Kernel Spectral Clustering for dynamic data using Multiple Kernel Learning

D. Peluffo-Ordóñez S. García-Vega R. Langone J. A. K. Suykens G. Castellanos-Domínguez

***Abstract*— In this paper we propose a kernel spectral clustering-based technique to catch the different regimes experienced by a time-varying system. Our method is based on a multiple kernel learning approach, which is a linear combination of kernels. The calculation of the linear combination coefficients is done by determining a ranking vector that quantifies the overall dynamical behavior of the analyzed data sequence over-time. This vector can be calculated from the eigenvectors provided by the the solution of the kernel spectral clustering problem. We apply the proposed technique to a trial from the Graphics Lab Motion Capture Database from Carnegie Mellon University, as well as to a synthetic example, namely three moving Gaussian clouds. For comparison purposes, some conventional spectral clustering techniques are also considered, namely, kernel k-means and min-cuts. Also, standard k-means. The normalized mutual information and adjusted random index metrics are used to quantify the clustering performance. Results show the usefulness of proposed technique to track dynamic data, even being able to detect hidden objects.**

I. Introduction

Spectral clustering has taken an important place in pattern recognition due to its capability of accurately grouping data having complex structure. There are several spectral clustering approaches mainly related to graph partitioning [1]. The most suitable techniques are those based on kernels. Nevertheless, one of the biggest disadvantages of spectral clustering techniques is that most of them have been designed for static data analysis, that is to say, without taking into consideration the changes along the time. Some works have been developed taking into account the temporal information for the clustering task, mainly in segmentation of human motion [2], [3]. Other approaches include either the design of dynamic kernels for clustering [4], [5] or a dynamic kernel principal component analysis (KPCA) based model [6], [7]. Another study [8], [9] modifies the primal functional of a KPCA formulation for spectral clustering to add the memory effect.

Another approach, known as multiple kernel learning (MKL), has emerged to deal with different issues in machine learning, mainly, regarding support vector machines (SVM) [10], [11]. The intuitive idea of MKL is that learning can be enhanced when using different kernels instead of an unique kernel. Indeed, local analysis provided by each kernel is of benefit to examine the structure of the whole data.

From this idea, in this work, we introduce a dynamic kernel spectral clustering (DKSC) approach that is based on MKL. Our approach uses the so-called kernel spectral clustering (KSC), introduced in [12], which is based on KPCA formulation from least-square support vector machines and has shown to be a powerful tool for clustering hardly separable data allowing also out-of-samples extensions [13]. MKL is used in such a manner that kernel matrices are computed from an input data sequence, in which each data matrix represents a frame in a different time instance. Afterwards, an accumulative kernel is calculated as a linear combination of the previously obtained kernels where the weighting factors are obtained by ranking each sample contained in the frame. Such ranking is done by combining the relevance procedure proposed in [14] and the MKL approach presented in [15]. Experiments are carried out using two databases. On one hand, a subject from Graphics Lab Motion Capture Database from Carnegie Mellon University, here called Motion Caption. On the other hand, an artificial three-moving Gaussian clouds in which the mean of each cloud is changed along the frames. For comparison purposes, some conventional spectral clustering techniques are also considered, namely, kernel k-means (KKM) and min-cuts (MC) [1]. Also, standard k-means is considered. The normalized mutual information [16] and adjusted random index [17] metrics are used to quantify the clustering performance.

D. Peluffo-Ordóñez, S. García-Vega and G. Castellanos-Domínguez are with the Department of Electric Engineering, Electronics and Computer Science, Universidad Nacional de Colombia - Manizales (email: {dhpeluffoo, sgarciave, cgcastallanosd}@unal.edu.cp). R. Langone and J. A. K. Suykens are with the Katholieke Universiteit Leuven, Belgium. {rocco.langone, Johan.Suykens}@esat.kuleuven.be

II. Theoretical Background

A. Kernel Spectral Clustering

The kernel spectral clustering (KSC), introduced in [12], is aimed to split a data set of N samples into K homogeneous and disjoint subsets. Define a data matrix $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ in the form $\boldsymbol{X} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_N^\top]^\top$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is a sample vector. Data are first mapped and then projected. Assume a mapping function $\phi(\cdot)$, which maps data from the original dimension to a higher one d_h, so: $\phi(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}, \boldsymbol{x}_i \mapsto \phi(\boldsymbol{x}_i)$. Then, the mapping matrix $\boldsymbol{\Phi} = \left[\boldsymbol{\phi}(\boldsymbol{x}_1)^\top, \ldots, \boldsymbol{\phi}(\boldsymbol{x}_N)^\top\right]^\top$, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times d_h}$. The projections $\boldsymbol{E} \in \mathbb{R}^{N\times n_e}$ follow a latent variable model in the form:

$$\boldsymbol{E} = \boldsymbol{\Phi}\boldsymbol{W} + \mathbf{1}_N \otimes \boldsymbol{b}^\top, \tag{1}$$

