

Kernel Spectral Clustering for Motion Tracking: A First Approach

D. Peluffo-Ordóñez*, S. García-Vega, and C.G. Castellanos-Domínguez

Universidad Nacional de Colombia – Manizales,
Km 7 vía al Magdalena
{dhpeluffoo,segarciave,cgcastellanosd}@unal.edu.com
<http://www.fia.unal.edu.co/gta/signal/index.php>

Abstract. This work introduces a first approach to track moving-samples or frames matching each sample to a single meaningful value. This is done by combining the kernel spectral clustering with a feature relevance procedure that is extended to rank the frames in order to track the dynamic behavior along a frame sequence. We pose an optimization problem to determine the tracking vector, which is solved by the eigenvectors given by the clustering method. Unsupervised approaches are preferred since, for motion tracking applications, labeling is unavailable in practice. For experiments, two databases are considered: Motion Caption and an artificial three-moving Gaussian in which the mean changes per frame. Proposed clustering is compared with kernel K-means and Min-Cuts by using normalized mutual information and adjusted random index as metrics. Results are promising showing clearly that there exists a direct relationship between the proposed tracking vector and the dynamic behavior.

Keywords: Kernels, motion tracking, spectral clustering.

1 Introduction

Nowadays, dynamic or time-varying data analysis is of great interest for the scientific community, specially in automation and pattern recognition. Video analysis [1] and motion identification [2] are some of the applications. Spectral matrix analysis is one of the approaches to address this issue. Spectral techniques, particularly within the graph-cut framework, has been a suitable tool in several aspects of interest in pattern recognition and machine learning even when data are time-varying, such as the estimation of the number of clusters [3], feature extraction and selection [4] as well as clustering [5] and classification [2]. Regarding spectral clustering, in [6] it is introduced the so-called kernel spectral clustering (KSC), which is based on least-square support vector machines and has shown to be a powerful tool for clustering hardly separable data allowing

* This work is supported by “*Jóvenes Investigadores*” COLCIENCIAS program with the project entitled “*Comparativo de métodos kernel para agrupamiento espectral de datos desde un enfoque primal-dual*”.

also extensions to out-of-samples [7]. In another work [4], the authors present a definition of feature relevance 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. Such maximization is done over a functional such that sparse solutions for the ranking values are obtained by a spectral decomposition of non-negative matrix [4]. What makes this approach interesting within the context that we are concerned in this work is the possibility to get a ranking value for each frame in the process of time-varying data analysis. In addition, the feature relevance approach measures the relevance of a subset of features against its influence on the cluster assignment.

In this work, from the KSC combined with the relevance ranking proposed in [4], we introduce a novel approach to track moving samples or frames matching each sample to an unique meaningful value. Notice that we prefer the unsupervised approaches since, in practice, labeling is not available for motion tracking applications. Our approach works as follows: it starts by clustering the input data by means of KSC with a manually established of groups and a determined kernel function. Then, we take the kernel matrix as another data representation that can be linearly projected in order to apply the relevance ranking over the samples instead of the features as proposed in [4]. The projection matrix is obtained as a sparse solution of a quadratic optimization problem, where a projection energy (also called variance) term is maximized. It is proved that projection matrix is the same as the eigenvector matrix given by KSC. Finally, a tracking vector is obtained by a linear combination of such eigenvectors similarly as described in [8]. Experiments are carried out using two databases: Motion Caption and an artificial three-moving Gaussian in which the mean is changed throughout the frames. Proposed KSC is compared with kernel K-means and Min-Cuts [9]. To quantify the clustering performance, normalized mutual information [10] and adjusted random index [11] metrics are used. Obtained results are promising showing clearly that there exists a direct relationship between the proposed tracking vector and the analyzed dynamic data.

