Kernel-Spectral-Clustering-Driven Motion Segmentation: Rotating-Objects First Trials

Chapter · November 2019

DOI: 10.1007/978-3-030-36636-0_3

Kernel-Spectral-Clustering-Driven Motion Segmentation: Rotating-Objects First Trials

O. Oña-Rocha[1,2], J. A. Riascos-Salas[3,7](✉), I. C. Marrufo-Rodríguez[4], M. A. Páez-Jaime[4], D. Mayorca-Torres[5], K. L. Ponce-Guevara[6], J. A. Salazar-Castro[7], and D. H. Peluffo-Ordóñez[1,4]

[1] Universidad Técnica del Norte, Ibarra, Ecuador
[2] Universidad de las Fuerzas Armadas - ESPE, Sangolquí, Ecuador
[3] SDAS Research Group, Ibarra, Ecuador
[4] Yachay Tech University, Urcuquí, Ecuador
[5] Universidad Mariana, Pasto, Colombia
[6] Universidade Federal de Pernambuco, Recife, Brazil
[7] Corporación Universitaria Autónoma de Nariño, Pasto, Colombia
jarsalas@inf.ufrgs.br
https://sdas-group.com/

Abstract. Time-varying data characterization and classification is a field of great interest in both scientific and technology communities. There exists a wide range of applications and challenging open issues such as: automatic motion segmentation, moving-object tracking, and movement forecasting, among others. In this paper, we study the use of the so-called kernel spectral clustering (KSC) approach to capture the dynamic behavior of frames - representing rotating objects - by means of kernel functions and feature relevance values. On the basis of previous research works, we formally derive a here-called tracking vector able to unveil sequential behavior patterns. As a remarkable outcome, we alternatively introduce an encoded version of the tracking vector by converting into decimal numbers the resulting clustering indicators. To evaluate our approach, we test the studied KSC-based tracking over a rotating object from the COIL 20 database. Preliminary results produce clear evidence about the relationship between the clustering indicators and the starting/ending time instance of a specific dynamic sequence.

Keywords: Kernels · Motion tracking · Spectral clustering

1 Introduction

Today, the analysis of dynamic (also known as time-varying) data is a great-of-interest and highly-relevant topic within areas such as: data science, automation

O. Oña-Rocha—This work is supported by SDAS Research Group (www.sdas-group.com).

V. R. Cota et al. (Eds.): LAWCN 2019, CCIS 1068, pp. 30–40, 2019.
https://doi.org/10.1007/978-3-030-36636-0_3

and pattern recognition - benefiting then several scientific and technology fields. Among its remarkable applications, it is worth mentioning: motion segmentation [1], video analysis [2], and object tracking [3]. In this connection, the theoretical approaches that have shown to be a significant tool for dealing with dynamic data are the matrix spectral techniques along with graph-cut approaches. Specifically, the so-called kernel spectral clustering (KSC), introduced in [4], is a well-reputed state-of-the-art method. KSC - broadly speaking - is a generalization of a weighted, kernelized version of principal component analysis within a non-supervised, least-squares-support-vector-machines framework. Furthermore, in a previous work [5], we demonstrated the usefulness of KSC - powered by a feature relevance analysis [6] - for dealing with time-varying data problems. Particularly, the segmentation of a sequence of moving level curves into motion clusters was studied.

In this work, from such previous studies, we explore the use of the KSC-based tracking approach to segment into meaningful motion stages a sequence of frames describing rotating objects. A noticeable contribution of this work is the possibility to validate an afore-introduced approach for estimating a tracking vector, by means of an encoded version thereof. Such an encoding procedure is carried out so that clustering indicators matrix is converted into a vector holding decimal numbers, and therefore the clustering membership is truly unveiled. Experiments are carried out over a sequence of frames of a rotating object from the `COIL 20` [7]. Clustering parameters, such as the number of clusters, type of kernel function, and kernel parameters are empirically set. Obtained results proves that the explored tracking vector is able to automatically identify motion stages in a sequence of frames (video) of of objects submitted to a rotational movement.

The remaining of this paper is structured as follows: Sect. 2 briefly outlines the KSC formulation and its general use for unsupervised grouping. Then, in Sect. 3, both the already-developed KSC-based tracking and the novel encoded tracking vector are explained. Sections 4 and 5 holds the experimental setup and results, respectively. Finally, in Sect. 6 the concluding remarks are drawn.

2 Kernel Spectral Clustering

Spectral clustering techniques have successfully been used for separating a dataset into a K disjoint subsets [8]. The Kernel Spectral Clustering (KSC) [4] consists in using a Least-Squares Support Vector Machine (LS-SVM) as a clustering technique. For further statements, consider the notation described in Table 1.