Notation $\otimes$ the Kronecker product and term n_e denotes the number of considered support vectors. Then, according to [12] within a least-square-support vector machine framework, a matrix primal formulation of KSC can be stated as:

$$\min_{\boldsymbol{E},\boldsymbol{W},\boldsymbol{b}} \quad \frac{1}{2N}\,\mathrm{tr}(\boldsymbol{E}^\top \boldsymbol{V}\boldsymbol{E}\boldsymbol{\Gamma}) - \frac{1}{2}\,\mathrm{tr}(\boldsymbol{W}^\top\boldsymbol{W}) \tag{2a}$$

$$\text{s.t.} \quad \boldsymbol{E} = \boldsymbol{\Phi}\boldsymbol{W} + \mathbf{1}_N \otimes \boldsymbol{b}^\top \tag{2b}$$

where $\boldsymbol{\Gamma} = \mathrm{Diag}([\gamma_1, \ldots, \gamma_{n_e}])$ is a diagonal matrix formed by the regularization terms. Notations $\mathrm{tr}(\cdot)$ and $\otimes$ denote the trace and the Kronecker product, respectively.

To solve the KSC problem, we form the corresponding Lagrangian of previous problem, as follows:

$$\mathcal{L}(\boldsymbol{E},\boldsymbol{W},\boldsymbol{\Gamma},\boldsymbol{A}) = \frac{1}{2N}\ \mathrm{tr}(\boldsymbol{E}^\top \boldsymbol{V}\boldsymbol{E}) - \frac{1}{2}\ \mathrm{tr}(\boldsymbol{W}^\top\boldsymbol{W}) - \ \mathrm{tr}(\boldsymbol{A}^\top(\boldsymbol{E}-\boldsymbol{\Phi}\boldsymbol{W}-\mathbf{1}_N\otimes\boldsymbol{b}^\top))$$

where matrix $\boldsymbol{A}\in\mathbb{R}^{N\times n_e}$ contains the Lagrange multiplier vectors such that $\boldsymbol{A}=[\boldsymbol{\alpha}^{(1)},\cdots,\boldsymbol{\alpha}^{(n_e)}]$, and $\boldsymbol{\alpha}^{(l)}\in\mathbb{R}^N$ is the l-th vector of Lagrange multipliers.

Then, we determine the Karush-Kuhn-Tucker conditions by solving the partial derivatives on $\mathcal{L}(\boldsymbol{E},\boldsymbol{W},\boldsymbol{\Gamma},\boldsymbol{A})$. Afterwards, by eliminating the primal variables, the optimization problem posed in equation (2) is reduced to the following dual problem:

$$\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{V}\boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{A}, \tag{3}$$

where $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1,\ldots,\lambda_{n_e})$ is a diagonal matrix formed by the eigenvalues $\lambda_l = N/\gamma_l$, matrix $\boldsymbol{H}\in\mathbb{R}^{N\times N}$ is the centering matrix that is defined as $\boldsymbol{H} = \boldsymbol{I}_N - 1/(\mathbf{1}_N^\top\boldsymbol{V}\mathbf{1}_N)\mathbf{1}_N\mathbf{1}_N^\top\boldsymbol{V}$, being $\boldsymbol{I}_N$ a N-dimensional identity matrix. In addition, by applying the kernel trick in such a way that $\boldsymbol{\Omega}\in\mathbb{R}^{N\times N}$ be the kernel matrix $\boldsymbol{\Omega} = [\Omega_{ij}] = \mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x}_j), i,j\in[N]$, we have that $\boldsymbol{\Omega} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top$. Notation $\mathcal{K}(\cdot,\cdot):\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ stands for the kernel function. Notice that matrix $\boldsymbol{A}$ becomes the eigenvectors. As a result, the set of projections can be calculated as follows:

$$\boldsymbol{E} = \boldsymbol{\Omega}\boldsymbol{A} + \mathbf{1}_N\otimes\boldsymbol{b}^\top \tag{4}$$

Taking into account that the kernel matrix represents the similarity matrix of a graph with K connected subgraphs and assuming $\boldsymbol{V} = \boldsymbol{D}^{-1}$ being $\boldsymbol{D}\in\mathbb{R}^{N\times N}$ the degree matrix defined as $\boldsymbol{D} = \mathrm{Diag}(\boldsymbol{\Omega}\mathbf{1}_N)$; we can infer that the $K-1$ eigenvectors associated to the largest eigenvalues are cluster indicators [13]. Therefore, value n_e is fixed to be $K-1$. Afterwards, since each cluster is represented by a single coordinate in the $K-1$-dimensional eigenspace, we can encode the eigenvectors considering that two points are in the same cluster if they are in the same orthant in the corresponding eigenspace [13]. Therefore, by binaryzing the rows of the projection matrix $\boldsymbol{E}$, we obtain the code book as $\widetilde{\boldsymbol{E}} = \mathrm{sgn}(\boldsymbol{E})$, where $\mathrm{sgn}(\cdot)$ is the sign function. Thus, its corresponding rows are codewords, which allow to form the clusters according to the minimal Hamming distance.