2 Theoretical Background

2.1 Kernel Spectral Clustering

The aim of kernel spectral clustering (KSC), introduced in [6], is to divide a data set of N samples into K homogeneous and disjoint subsets. Consider that the data are arranged in the data matrix $\mathbf{X} \in \mathbb{R}^{N \times d}$ assuming that each sample vector \mathbf{x}_i is d -dimensional, then $\mathbf{X} = [\mathbf{x}_1^\top, \dots, \mathbf{x}_N^\top]^\top$. Assume a latent variable model for the projections $\mathbf{E} \in \mathbb{R}^{N \times n_e}$ as $\mathbf{E} = \mathbf{\Phi}\mathbf{W} + \mathbf{1}_N \otimes \mathbf{b}^\top$, being matrix $\mathbf{\Phi} = [\phi(\mathbf{x}_1)^\top, \dots, \phi(\mathbf{x}_N)^\top]^\top$, $\mathbf{\Phi} \in \mathbb{R}^{N \times d_h}$, is a high dimensional representation of data such that $\phi(\cdot)$ maps data from the original dimension to a higher one d_h , i.e., $\phi(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{d_h}$, $\mathbf{W} = [\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(n_e)}]$ is the weighting factor matrix, $\mathbf{W} \in \mathbb{R}^{d_h \times n_e}$, and $\mathbf{b} = [b_1, \dots, b_{n_e}]$ is a vector containing bias terms, $\mathbf{b} \in \mathbb{R}^{n_e}$. Notation

\otimes the Kronecker product and term n_e denotes the number of considered support vectors. Then, according to [6] within a least-square-support vector machine framework, a matrix primal formulation of KSC can be stated as:

$$\min_{\mathbf{E}, \mathbf{W}, \mathbf{b}} \frac{1}{2N} \text{tr}(\mathbf{E}^\top \mathbf{V} \mathbf{E} \mathbf{\Gamma}) - \frac{1}{2} \text{tr}(\mathbf{W}^\top \mathbf{W}); \quad \text{s.t.} \quad \mathbf{E} = \mathbf{\Phi} \mathbf{W} + \mathbf{1}_N \otimes \mathbf{b}^\top \quad (1)$$

where $\mathbf{\Gamma} = \text{Diag}([\gamma_1, \dots, \gamma_{n_e}])$ is a diagonal matrix formed by the regularization terms. Notations $\text{tr}(\cdot)$ 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:

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

where matrix $\mathbf{A} \in \mathbb{R}^{N \times n_e}$ contains the Lagrange multiplier vectors such that $\mathbf{A} = [\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(n_e)}]$, 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 $\mathcal{L}(\mathbf{E}, \mathbf{W}, \mathbf{\Gamma}, \mathbf{A})$. Afterwards, by eliminating the primal variables, the optimization problem posed in equation (1) is reduced to the following dual problem: $\mathbf{A} \mathbf{\Lambda} = \mathbf{V} \mathbf{H} \mathbf{\Phi} \mathbf{\Phi}^\top \mathbf{A}$, where $\mathbf{\Lambda} = \text{Diag}(\lambda_1, \dots, \lambda_{n_e})$ is a diagonal matrix formed by the eigenvalues $\lambda_l = N/\gamma_l$, matrix $\mathbf{H} \in \mathbb{R}^{N \times N}$ is the centering matrix that is defined as $\mathbf{H} = \mathbf{I}_N - 1/(\mathbf{1}_N^\top \mathbf{V} \mathbf{1}_N) \mathbf{1}_N \mathbf{1}_N^\top \mathbf{V}$, being \mathbf{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 $\boldsymbol{\Omega} = [\Omega_{ij}] = \mathcal{K}(\mathbf{x}_i, \mathbf{x}_j)$, $i, j \in [N]$, we have that $\boldsymbol{\Omega} = \mathbf{\Phi} \mathbf{\Phi}^\top$. Notation $\mathcal{K}(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ stands for the kernel function. Notice that matrix \mathbf{A} becomes the eigenvectors. As a result, the set of projections can be calculated as follows:

$$\mathbf{E} = \boldsymbol{\Omega} \mathbf{A} + \mathbf{1}_N \otimes \mathbf{b}^\top \quad (2)$$

Taking into account that the kernel matrix represents the similarity matrix of a graph with K connected subgraphs and assuming $\mathbf{V} = \mathbf{D}^{-1}$ being $\mathbf{D} \in \mathbb{R}^{N \times N}$ the degree matrix defined as $\mathbf{D} = \text{Diag}(\boldsymbol{\Omega} \mathbf{1}_N)$; we can infer that the $K - 1$ eigenvectors associated to the largest eigenvalues are cluster indicators [7]. 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 [7]. Therefore, by binaryzing the rows of the projection matrix \mathbf{E} , we obtain the code book as $\tilde{\mathbf{E}} = \text{sgn}(\mathbf{E})$, where $\text{sgn}(\cdot)$ is the sign function. Thus, its corresponding rows are codewords, which allow to form the clusters according to the minimal Hamming distance.

