorem allows us to express the nonlinearities of a wide range of problems as a kernel expansion in terms of the training data

$$f(\mathbf{x}_j) = \sum_{m=1}^{M} \alpha_m \kappa(\mathbf{x}_j[m], \mathbf{x}_j), \tag{2}$$

where $\boldsymbol{x}_j[m]$ represents the m-th training input, and $\kappa(\cdot,\cdot)$ is a positive semidefinite *kernel* function. When the desired outputs $y_j[m]$ are available for the training data, these can be used to estimate the optimal expansion coefficients α_m. Given a time series $\mathbf{x}_j[t]$, for $t = 1, \ldots, T$, the problem of one-step ahead prediction can be formulated as a regression problem with input-output pairs $\{\boldsymbol{x}_j[t], y_j[t]\}$ in which the input data is taken as the time-embedded version of the series with L lags, $\boldsymbol{x}_j[t] = [x_j[t], x_j[t-1], \ldots, x_j[t-L+1]]$, and the desired output is the next sample, $y_j[t] = x_j[t+1]$.

We will employ kernel adaptive filtering techniques to solve the described regression problem. These algorithms estimate a model of the form (2) by minimizing the least-squares error between the true labels $y_j[m]$ and their predictions $f(\mathbf{x}_j[m])$. Furthermore, they do so in an online fashion, i.e. by performing one or more passes over the data, which is preferred if not all data fits in memory.

KAF algorithms limit the amount of stored training data $\boldsymbol{x}_j[m]$ by constructing a compact *dictionary* or *codebook*.

Among KAF algorithms, we are interested in those that only store a fixed amount R of training data in the codebook, which is referred to as the *budget*. In particular, we will use the Kernel Recursive Least-Squares Tracker (KRLST) algorithm [11], which represents the state of the art in kernel adaptive filtering. KRLST allows to maintain a fixed budget during operation, and it obtains high accuracy in a wide range of regression and prediction tasks. The prediction of KRLST represents an estimate of the latent noiseless time series, which is unobserved. KRLST is based on a probabilistic input-output model, which provides some additional advantages compared to other KAFs. A Matlab implementation of KRLST and other KAF algorithms is available in [3].

The steps followed to obtain the dynamical channel model $\mathcal{M}_j$ for a time series $\mathbf{x}_j[t]$ can be summarized as follows:

1. Construct input-output pairs $\{\boldsymbol{x}_j[t], y_j[t]\}$ with time-embedding L for the inputs.
2. Run a training pass over all $\{\boldsymbol{x}_j[t], y_j[t]\}$ using KRLST.

The final model for each time series consists in the codebook data selected by KRLST out of the observed inputs, which we will refer to as the *centers* or *centroids* $\boldsymbol{c}_j[r]$, and their corresponding estimated latent function outputs, or *desired values* $d_j[r]$:

$$\mathcal{M}_j = \{\boldsymbol{c}_j[r], d_j[r]\}, \quad r = 1, \ldots, R. \tag{3}$$

2.2 Similarity measure between models

Let us consider two different models $\mathcal{P} = \{\mathbf{p}_r\}_{r=1}^{P}$ and $\mathcal{Q} = \{\mathbf{q}_r\}_{r=1}^{Q}$. Each model represents the dynamic behavior of a given univariate time series and it is composed of a sequence of ordered pairs of codebook elements and output latent functions obtained by the KRLST algorithm; that is, $\mathbf{p}_r = (\mathbf{c}_p[r], d_p[r])$ and $\mathbf{q}_r = (\mathbf{c}_q[r], d_q[r])$. For generality purposes, we assume in this section that each model can have a different number of elements (complexity).