B. Multiple Kernel Learning

Let us consider a sequence of N_{fm} input data matrices such that $\{\boldsymbol{X}^{(1)},\ldots,\boldsymbol{X}^{(N_{fm})}\}$, where $\boldsymbol{X}^{(t)} = [\boldsymbol{x}_1^{(t)\top},\ldots,\boldsymbol{x}_N^{(t)\top}]^\top$ is the data matrix associated to time instance t. In order to take into consideration the time effect within the computation of kernel matrix, we can apply a Multiple Kernel Learning approach, namely a linear combination of all the input data matrices until the current matrix. Then, at instance T, the accumulated kernel matrix can be computed as:

$$\widetilde{\boldsymbol{\Omega}}^{(T)} = \sum_{t=1}^{T}\eta_t\boldsymbol{\Omega}^{(t)} \tag{5}$$

where $\boldsymbol{\eta} = [\eta_1,\ldots,\eta_T]$ are the weighting coefficients or coefficients and $\boldsymbol{\Omega}^{(t)}$ is the kernel matrix associated to $\boldsymbol{X}^{(t)}$ such that $\Omega_{ij}^{(t)} = \mathcal{K}(\boldsymbol{x}_i^{(t)},\boldsymbol{x}_j^{(t)})$.

Regarding the weighting factor estimation, we take advantage of the relevance ranking introduced in [14], which is aimed to selecting a subset of features founded on spectral properties of the Laplacian of data matrix. This approach is based on a continuous ranking of the features by means of a least-squares maximization problem. Here, instead of using this approach for feature selection, we introduce a new formulation able of getting ranking values for the corresponding frames in the analyzed sequence. Also, it is worth mentioning that the proposed approach coheres to the clustering method. In this connection, the optimization problem formulation is as follows: Consider the frame matrix $\boldsymbol{\mathcal{X}}$ which is formed in such a way that each row is a frame by letting $\hat{\boldsymbol{x}}_t$ is the vectorization of coordinates representing the t-th frame. In other words, $\boldsymbol{\mathcal{X}} = [\hat{\boldsymbol{x}}_1^\top,\ldots,\hat{\boldsymbol{x}}_T]$ and $\hat{\boldsymbol{x}}_t = \mathrm{vec}(\boldsymbol{X}^{(t)})$. Also, consider its corresponding kernel matrix $\widehat{\boldsymbol{\Omega}}\in\mathbb{R}^{N_{fm}\times N_{fm}}$ such that $\widehat{\Omega}_{ij} = \mathcal{K}(\hat{\boldsymbol{x}}_i,\hat{\boldsymbol{x}}_j)$. By recalling equation (4), an energy maximization problem can be written as:

$$\max_{\boldsymbol{U}}\ \mathrm{tr}(\boldsymbol{U}^\top\widehat{\boldsymbol{\Omega}}^\top\widehat{\boldsymbol{\Omega}}\boldsymbol{U});\quad \text{s.t.}\quad \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I}_{n_e} \tag{6}$$

Note that previous statement comes from a liner projection of kernel matrix in the form $\boldsymbol{Z} = \boldsymbol{\Omega}\boldsymbol{U}$, where $\boldsymbol{U}$ is an orthonormal matrix in size $N_{fm}\times\hat{n}_e$ when considering $\hat{n}_e$ support vectors. According to the clustering method described in section II-A, we can infer that

$$\mathrm{tr}(\boldsymbol{U}^\top\widehat{\boldsymbol{\Omega}}^\top\widehat{\boldsymbol{\Omega}}\boldsymbol{U}) = \ \mathrm{tr}(\boldsymbol{Z}^\top\boldsymbol{Z}) = \ \mathrm{tr}(\widehat{\boldsymbol{\Lambda}}^2),$$

and therefore a feasible solution of the problem is $\boldsymbol{U} = \boldsymbol{A}$. Similarly as the MKL approach explained in [15], we introduce coefficient vector $\boldsymbol{\eta}\in\mathbb{R}^N$ as the solution of minimizing $\|\boldsymbol{Z}-\widetilde{\boldsymbol{Z}}\|_2^2$ subject to some orthogonality conditions, being $\widetilde{\boldsymbol{Z}}$ a lower-rank representation of $\boldsymbol{Z}$. The solution can be written as:

$$\boldsymbol{\eta} = \sum_{l=1}^{\hat{n}_e}\hat{\lambda}_l\hat{\boldsymbol{\alpha}}^{(l)}\circ\hat{\boldsymbol{\alpha}}^{(l)} \tag{7}$$

where $\circ$ denotes Hadamard (element-wise) product. Accordingly, the ranking factor η_i is a single value representing an unique frame in a sequence. Notation $\hat{a}$ means that variable a is related to $\widehat{\boldsymbol{\Omega}}$.