2.2 Tracking by KSC

Similarly as the relevance analysis explained in [4] in which a functional regarding a non-negative matrix is introduced, we pose an optimization problem with the

difference that we are concerned of obtaining the ranking values for samples instead of features, as follows: Focusing on the task of interest, consider that the non-negative matrix is chosen as Ω and the data matrix \mathbf{X} is formed in such a way that each row is a frame, i.e., \mathbf{x}_i is the vectorization of coordinates representing the i -th frame. By recalling equation (2), an energy maximization problem can be written as:

$$\max_U \text{tr}(\mathbf{U}^\top \Omega^\top \Omega \mathbf{U}); \quad \text{s.t.} \quad \mathbf{U}^\top \mathbf{U} = \mathbf{I}_{n_e} \quad (3)$$

The orthonormal rotation matrix is \mathbf{U} is in size $N \times n_e$, such that the linear transformation $\mathbf{Z} \in \mathbb{R}^{N \times n_e}$ of kernel matrix is in the form $\mathbf{Z} = \Omega \mathbf{U}$. Considering the procedure described in section 2.1, we can notice that $\text{tr}(\mathbf{U}^\top \Omega^\top \Omega \mathbf{U}) = \text{tr}(\mathbf{A}^2)$ and therefore a feasible solution of the problem is $\mathbf{U} = \mathbf{A}$. Now, as explained in [8], a ranking vector $\boldsymbol{\eta} \in \mathbb{R}^N$ can be expressed as a linear combination of vectors $\boldsymbol{\alpha}^{(l)}$ so:

$$\boldsymbol{\eta} = \sum_{l=1}^{n_e} \lambda_l |\boldsymbol{\alpha}^{(l)}| \quad (4)$$

where $|\cdot|$ denotes absolute value. Accordingly, the ranking factor η_i is a single value representing an unique frame in a sequence. In this sense, $\boldsymbol{\eta}$ can be called tracking vector.

3 Experimental Set-Up

3.1 Databases

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. In this work, we use the trial number 4 (02_04), particularly, the subject #2 (jump, balance). Each frame is in size 38×3 representing the coordinates X , Y and Z of the subject's body points, therefore each \mathbf{x}_i is 114-dimensional. Since 484 frames are considered, data matrix is in size 484×114 .*

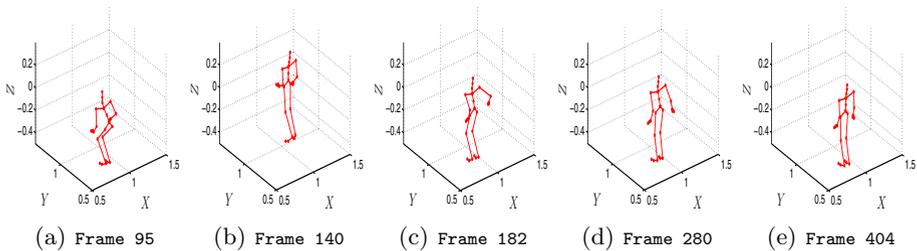


Fig. 1. Motion Caption Database

Three-Moving Gaussian. An artificial three dimensional Gaussian data sequence is also considered, which consist 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 at r -th frame are respectively in the form $\boldsymbol{\mu} = [\mu_1, \mu_2, \mu_3] = [-20(1 - r/N_{fm}), 0, -20(1 - r/N_{fm})]$ and $\boldsymbol{s} = [s_1, s_2, s_3] = [1, 2, 3]$, being μ_j and s_j the mean and standard deviation corresponding to the j -th cluster, respectively; as well as $j \in \{1, 2, 3\}$ and $r \in \{1, \dots, N_{fm}\}$. The number of data samples per cluster is 300 and the considered total of frames is 100. Thus, each frame is in size 900×3 which means that \boldsymbol{x}_i is of length 2700 as well as data matrix is $\boldsymbol{X} \in \mathbb{R}^{100 \times 2700}$. In Fig. 2, some frames of moving Gaussian are depicted.