The elements of each model or model samples, as given by the KRLST, are not ordered; therefore, any permutation or reordering of the elements represents the same model. Bearing this in mind, we interpret each model as a cluster of points in the input space. We now define a mapping from the set of models $\mathcal{Z}$ to a RKHS $\mathcal{H}$ as follows

$$\begin{aligned} \Phi : \mathcal{Z} &\longrightarrow \mathcal{H}, \\ \{\mathbf{p}_r\}_{r=1}^{P} &\longmapsto \{\Phi(\mathbf{p}_r)\}_{r=1}^{P} \end{aligned}$$

which maps each model in the input space to a model in the feature space. A model can be interpreted as a distribution function from which P realizations are available. Then, to define a distance between models we resort to the Maximum Mean Discrepancy (MMD) defined by Gretton in [7]. Given two models $\mathcal{P}$ and

$\mathcal{Q}$, the MMD criterion computes the distance between them, $\mathfrak{d}^2(\mathcal{P},\mathcal{Q})$, as the squared Euclidean distance between the sample means of the two distributions, i.e.,

$$\begin{aligned}\mathfrak{d}^2(\mathcal{P},\mathcal{Q}) &= \left\|\frac{1}{P}\sum_{r=1}^{P}\Phi(\mathbf{p}_r)-\frac{1}{Q}\sum_{r=1}^{Q}\Phi(\mathbf{q}_r)\right\|_2^2\\ &= \frac{1}{P^2}\sum_{r=1}^{P}\sum_{r'=1}^{P}\kappa(\mathbf{p}_r,\mathbf{p}_{r'})\;+\;\frac{1}{Q^2}\sum_{r=1}^{Q}\sum_{r'=1}^{Q}\kappa(\mathbf{q}_r,\mathbf{q}_{r'})+\\ &-\frac{2}{PQ}\sum_{r=1}^{P}\sum_{r'=1}^{Q}\kappa(\mathbf{p}_r,\mathbf{q}_{r'}),\end{aligned}\tag{4}$$

where $\kappa(\mathbf{p}_r,\mathbf{p}_{r'})$ is the kernel function between two model samples.

Without loss of generality and to simplify notation, let us denote two model samples as $\mathbf{p}=(\mathbf{c}_p,d_p)$ and $\mathbf{q}=(\mathbf{c}_q,d_q)$. Assuming a separable model that decouples the influence of the input and the output [12], the kernel function between two model samples is,

$$\kappa(\mathbf{p},\mathbf{q})=\kappa(\mathbf{c}_p,\mathbf{c}_q)\;\kappa(d_p,d_q).$$

Assuming a linear kernel for the output and the usual Gaussian kernel for the input, the proposed input-output kernel is finally defined as

$$\kappa(\mathbf{p},\mathbf{q})=\exp\left(-\frac{||\mathbf{c}_p-\mathbf{c}_q||^2}{2\sigma_c^2}\right)d_pd_q.\tag{5}$$

Using this separable kernel, the distance between models in Eq. (4) can be rewritten more compactly in terms of kernel matrices as

$$\mathfrak{d}^2(\mathcal{P},\mathcal{Q})=\frac{1}{P^2}(\mathbf{d}_p^T\boldsymbol{K}_{pp}\mathbf{d}_p)+\frac{1}{Q^2}(\mathbf{d}_q^T\boldsymbol{K}_{qq}\mathbf{d}_q)-\frac{2}{PQ}(\mathbf{d}_p^T\boldsymbol{K}_{pq}\mathbf{d}_q),\tag{6}$$

where $\boldsymbol{K}_{pq}(r,r')=\exp(-\|\mathbf{c}_p[r]-\mathbf{c}_q[r']\|^2/2\sigma_c^2)$.

2.3 Multikernel learning for relevance assessment

In the previous subsection we have introduced a similarity measure between pairs of models each one extracted from a different real-valued time series. Here we extend the procedure to deal with multichannel time series.