Given a set of N data points $\boldsymbol{X} = \{\boldsymbol{x}\}_{i=1}^{N}$, being $\boldsymbol{x}_i \in \mathbb{R}^d$ the i-th data point, and $\boldsymbol{X} \in \mathbb{R}^{N\times d}$ the data matrix, it is possible to assume a latent variable $\boldsymbol{E} \in \mathbb{R}^{N\times n_e}$ as $\boldsymbol{E} = \boldsymbol{\Phi W} + \boldsymbol{1}_N \otimes \boldsymbol{b}^\top$ as a model for the projections with $\boldsymbol{\Phi} = \left(\boldsymbol{\phi}(\boldsymbol{x}_1)^\top, \ldots, \boldsymbol{\phi}(\boldsymbol{x}_N)^\top\right)^\top$, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times d_h}$ being the high dimensional representation of the input data such that $\phi(\cdot)$ is the function that maps data from the original dimension to a higher one d_h, i.e., $\boldsymbol{\phi}(\cdot) : \mathbb{R}^d \to \mathbb{R}^{d_h}$; meanwhile, the weighting

Table 1. Mathematical notation

Notation	Description
$\boldsymbol{A}^\top$	Transpose of the vector or matrix $\boldsymbol{A}$
$\mathbf{I}_n$	n-dimensional identity matrix
$\mathbf{1}_n$	n-dimensional ones vector
$\phi(\cdot)$	Feature mapping function
$\mathcal{K}(\cdot,\cdot)$	Kernel function
$\boldsymbol{\Omega} = [\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)]$	Kernel matrix
$\otimes$	Kronecker product
$\mathrm{tr}(\cdot)$	Trace operator
$\mathrm{sgn}(\cdot)$	Sign function
$\circ$	Hadammard product

factor matrix is defined by $\boldsymbol{W} = (\boldsymbol{w}^{(1)}, \cdots, \boldsymbol{w}^{(n_e)})$, $\boldsymbol{W} \in \mathbb{R}^{d_h \times n_e}$; and $\boldsymbol{b} = [b_1, \ldots, b_{n_e}]$ the vector that contains the bias terms, $\boldsymbol{b} \in \mathbb{R}^{n_e}$ with n_e as the number of considered support vectors.

Then, following a LS-SVM [4] formulation, the primal formulation of KSC optimization problem can be expressed in matrix terms [9], as follows:

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

Being $\boldsymbol{\Gamma} = \mathrm{Diag}([\gamma_1, \ldots, \gamma_{n_e}])$ the diagonal matrix composed by the regularization terms. For solving KSC problem, it is necessary to form the corresponding Lagrangian of previous problem, as follows:

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

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

Consequently, we define the Karush-Kuhn-Tucker (KKT) conditions by solving the partial derivatives on $\mathcal{L}(\boldsymbol{E}, \boldsymbol{W}, \boldsymbol{\Gamma}, \boldsymbol{A})$. Then, the optimization problem defined in the Eq. (1) becomes a dual problem: $\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{V}\boldsymbol{H}\boldsymbol{\Phi}\boldsymbol{\Phi}^\top\boldsymbol{A}$, by eliminating the primal variables, where $\boldsymbol{\Lambda} = \mathrm{Diag}(\lambda_1, \ldots, \lambda_{n_e})$ is a diagonal matrix formed by the eigenvalues $\lambda_l = N/\gamma_l$; $\boldsymbol{H} \in \mathbb{R}^{N \times N}$ is the centering matrix define as

$$\boldsymbol{H} = \boldsymbol{I}_N - 1/(\mathbf{1}_N^\top \boldsymbol{V} \mathbf{1}_N)\mathbf{1}_N \mathbf{1}_N^\top \boldsymbol{V}. \tag{2}$$

Additionally, in order to satisfying the condition $\boldsymbol{b}^\top \mathbf{1}_N = 0$ resulting from KKT conditions, the bias vector $\boldsymbol{b}$ can be chosen as a centering vector (i.e. with zero mean) as follows:

$$b_l = -1/(\mathbf{1}_N^\top \boldsymbol{V} \mathbf{1}_N)\mathbf{1}_N^\top \boldsymbol{V} \boldsymbol{\Omega} \boldsymbol{\alpha}^{(l)}. \tag{3}$$

Moreover, the kernel matrix $\boldsymbol{\Omega} = [\Omega_{ij}] = \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j), i, j \in [N]$, is created applying the kernel trick $\boldsymbol{\Omega} \in \mathbb{R}^{N \times N}$ with $\boldsymbol{\Omega} = \boldsymbol{\Phi}\boldsymbol{\Phi}^\top$. Likewise, the matrix $\boldsymbol{A}$ turns into the eigenvectors, resulting in a set of projections calculated by means of the following formula:

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

Considering that the kernel matrix is mathematically equivalent to the similarity matrix used in conventional graph-based clustering methods, and considering $\boldsymbol{V} = \boldsymbol{D}^{-1}$ with $\boldsymbol{D} = \mathrm{Diag}(\boldsymbol{\Omega}\mathbf{1}_N)$, $\boldsymbol{D} \in \mathbb{R}^{N \times N}$ begin the degree matrix; thus, it is possible to infer that the $K-1$ eigenvectors composed by the largest eigenvalues are cluster indicators and therefore, $n_e = K - 1$ [10]. Afterward, the eigenvectors can be codified based on that both each cluster has a single and unique coordinate system in the $K-1$-dimensional eigenspace; and two points, of the same orthant in the corresponding eigenspace, belong to the same cluster [10]. Therefore, we obtain the code book

$$\widetilde{\boldsymbol{E}} = \mathrm{sgn}(\boldsymbol{E}), \tag{5}$$

by binaryzing the rows of the projection matrix $\boldsymbol{E}$ (using the the sign function $\mathrm{sgn}(\cdot)$), and therefore its corresponding rows become codewords enabling the the formation of the holding-similar-samples clusters according to the minimal Hamming distance. Following the pseudo-code (Algorithm 1) to perform KSC is shown.