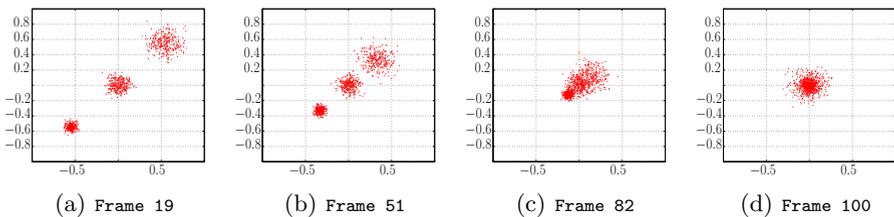


Fig. 2. Three Gaussian Database

Data matrices from the both above databases are z -score normalized regarding their columns before starting the clustering process.

3.2 Clustering and Kernel Parameters

All the experiments are performed under specific initial parameters, namely, the number of clusters K and kernel function. For Motion Caption database (*subject #2*), parameter K is set to be 5. In case of moving Gaussian, we empirically determine that $K = 4$. The considered kernel for both cases is the local-scaled Gaussian kernel [12] defined as: $\Omega_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2 / (\sigma_i \sigma_j))$, 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 neighbor. Free parameter m is empirically set by varying it within an interval and then it is chosen as that one achieving greatest Fisher's criterion value. In the case of Motion Caption, we obtain $m = 35$; while $m = 10$ for the moving Gaussian.

For comparison purposes, kernel K-means (KKM) and Min-Cuts are also considered [9]. The clustering performance is quantified by two metrics: normalized mutual information (NMI) [10] and adjusted random index (ARI) [11]. Both metrics return values ranged into the interval $[0, 1]$, being closer than 1 when better is the clustering performance.

4 Results

4.1 Results for Motion Caption Database

After applying KSC over matrix \mathbf{X} , vector $\boldsymbol{\eta}$ is calculated using (4). Fig. 3 is the dynamic analysis achieved by KSC. The tracking vector plotting is shown in Fig. 3(a). As it can be appreciated, $\boldsymbol{\eta}$ has a multi-modal shape. Since the eigenvectors $\boldsymbol{\alpha}^{(l)}$ point out the direction where samples have the most variability measured in term of a generalized inner product ($\boldsymbol{\Phi}^\top \boldsymbol{\Phi}$), we can infer that each mode might represent a different cluster. In Fig. 3(b), the reference labeling vector is shown which is obtained by detecting the local minima through a simple search. Such vector is henceforth considered as the reference labels to quantify the clustering performance by NMI and ARI.