C. Dynamic KSC

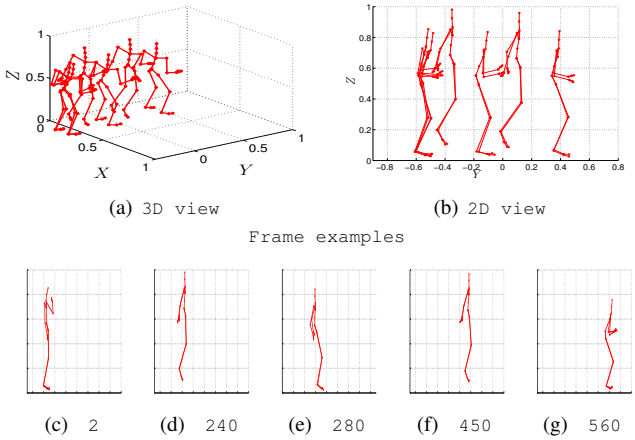
By combining MKL and KSC, we introduce a KSC for dynamic data, termed DKSC. This approach works as follows: Given a sequence of data matrices $\{\boldsymbol{X}^{(1)},\ldots,\boldsymbol{X}^{(N_{fm})}\}$ representing frames, being N_{fm} the number of frames, the corresponding kernel matrices are calculated $\{\boldsymbol{\Omega}^{(1)},\ldots,\boldsymbol{\Omega}^{(N_{fm})}\}$ with $\Omega_{ij}^{(t)} = \mathcal{K}(\boldsymbol{x}_i^{(t)},\boldsymbol{x}_j^{(t)})$. Then, the weighting factor or coefficient vector $\boldsymbol{\eta}$ is calculated by using (7) with the frame matrix $\boldsymbol{\mathcal{X}}$. Afterwards, MKL is applied by means of equation (5) to obtain the accumulated kernel matrices $\{\widetilde{\boldsymbol{\Omega}}^{(1)},\ldots,\widetilde{\boldsymbol{\Omega}}^{(N_{fm})}\}$. Finally, assuming a certain number of clusters K, KSC is applied

over each pair $(\boldsymbol{X}^{(t)}, \widetilde{\boldsymbol{\Omega}}^{(t)})$ with $t \in \{1, \ldots, N_{fm}\}$. Since accumulated kernel matrix is used, when clustering data at time instance T the information of the previous frame clustering, besides the current frame, is taken into account. Hence, this approach can be called as dynamic.

III. Experimental Set-Up

A. Databases

1) Motion caption: The data used in this work was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217. Such database is named Graphics Lab Motion Capture Database from Carnegie Mellon University. In this work, we use the trial number 1 (01_01.bvh), particularly, the *subject #1* (progressive jump). The two first jumps are considered, such that the first one is between frames 1 and 280, while the second one between 281 and 560. Each frame $\boldsymbol{X}^{(t)}$ per jump is in size 280×114 whose rows contain the vectorization of coordinates X, Y and Z of the subject's body points, therefore each $\hat{\boldsymbol{x}}_i$ is the dimension of 31920. Note that we consider two jumps which means $N_{fm} = 2$. Then, frame matrix $\boldsymbol{\mathcal{X}}$ is in size 2×31920.

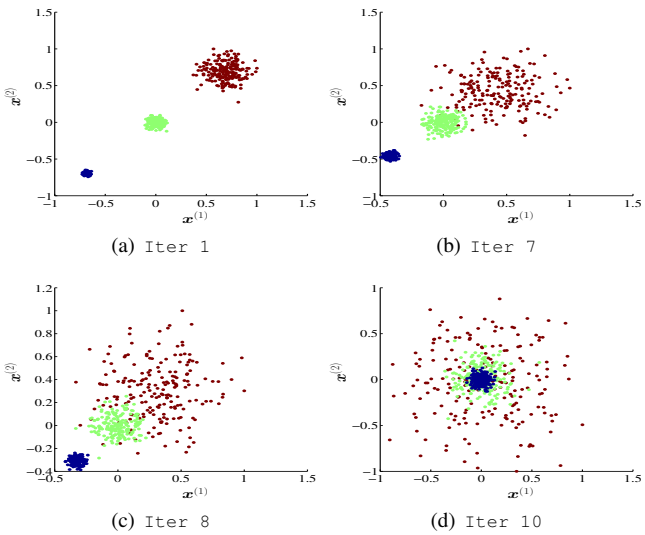


(a) 3D view (b) 2D view

(c) 2 (d) 240 (e) 280 (f) 450 (g) 560

Fig. 1. Motion Caption Database

2) Three-moving Gaussian clouds: An artificial three dimensional Gaussian data sequence is considered, which consists of Gaussian data with 3 clusters in such a way the deviation standard is static for all the frames and means are decreasing to move per frame each cluster towards each other. Namely, for a total of N_{fm} frames the mean and standard deviation vectors for t-th frame are respectively in the form $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3] = [-5 - t, 0, -5 - 0.5 * t]$ and $\boldsymbol{s} = [s_1, s_2, s_3] = [0.1, 0.3, 0.8]$, being μ_j and s_j the mean and standard deviation corresponding to the j-th cluster, respectively, with $j \in \{1, 2, 3\}$ and $r \in \{1, \ldots, N_{fm}\}$. The number of data samples per cluster is 200 and the considered total of frames is 10. Thus, each frame $\boldsymbol{X}^{(t)}$ is in size 600×3 which means that $\hat{\boldsymbol{x}}_i$ is of length 1800 as well as frame matrix $\boldsymbol{\mathcal{X}}$ in size 10×1800. In Fig. 5, some frames of moving Gaussian clouds are depicted.