Assume that we have a labeled set of J-dimensional time series that we denote as $\mathbf{X}^n\in\mathbb{R}^{J\times T}$, $n=1,\ldots,N$. The j-th row of $\mathbf{X}^n$ is a real-valued time series acquired by the j-th sensor. We first extract a model for each channel using the KRLST algorithm. Then, for the n-th multichannel time series we have a collection of J models that we denote as $\{\mathcal{M}_j[n]\}_{j=1}^J$. With some abuse of notation, let us denote as $\mathbf{K}_j$ the $N\times N$ kernel matrix that measures the (di)similarities for the j-th channel between the N time series in the training data set. The element (n, m) of this kernel matrix is given by

$$\mathbf{K}_j(n,m) = \exp -\left(\frac{\mathfrak{d}^2(\mathcal{M}_j[n], \mathcal{M}_j[m])}{2\sigma_{\mathfrak{d}}^2}\right), \tag{7}$$

where $\mathfrak{d}^2(\mathcal{M}_j[n], \mathcal{M}_j[m])$ is the pairwise distance between models described in Section 2.2 (Eq. (6)).

To combine the information from the J channels we propose to use a multikernel constructed as follows

$$\hat{\boldsymbol{K}} = \sum_{j=1}^{J} \alpha_j \boldsymbol{K}_j, \tag{8}$$

where the weights α_j $j = 1, \ldots, J$ are yet to be determined. A simple solution would be to choose $\alpha_j = \frac{1}{J}$, but this would overlook differences in relevance or discriminative power between the channels. To find more informative weights that allow us to quantify the relevance of individual channels, we propose to use a centered kernel alignment procedure [10]. The basic idea is to find the optimal α_j^* maximizing the alignment between the multikernel matrix $\boldsymbol{K}$ and the target kernel matrix $\boldsymbol{K_l} = \kappa(\boldsymbol{l}, \boldsymbol{l}') = \boldsymbol{l}\boldsymbol{l}^T$, which is calculated from the known label classes $\boldsymbol{l} = \{l[i]\}_{i=1}^N$. For a given set of weights α_j, the centered correlation or alignment between matrix kernels $\boldsymbol{K}$ and $\boldsymbol{K_l}$ is given by

$$\rho\left(\boldsymbol{K}, \boldsymbol{K_l}; \alpha\right) = \frac{\langle \boldsymbol{HKH}, \boldsymbol{HK_lH} \rangle}{\|\boldsymbol{HKH}\|_F \, \|\boldsymbol{HK_lH}\|_F}, \qquad \rho \in [0,1] \tag{9}$$

where $\boldsymbol{H}{=}\boldsymbol{I} - N^{-1}\mathbf{1}\mathbf{1}^\top$ is a centering matrix, $\boldsymbol{I}{\in}\mathbb{R}^{N\times N}$ is the identity matrix, $\mathbf{1}{\in}\mathbb{R}^N$ is an all-ones vector, and notations $\langle \cdot, \cdot \rangle$ and $\|\cdot, \cdot\|_F$ stand for the inner product and the Frobenius norm, respectively.

Then, the optimal relevance weights are $\alpha^* = \text{argmax}\ \rho(\boldsymbol{K}, \boldsymbol{K_l}, \alpha)$ subject to the constraint $||\alpha^*|| = 1$. This problem is solved by the Centered Kernel Alignment (CKA) algorithm [10].