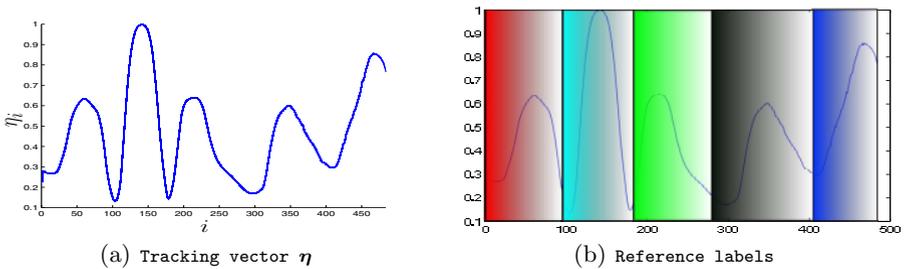


Fig. 3. Dynamic analysis of *Subject # 2* by KSC

In Fig. 4, we can observe the dynamic behavior tracking done for all considered methods. We match the tracking vector with cluster indicators yielded by each method. It is easy to appreciate that KSC performs a better clustering in comparison to the other methods. This fact can be attributed to the coherence and evident linkage between the tracking vector and KSC. Also, note that the samples are organized in sequence, then we expect resultant clusters contain no isolated or disconnected (out of sequence) samples. In other words, clustering indicators reach low values when samples or frames are intermittently clustered along time. Then, KKM and Min-Cuts generate partially wrong grouping. Nevertheless, it is worth mentioning that this experimental framework was done with a specific type of kernel as well as a determined parameter tuning, i.e., maximal Fisher's criterion in this case. Thus, it is possible that such a tuning process may not be appropriate for those methods. Albeit, it must be taken into account that Fisher's criterion is applied regardless the method, that is to say that tuning process is done considering no the clustering method but the nature of data by means of the within- and between-variance ratio (Fisher's criterion). The trial 4 (02_04) is a vertical jump as shown. Then, by comparing Fig. 1 with Fig. 3, we can appreciate that the identified local minima determine five clusters representing five dynamic instances along time, associated to subject #2.

We name those clusters as follows: *starting jump* (between frames 1 and 95), *jumping on the air* (between frames 96 and 182), *arrival to the ground* (between frames 183 and 280), *back step* (between 281 and 404) and *quiet - Standing* (between frames 405 and 484).

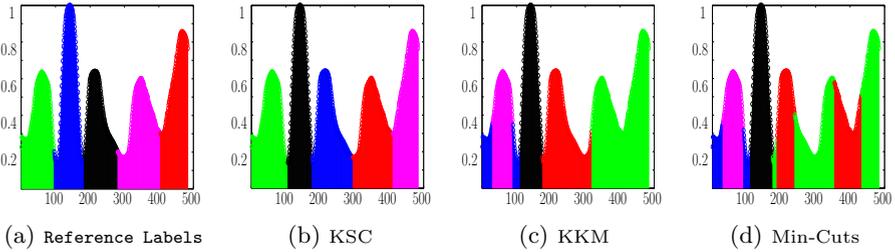


Fig. 4. Clustering results for *Subject #2*

The direct relationship between tracking vector η and the partition of natural movements from *Subject #2* can be plainly appreciated in Fig. 5, where the top row shows one representative frame per cluster (in different color) while bottom row depicts the η curve until the last frame of the corresponding cluster.

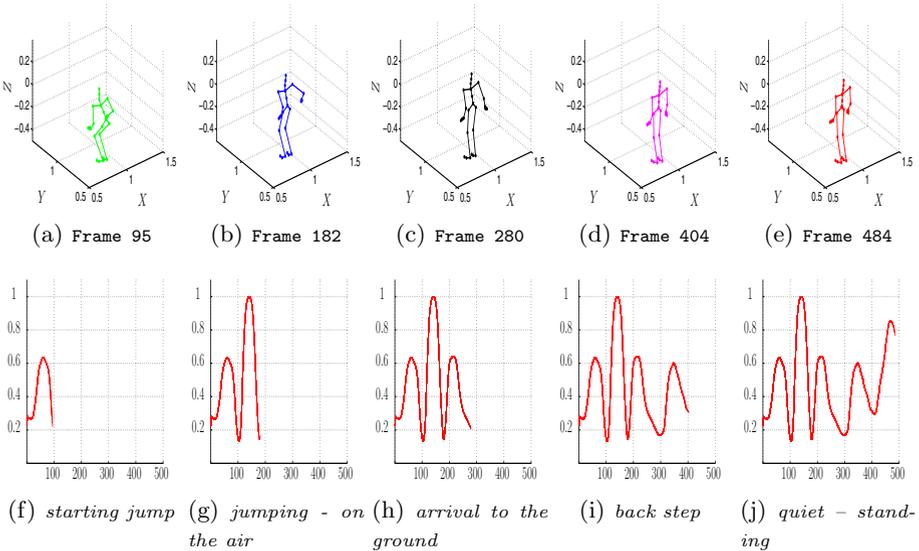


Fig. 5. *Subject #2* tracking

As expected, each mode – frames between inflections forming a concave curve – from η plotting corresponds to a natural cluster, which may even be determined by simple inspection. The tracking effect can also be verified in Table 1. In terms of the considered measures, we can observe that KSC outperforms both KKM and Min-Cuts, which means that the clustering method might be coherent with the tracking procedure. In this connection, KSC is the most suitable since η comes from the eigenvectors.