(a) Iter 1 (b) Iter 7

(c) Iter 8 (d) Iter 10

Fig. 2. Three-moving Gaussian clouds

In addition, before starting the clustering process, data matrices from the both above databases are z-score normalized regarding their columns.

B. Clustering and Kernel Parameters

All the experiments are performed under specific initial parameters, namely, the number of clusters K per each frame and the kernel function. For Motion Caption database (*subject #1*), parameter K is set to be 3, in order to recognize three underlying movements. In the case of moving Gaussian clouds, we beforehand know that $K = 3$. The kernel matrices associated to the data sequence are calculated by the local-scaled Gaussian kernel [18]. Then, each entry of kernel matrix related to frame t is given by:

$$\Omega_{ij}^{(t)} = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{\sigma_i \sigma_j}\right), \qquad (8)$$

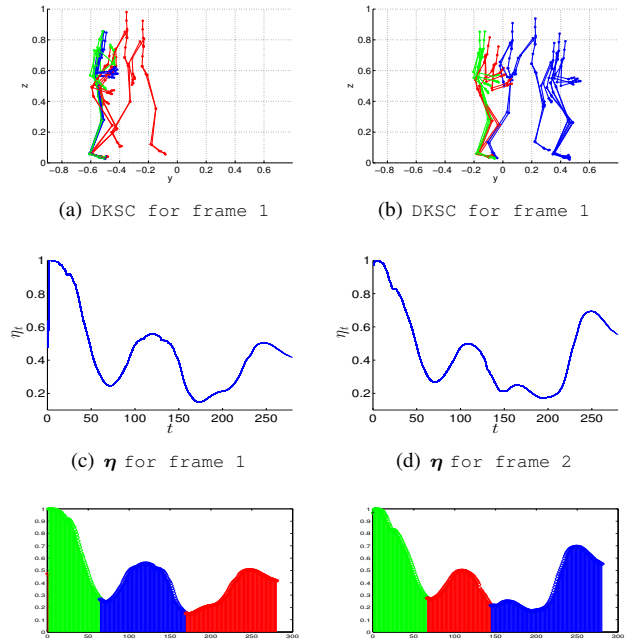
where $\|\cdot\|$ denotes the Euclidean norm and the scale parameter σ_i is chosen as the Euclidean distance between the sample $\boldsymbol{x}_i$ and its corresponding m-th nearest neighbour. Free parameter m is empirically set by varying it within an interval and then it is chosen as that one showing greatest Fisher's criterion value. In the case of Motion Caption, we obtain $m = 10$; while $m = 10$ for the moving Gaussian clouds. To compute $\widehat{\boldsymbol{\Omega}}$ is applied (8) as well. The clustering for the pair $(\boldsymbol{\mathcal{X}}, \widehat{\boldsymbol{\Omega}})$ is done by setting the number $K = N_{fm}$ and m as the entire number closest to $0.1N_{fm}$ for Motion Caption, and 1 for moving-Gaussian clouds.

For comparison purposes, kernel K-means (KKM) and min-cuts(MC) are also considered [1], which are applied over the data sequence by applying the same MKL approach as as that considered for KSC. The clustering performance is quantified by two metrics: normalized mutual information (NMI) [16] and adjusted random index (ARI) [17]. Both metrics return values ranged into the interval $[0, 1]$, being closer than 1 when better is the clustering performance.

IV. Results and Discussion

A. Results for Motion Caption Database

Motion caption database has not a ground truth to apply label-based metric to assess the clustering performance. However, because weighting factors $\boldsymbol{\eta}$ are ranking values related to samples, we can consider each instance (man position) as a sample. Then, KSC can be applied to generate the eigenvectors needed to compute $\boldsymbol{\eta}$. If analyzing each jump separately, corresponding $\boldsymbol{\eta}$ vectors should provide information about the clusters contained in the frame (jump). Fig. 3 shows the $\boldsymbol{\eta}$ vector corresponding to each jump.