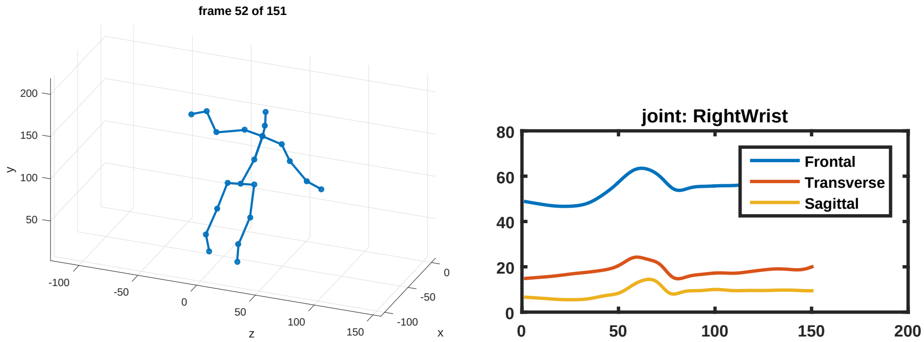
3 Experimental setup

3.1 Database description

The data were collected from 17 high-performance tennis players of the Caldas-Colombia tennis league. The players had the following anthropometric parameters: age 18.9 ± 2.7, mass 64 ± 14.9 *kg*, height 168.8 ± 8.4 cm and all players are right-handed. The employed motion capture protocol was Biovision Hierarchy (BVH) with a full body skeleton of 23 channels. Optitrack Flex V100 (100 *Hz*) infrared videography was collected from six cameras to acquire sagittal, frontal, and lateral planes and skeleton and multichannel time series were estimated in Optitrack Arena®. All subjects were encouraged to hit the ball with the same velocity and action just as they would in a match. They were instructed to hit one series continuously by 30 seconds of each indicated stroke. The strokes indicated in each record were: forehand, backhand, volley, and backhand volley.

3.2 MoCap data

Let $\boldsymbol{U}\in\mathbb{R}^{T\times(J\times D)}$ be a multi-channel input matrix that holds T frames and $J\times D$ channels. Each $\boldsymbol{U}_i = \left\{\mathbf{u}_{ij}\in\mathbb{R}^D : j\in J\right\}$ gathers the skeletal posture at the i-th frame with J D-dimensional body-joints (see Fig. 1(a)). Meanwhile, each $\boldsymbol{U}_j = \left\{\mathbf{u}_{ij}\in\mathbb{R}^D : i\in T\right\}$ assembles time behavior of D-dimensional body-joint j (see Fig. 1(b)).



(a) Skeletal posture $\boldsymbol{U}_i$ (b) Joint behavior $\boldsymbol{U}_j$

Fig. 1. MoCap data. Tennis serve example

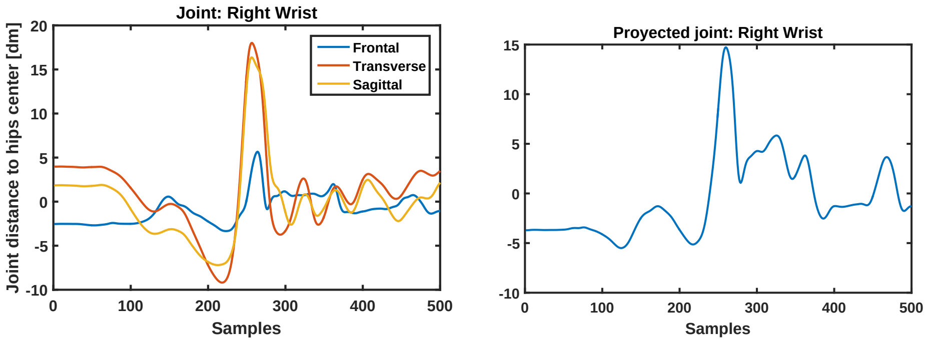
3.3 PCA Preprocessing

Initially, all channels are centered respect to the limb center. Then, to describe the time behavior of the j-th body-joint from $\boldsymbol{U}_j$, we perform a dimensional reduction stage from $\mathbb{R}^D \to \mathbb{R}$ to obtain a compact representation of its time behavior. In this case, from the covariance matrix $\boldsymbol{W}\in\mathbb{R}^{D\times D}$ we consider only the first principal component $\mathbf{w}_1$, obtained from the first eigenvector of the covariance matrix. Then, we obtain the linear projection $\mathbf{x}_j = \boldsymbol{U}_j\mathbf{w}_1$, where $\mathbf{w}_1\in\mathbb{R}^{D\times 1}$ (see Fig. 2).

3.4 Channels dynamic models estimated by KRLS tracker

Before the encoding, each $\mathbf{x}_j$ is decimated by a factor of 5 in order to reduce the computational complexity of the KAF. We compute each model $\mathcal{M}_j$ with KRLST parameters set as follows: forgetting factor 1, time embedding $L = 6$, codebook size $R = 50$, noise to signal ratio $\lambda = 10^{-6}$, a Gaussian kernel with σ calculated as the median value of channel $\mathbf{x}_j$ and the initial codebooks are built directly from the input time series $\mathbf{x}_j\in\mathbb{R}^{T\times 1}$. Each model is validated doing a simple task: predict $x(t+1)$ from data available up to time t.