Table 1. Clustering performance for *Subject # 2* in terms of considered metrics

| Measure | Clustering Method | | |
|---------|-------------------|--------|----------|
| | KSC | KKM | Min Cuts |
| NMI | 0.8410 | 0.6982 | 0.6974 |
| ARI | 0.8163 | 0.5752 | 0.5400 |

4.2 Results for Three-Moving Gaussian Database

Similarly to the previous database, now we explore the dynamic behavior when Gaussian data are moving towards each other. Fig. 6 depicts the tracking vectors when varying the number of clusters from 2 to 6 on top, and the corresponding reference labeling vectors at bottom.

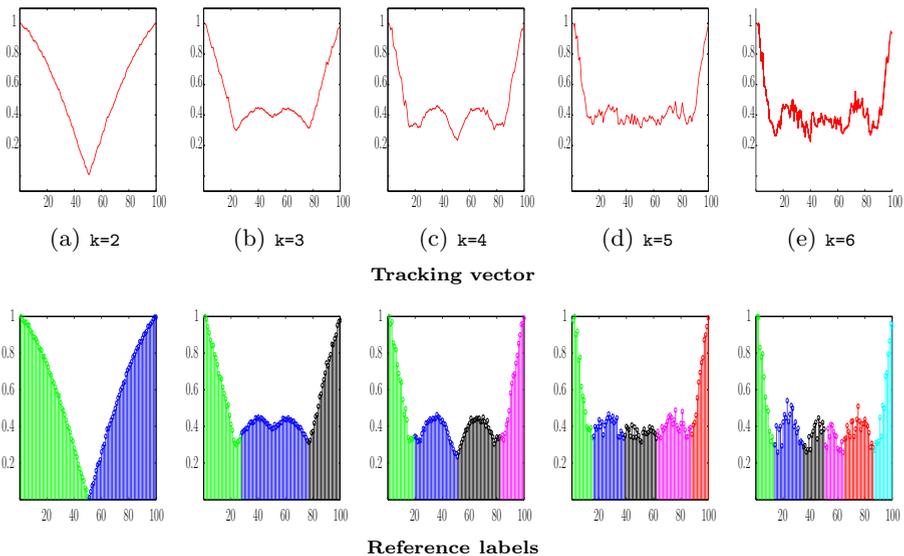
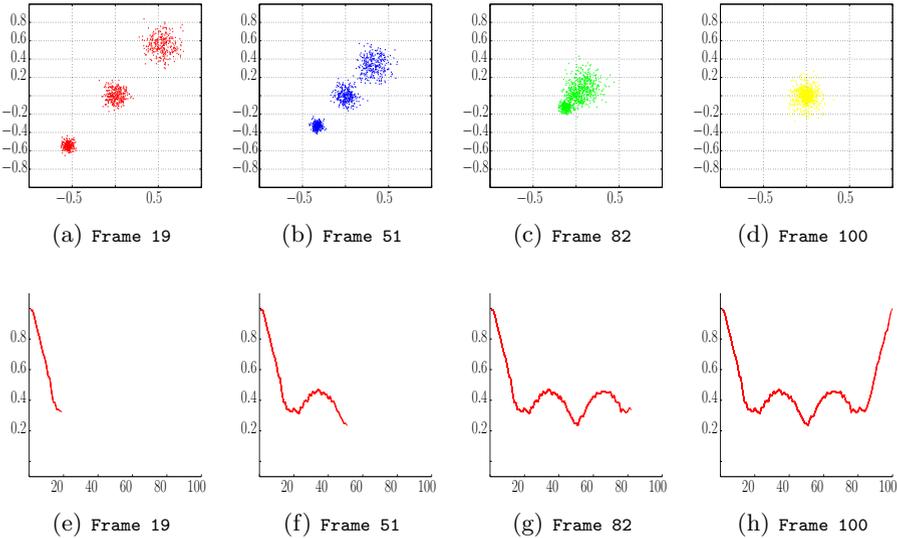