(a) DKSC for frame 1 (b) DKSC for frame 1

(c) $\boldsymbol{\eta}$ for frame 1 (d) $\boldsymbol{\eta}$ for frame 2

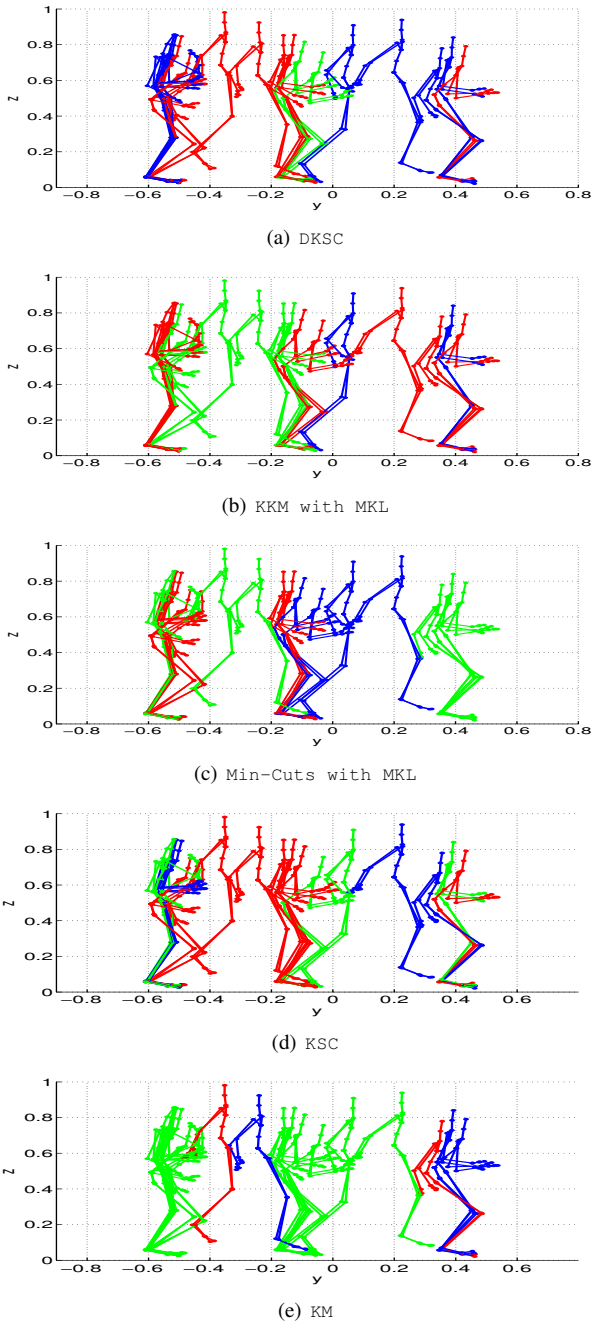
(e) Reference labels for $\boldsymbol{X}^{(1)}$ (f) Reference labels for $\boldsymbol{X}^{(2)}$

Fig. 3. MKL weighting factors for *Subject #1*

We can observe that $\boldsymbol{\eta}$ has a multi-modal shape. According to (7), $\boldsymbol{\eta}$ is computed from the eigenvectors $\boldsymbol{\alpha}^{(l)}$. Such eigenvectors point out the direction where samples have the most variability measured in term of a generalized inner product ($\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$). Then, we can argue that each mode might represent a different cluster. Under this assumption, we obtain the reference label vectors by detecting the local minima, considering each inflection as a cluster.

In Fig. 4, we can notice that for both jumps DKSC identifies three meaningful movements, namely: `starting/preparing the jump`, `on the air` and `arrival to ground`. In contrast, the remaining methods cluster either no contiguous instances what does not make sense since they are in a sequence, and incomplete underlying movements, i.e., incomplete jumps or static position split into two clusters.

Despite of that kernel k-means and min-cuts are applied within the proposed MKL framework, DKSC outperforms them. It can be also appreciated in Table I. Results are obtained by comparing the clustering indicators of each method



(a) DKSC

(b) KKM with MKL

(c) Min-Cuts with MKL

(d) KSC

(e) KM

Fig. 4. Clustering results for *Subject #1*

with the determined reference labels. Our approach reaches greater values than the other methods, then in terms of NMI and ARI it is possible to affirm that DKSC is a suitable approach to cluster frames in this kind of applications.

B. Results for Three-moving Gaussian clouds

In Fig 5, we can appreciate 4 selected frames from the total of 10 representing the three-moving Gaussian clouds. In particular, we select 1, 7, 8 and 10 since they show significant changes in the performance of considered clustering methods. We can appreciate when Gaussian clouds are relatively far to each other, all considered clustering

Measure	Frame	Clustering Method			
		DKSC	KKM	KM	MC
NMI	1	0.9181	0.8427	0.4736	0.7065
	2	0.7624	0.7202	0.6009	0.4102
ARI	1	0.9448	0.8761	0.3777	0.6239
	2	0.7000	0.6762	0.4991	0.2755

TABLE I

NMI AND ARI FOR *Subject # 1* CLUSTERING PERFORMANCE

methods work well. In contrast, when they are closer – showing overlapping– the best performance is achieved by DKSC. K-means, since it is a center-based approach, is not able to identify the clusters properly. Even, kernel k-means, despite of the use of MKL, no performs a right clustering in all cases. For instance, note that in frame 7 and 8 clusters are mixed. This can be attributed to the random initial centers selected to start the algorithm. The NMI and ARI values for the clustering performance are shown in Table II. Again, we can appreciate that proposed method outperforms kernel k-means despite the dynamic scheme as well as well as the standard k-means.