Fig. 3 shows the mean prediction error in each channel j for all sets of multichannel data, in this case, *N=68*. Although the number of outliers looks high,



(a) Joint behavior $\boldsymbol{U}_j$ (b) Projected body-joint $\mathbf{x}_j$

Fig. 2. Body-joint time behavior projected $\mathbb{R}^{T\times 3} \to \mathbb{R}^{T\times 1}$. Example on Right wrist joint in a serve record

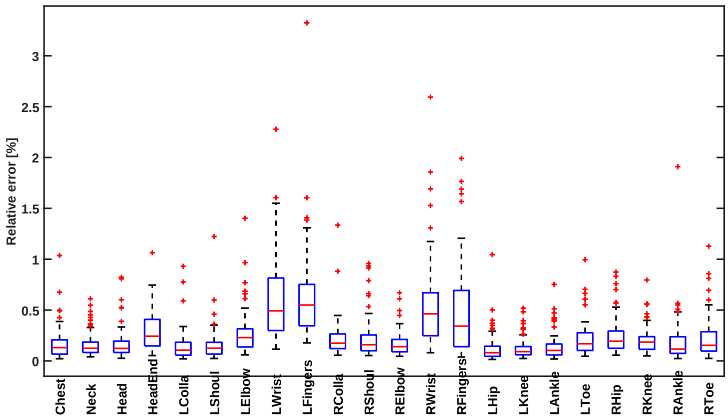
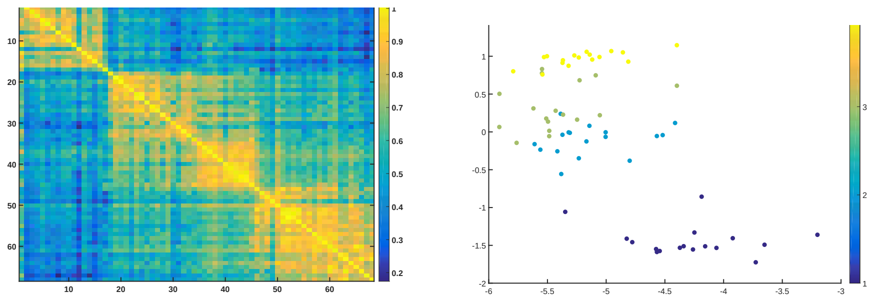


Fig. 3. Relative error results for each joint model $\mathcal{M}_j^n$ estimated over N records with four different classes

it shows a low and regular mean error, which is significant due to the high variability of both: inter-subject and inter-class variability. Besides, our approach works with the 30 seconds full-long one take videos where several and continuous actions were recorded. There are approximately 12 to 16 strokes in each individual record. It is worth saying that segmentation and selection of actions are not required in our modeling process.

3.5 Similarity between models

Our proposed functional $\mathfrak{d}^2$ allows us to construct a kernel similarity measure $\kappa(\mathcal{M}_j[n], \mathcal{M}_j[m])$ which highlights each group of actions without previous information about the classes. In Fig. 4(a) we can see the block diagonal structure of the Gram matrix $\boldsymbol{K}$ constructed over records of the right wrist joint. In fact, KPCA 2D-embedding in Fig. 4(b) shows the separability between groups of records that are colored according to its true label.



(a) Kernel similarity matrix (b) KPCA 2D-embedding

Fig. 4. Model similarity comparison for right wrist body-joint over 68 records. In both plots, the four classes of 17 strokes records are distinguishable

4 Relevance and classification results

The goal of multikernel learning described in Section 2.3 was to develop a supervised classifier for multichannel time series which, at the same time, allows us to assess the relevance of each channel for the given classification problem. Once the multikernel $\hat{\boldsymbol{K}}$ is constructed it allows compare multichannel data, so that we can apply any kernel-based classifier. In this work, we use a kernel nearest neighbor (KNN). The KNN classifier finds the k samples in the training dataset closest to test data (with maximum similarity) and carries out majority vote. Classification performance and relevance are computed using a cross-validation validation scheme.