Fig. 6. Tracking Vectors for three-Gaussian moving

Table 2. Effect of number of groups over the *Subject #2*

| Measure | K | Clustering Method | | |
|---------|-----|-------------------|--------|----------|
| | | KSC | KKM | Min-Cuts |
| NMI | 2 | 1 | 0.9291 | 1 |
| | 3 | 0.5817 | 0.8068 | 0.7257 |
| | 4 | 0.8839 | 0.8306 | 0.6408 |
| | 5 | 0.7426 | 0.9470 | 0.5966 |
| | 6 | 0.6998 | 0.7803 | 0.5900 |
| ARI | 2 | 1 | 0.9600 | 1 |
| | 3 | 0.4017 | 0.8135 | 0.7046 |
| | 4 | 0.8924 | 0.8227 | 0.5037 |
| | 5 | 0.5840 | 0.9416 | 0.4148 |
| | 6 | 0.5281 | 0.6761 | 0.3393 |

**Fig. 7.** Three-moving Gaussian regarding tracking-generated reference labels

As it can be noted, either $K = 2$ or $K = 4$ seems to be the proper number of clusters because in those modes are well formed and then easily identifiable. Then, our approach can also be used to determine the number of groups in a sequence of frames. Indeed, in Table 2 the best performance is reached when clustering with those values of K .

For instance in Fig. 7, the tracking effect with $K = 4$ is shown. Again, each mode represents a single cluster.

5 Conclusions

This work presents a novel approach for motion tracking from a sequence of moving samples or frames, which takes advantage of spectral properties from a kernel-based clustering as well as a relevance ranking procedure. Tracking is done by finding an unique value representing adequately each single frame. Our approach determines a tracking vector that has a direct relationship with the underlying dynamic behavior of analyzed sequence, allowing even to estimate the number of groups as well as the ground truth. As a future work, with the aim of designing an accurate tracking method able to be extended to prediction problems, new spectral properties and techniques are to be explored.

References

1. Shirazi, S., Harandi, M.T., Sanderson, C., Alavi, A., Lovell, B.C.: Clustering on grassmann manifolds via kernel embedding with application to action analysis. In: Proc. IEEE International Conference on Image Processing (2012)
2. Sudha, L., Bhavani, R.: Performance comparison of svm and knn in automatic classification of human gait patterns. *Int. J. Comput* 6(1), 19–28 (2012)
3. Lee, S., Hayes, M.: Properties of the singular value decomposition for efficient data clustering. *IEEE, Signal Processing Letters* 11(11), 862–866 (2004)
4. Wolf, L., Shashua, A.: Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach. *J. Mach. Learn. Res.* 6, 1855–1887 (2005), <http://portal.acm.org/citation.cfm?id=1046920.1194906>
5. Alzate, C., Suykens, J.: Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 335–347 (2008)
6. Alzate, C., Suykens, J.: A weighted kernel PCA formulation with out-of-sample extensions for spectral clustering methods. In: International Joint Conference on Neural Networks, IJCNN 2006, pp. 138–144. IEEE (2006)
7. Alzate, C., Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted kernel pca. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32(2), 335–347 (2010)
8. Molina-Giraldo, S., Álvarez-Meza, A.M., Peluffo-Ordoñez, D.H., Castellanos-Domínguez, G.: Image segmentation based on multi-kernel learning and feature relevance analysis. In: Pavón, J., Duque-Méndez, N.D., Fuentes-Fernández, R. (eds.) *IBERAMIA 2012. LNCS*, vol. 7637, pp. 501–510. Springer, Heidelberg (2012)
9. Guo, C., Zheng, S., Xie, Y., Hao, W.: A survey on spectral clustering. In: *World Automation Congress (WAC)*, pp. 53–56. IEEE (2012)
10. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, 583–617 (2002)
11. Hubert, L., Arabie, P.: Comparing partitions. *Journal of Classification* 1(2), 193–218 (1985)
12. Zelnik-manor, L., Perona, P.: Self-tuning spectral clustering. In: *Advances in Neural Information Processing Systems*, vol. 17, pp. 1601–1608. MIT Press (2004)