Measure	*Iteration*	Clustering Method		
		DKSC	KKM	KM
NMI	1	1	0.9549	1
	2	1	1	1
	3	1	1	1
	4	1	1	0.9904
	5	1	1	0.9763
	6	1	0.6570	0.9488
	7	1	1	0.9069
	8	1	1	0.8024
	9	1	0.6507	0.2864
	10	1	0.6637	0.2359
ARI	1	1	0.9703	1
	2	1	1	1
	3	1	1	1
	4	1	1	0.9950
	5	1	1	0.9851
	6	1	0.4535	0.9607
	7	1	1	0.9139
	8	1	1	0.7677
	9	1	0.4582	0.2096
	10	1	0.4664	0.1509

TABLE II

CLUSTERING PERFORMANCE PER FRAME FOR THREE-MOVING GAUSSIAN CLOUDS DATABASE

Then, our approach is an alternative to manage applications involving both hidden objects and dynamic data.

V. CONCLUSIONS

This work introduces an approach to tracking time varying data by means of spectral clustering within a multiple kernel framework. We proved that a linear combination of kernels is an alternative to cluster dynamic data taking into account past information, where coefficients or weighting factors can be obtained from an eigenvector-based problem. Also, we verified that there exists a direct relationship between the weighting factors and the supposed ground truth.

As a future work, we are aiming to exploit more spectral properties and techniques, mainly, those ones based on multiple kernel learning to design clustering approaches able to deal with dynamic data.

VI. ACKNOWLEDGMENTS

This research is supported by Research Council KUL: GOA/10/09 MaNet , PFV/10/002 (OPTEC), several PhD/postdoc & fellow grants; Flemish Government: IOF: IOF/KP/SCORES4CHEM. FWO: PhD/postdoc grants, projects: G.0588.09 (Brain-machine), G.0377.09 (Mechatronics MPC), G.0377.12 (Structured systems). IWT: PhD Grants, projects: SBO LeCoPro, SBO Climaqs, SBO POM, EUROSTARS SMART. iMinds 2013. Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017). EU: FP7-EMBOCON (ICT-248940), FP7-SADCO (MC ITN-264735), ERC ST HIGHWIND (259 166), ERC AdG A-DATADRIVE-B. COST: Action ICO806: IntelliCIS. Johan Suykens is a professor at the Katholieke Universiteit Leuven, Belgium. The scientific responsibility is assumed by its authors.

As well as by *"Jóvenes Investigadores"* COLCIENCIAS program with the project entitled *"Comparativo de métodos kernel para agrupamiento espectral de datos desde un enfoque primal-dual"*.

REFERENCES

[1] C. Guo, S. Zheng, Y. Xie, and W. Hao, "A survey on spectral clustering," in *World Automation Congress (WAC), 2012*. IEEE, 2012, pp. 53–56.

[2] B. Takács, S. Butler, and Y. Demiris, "Multi-agent behaviour segmentation via spectral clustering," in *Proceedings of the AAAI-2007, PAIR Workshop*, pp. 74–81.

[3] F. Zhou, F. Torre, and J. Hodgins, "Aligned cluster analysis for temporal segmentation of human motion," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*. IEEE, 2008, pp. 1–7.

[4] A. Chan and N. Vasconcelos, "Probabilistic kernels for the classification of auto-regressive visual processes," 2005.

[5] J. Keshet and S. Bengio, *Automatic speech and speaker recognition: Large margin and kernel methods*. Wiley, 2009.

[6] M. Maestri, M. Cassanello, and G. Horowitz, "Kernel pca performance in processes with multiple operation modes," *Chemical Product and Process Modeling*, vol. 4, no. 5, p. 7, 2009.

[7] S. Choi and I. Lee, "Nonlinear dynamic process monitoring based on dynamic kernel pca," *Chemical engineering science*, vol. 59, no. 24, pp. 5897–5908, 2004.

[8] R. Langone, C. Alzate, and J. A. K. Suykens, "Kernel spectral clustering with memory effect," *Physica A: Statistical Mechanics and its Applications*, vol. 392, no. 10, pp. 2588–2606, 2013.

[9] R. Langone and J. A. K. Suykens, "Community detection using kernel spectral clustering with memory," *Journal of Physics: Conference Series*, vol. 410, no. 1, p. 012100, 2012.

[10] F. González, D. Bermeo, L. Ramos, and O. Nasraoui, "On the robustness of kernel-based clustering," *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 122–129, 2012.