Fig. 5 shows the attained α values into boxplot depending on the channel relevance approach. Particularly, the body joints at the end of the limbs are the most relevant. These channels highlight the difference between the four classes of action executed. Nonetheless, the variability observed in the most relevant channels implies a strong dependency in the execution, namely, the angle of the racquet in the hit moment varies with the wrist and fingers channels relation.

Regarding to the classification results, as can be seen in Fig. 6(a), accuracies over 90% are attained for almost provided nearest neighbors. In Fig. 6(b), the

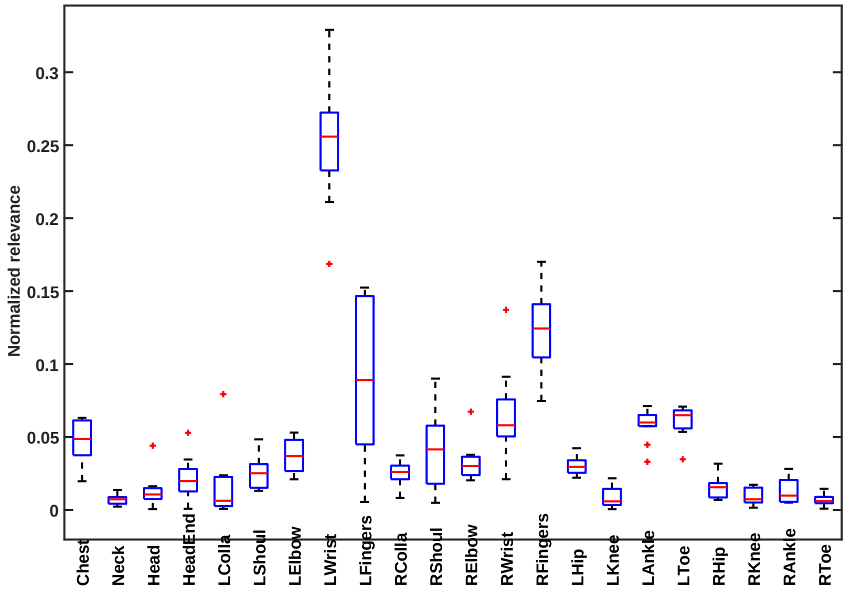
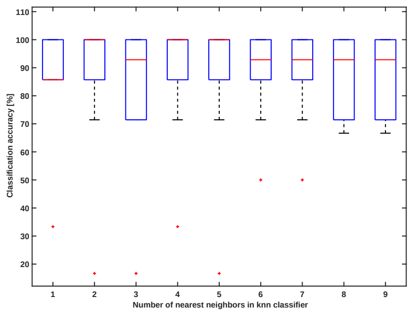


Fig. 5. Relevance body joint analysis in four activities. 10 folds in cross-validation were used over 68 records



(a) Classification performance versus number of nearest neighbors in KNN classifier

	Forehand	Backhand	Volley	BH volley
Forehand	100	0	0	0
Backhand	0	85	10	5
Volley	5	0	75	20
BH volley	0	0	10	90

(b) Confussion matrix with three nearest neighbors. Accuracy results in %

Fig. 6. Classification results in four activities. 10 folds in cross-validation were used over 68 records

lowest results must be analyzed in confrontation with the action, where backhand presents low ball speeds after the impact and it were closer to speeds obtained in volley strokes executions. Nevertheless, each record classified contains 12 to 16 continuously stroke executions without segmentation, so the confused actions depend of execution's speed after 30 seconds.

4.1 Discussion and concluding remarks

The proposed framework for MoCap multichannel analysis presents a methodology that first: obtains an appropriate and individual representation of the dynamic of each channel; and second: this channel representation based on KAFs allows us to combine similarity between several realizations. In fact, this framework easily matches with a multikernel algorithm as CKA, which merges multiple channels into just one kernel that can be used in classification tasks. It can be seen that CKA reveals the most significant channels in a set of actions, and these results are congruent with biomechanic theory in tennis actions execution [13].

This framework should be expanded to analyze the ideal optimal number and placement of sensors in human action recognition tasks, no matters its source; optical markers, inertial sensors or depth cameras. Besides, human motion action involves an interaction between all body segments: every action has a biomechanical chain that produces it, so relevance of channels must give information about the most relevant body segments involved across the time. The results encourage us to develop an algorithm for biomechanical chain generation without kinetic information, just from skeleton representations of actions.

As future work, this framework must be validated in larger action datasets, as well as must be evaluated in assessment motor disorders in order that relevance shows alterations in specific body segments or articulations.

Acknowledgments. This work is supported by the project 111077757982 and the program “Doctorados Nacionales 2014” number 647 funded by COLCIENCIAS, by “Convocatoria nacional para el apoyo a la movilidad internacional 2017-2018” funded by Universidad Nacional de Colombia, as well as PhD financial support from Universidad Autónoma de Occidente.

Bibliography

[1] Salah Althloothi, Mohammad H. Mahoor, Xiao Zhang, and Richard M. Voyles. Human activity recognition using multi-features and multiple kernel learning. *Pattern Recognition*, 47(5):1800 – 1812, 2014.

[2] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal, and R. Bajcsy. Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition. *J. Visual Communication and Image Representation*, 25(1):24–38, 2014.

[3] Steven Van Vaerenbergh and Ignacio Santamaría. A comparative study of kernel adaptive filtering algorithms. In *2013 IEEE Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pages 181–186, August 2013. Software available at `https://github.com/steven2358/kafbox/`.

[4] S. García-Vega, A. M. Álvarez-Meza, and G. Castellanos-Dominguez. *Pattern Recognition and Image Analysis: 7th Iberian Conference, IbPRIA 2015, Santiago de Compostela, Spain, June 17-19, 2015, Proceedings*, chapter

Time-Series Prediction Based on Kernel Adaptive Filtering with Cyclostationary Codebooks, pages 354–361. Springer International Publishing, Cham, 2015.

[5] J. D. Pulgarin-Giraldo, A. M. Alvarez-Meza, L. G. Melo-Betancourt, S. Ramos-Bermudez, and G. Castellanos-Dominguez. *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications - CIARP 2016, Proceedings*, chapter A Similarity Indicator for Differentiating Kinematic Performance Between Qualified Tennis Players, pages 309–317. Springer International Publishing, 2016.

[6] Edgar A. Valencia and Mauricio A. Álvarez. Short-term time series prediction using hilbert space embeddings of autoregressive processes. *Neurocomputing*, 266:595 – 605, 2017.

[7] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13:723–773, March 2012.

[8] Donghui Wu, Zhelong Wang, Ye Chen, and Hongyu Zhao. Mixed-kernel based weighted extreme learning machine for inertial sensor based human activity recognition with imbalanced dataset. *Neurocomputing*, 190:35 – 49, 2016.

[9] Fabio Aiolli and Michele Donini. EasyMKL: a scalable multiple kernel learning algorithm. *Neurocomputing*, 169:215 – 224, 2015.

[10] Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *J. Mach. Learn. Res.*, 13(1):795–828, March 2012.

[11] Steven Van Vaerenbergh, Miguel Lazaro-Gredilla, and Ignacio Santamaria. Kernel recursive least-squares tracker for time-varying regression. *IEEE Transactions on Neural Networks and Learning Systems*, 23(8):1313–1326, Aug 2012.

[12] Mauricio A. Álvarez, Lorenzo Rosasco, and Neil D. Lawrence. Kernels for vector-valued functions: A review. *Found. Trends Mach. Learn.*, 4(3):195–266, March 2012.

[13] J. Landlinger, S. Lindinger, T. Stoggl, H. Wagner, and E. Muller. Key factors and timing patterns in the tennis forehand of different skill levels. *Journal of sports science & medicine*, 9:643–651, 2010.