[11] H. Huang, Y. Chuang, and C. Chen, "Multiple kernel fuzzy clustering," *Fuzzy Systems, IEEE Transactions on*, vol. 20, no. 1, pp. 120–134, 2012.

[12] C. Alzate and J. Suykens, "A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods," in *Neural Networks, 2006. IJCNN'06. International Joint Conference on*. IEEE, 2006, pp. 138–144.

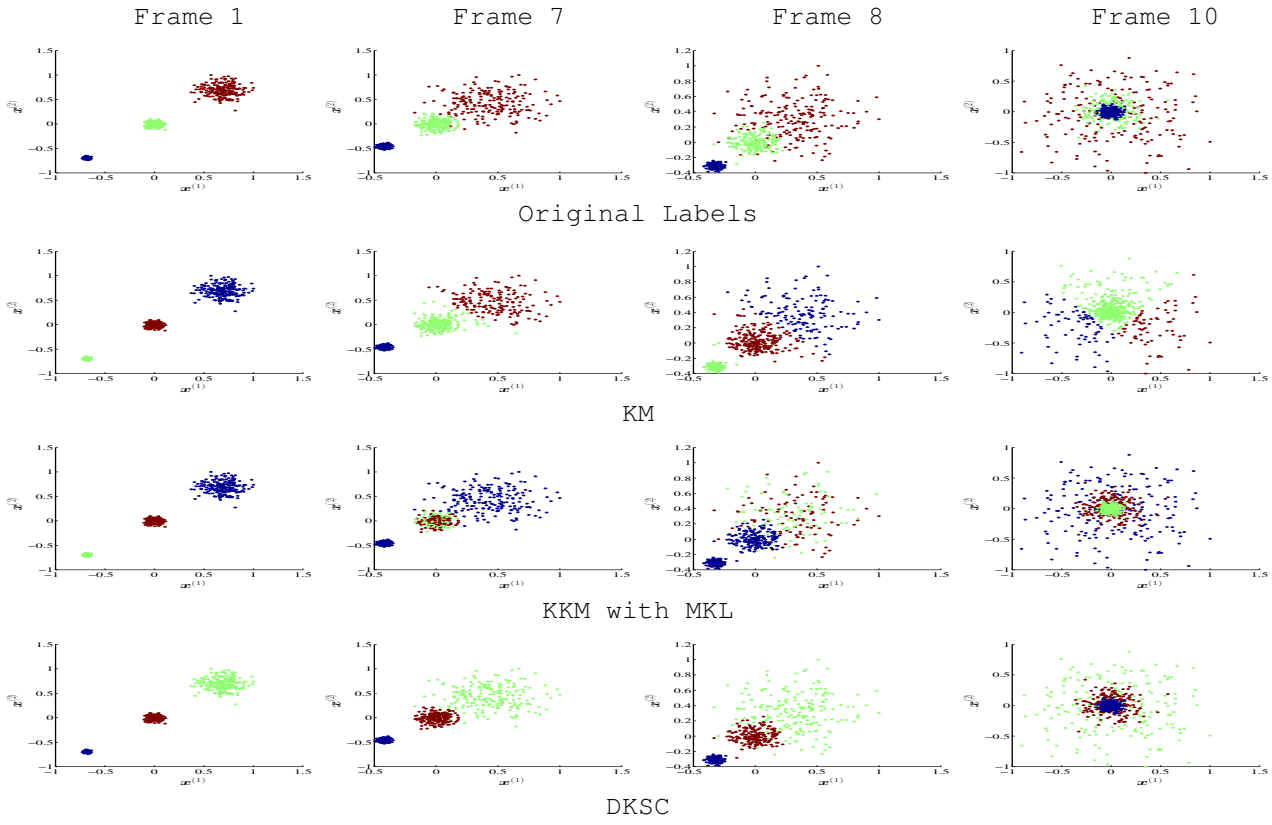


Fig. 5. Clustering performance for three-moving Gaussian clouds

[13] C. Alzate and J. A. K. Suykens, "Multiway spectral clustering with out-of-sample extensions through weighted kernel pca," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 2, pp. 335–347, 2010.

[14] L. Wolf and A. Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *J. Mach. Learn. Res.*, vol. 6, pp. 1855–1887, December 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1046920.1194906

[15] S. Molina-Giraldo, A. Álvarez-Meza, D. Peluffo-Ordoñez, and G. Castellanos-Domínguez, "Image segmentation based on multi-kernel learning and feature relevance analysis," *Advances in Artificial Intelligence–IBERAMIA 2012*, pp. 501–510, 2012.

[16] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, vol. 3, pp. 583–617, 2002.

[17] L. Hubert and P. Arabie, "Comparing partitions," *Journal of Classification*, vol. 1, no. 2, pp. 193–218, 1985.

[18] L. Zelnik-manor and P. Perona, "Self-tuning spectral clustering," in *Advances in Neural Information Processing Systems 17*. MIT Press, 2004, pp. 1601–1608.