Algorithm 1. Kernel spectral clustering: $[\boldsymbol{A}, \boldsymbol{\Lambda}, \widetilde{\boldsymbol{E}}] = \mathrm{KSC}(\boldsymbol{X}, \mathcal{K}(\cdot,\cdot), K)$

1: **Input**: K, $\boldsymbol{X}$, $\mathcal{K}(\cdot,\cdot)$
2: Form the kernel matrix $\boldsymbol{\Omega}$ such that $\Omega_{ij} = \mathcal{K}(\boldsymbol{y}_i, \boldsymbol{x}_j)$
3: Calculate matrix $\boldsymbol{H}$ and $\boldsymbol{b}$ as stated in equations (2) and (3), respectively.
4: Compute the eigendecomposition from the dual the problem: $\boldsymbol{A}\boldsymbol{\Lambda} = \boldsymbol{V}\boldsymbol{H}\boldsymbol{\Omega}\boldsymbol{A}$
5: Determine $\boldsymbol{E}$ through $\boldsymbol{E} = \boldsymbol{\Omega}\boldsymbol{A} + \mathbf{1}_N \otimes \boldsymbol{b}^\top$
6: Form the training codebook by binarizing $\widetilde{\boldsymbol{E}} = \mathrm{sgn}(\boldsymbol{E})$
7: **Output:** $\boldsymbol{A}, \boldsymbol{\Lambda}, \widetilde{\boldsymbol{E}}$

3 Time-Varying Data Analysis via KSC

3.1 KSC-Based Tracking

Following the work done by Wolf and Shashua [11], which introduces a function regarding a non-negative matrix for a relevance analysis, along with the developments presented in [6], we build an optimization problem for obtaining the ranking values for samples instead of features. Focusing on the task of interest, we define the non-negative matrix as $\boldsymbol{\Omega}$ and the data matrix $\boldsymbol{X}$ is formed taking each row as a frame, i.e., $\boldsymbol{x}_i$ represents the coordinates vectors of the i-th frame. More specifically, by considering a sequence of N_f, denoted as $\{\boldsymbol{\mathcal{X}}^{(0)}, \ldots, \boldsymbol{\mathcal{X}}^{(N_f-1)}\}$, the whole (frame) data matrix will be then $\boldsymbol{X} = (\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_{N_f}^\top)^\top$, such that $\boldsymbol{x}_t = \text{vec}(\boldsymbol{\mathcal{X}}^{(t)})$, where $t \in \{1, \ldots, N_f\}$ and vec$(\cdot)$ is a vectorization operator.

Thus, the Eq. (4) becomes an energy maximization problem, stated as follows:

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

The orthonormal rotation matrix $\boldsymbol{U} \in \mathbb{R}^{N \times n_e}$ is formulated such that the linear transformation of kernel matrix is in the form $\boldsymbol{Z} = \boldsymbol{\Omega}\boldsymbol{U}, \boldsymbol{Z} \in \mathbb{R}^{N \times n_e}$. Following the procedure described in Sect. 2, it is possible to formulate that $\text{tr}(\boldsymbol{U}^\top \boldsymbol{\Omega}^\top \boldsymbol{\Omega} \boldsymbol{U}) = \text{tr}(\boldsymbol{\Lambda}^2)$ and therefore a suitable solution for the problem is $\boldsymbol{U} = \boldsymbol{A}$. So, the ranking vector $\boldsymbol{\eta} \in \mathbb{R}^N$, as explained in [6], can be expressed as a linear combination of vectors $\boldsymbol{\alpha}^{(l)}$:

$$\boldsymbol{\eta} = \sum_{l=1}^{n_e} \lambda_l \boldsymbol{\alpha}^{(l)} \circ \boldsymbol{\alpha}^{(l)}. \tag{7}$$

Subsequently, the ranking factor η_i can be seen as a single value representing a unique frame in a sequence. In such vein, $\boldsymbol{\eta}$ becomes a tracking vector.

3.2 Encoded Tracking Vector

In this section we describe the proposed encoding approach for comparing frame tracking given by the original approach. This encoding approach is inspired by the procedure explained in [12].

As discussed in [5,13], given the KKT conditions applied to the dual formulation of the KSC problem, the clusters can directly be recognized, as the geometrical location of projected data points $\boldsymbol{E}$ in every single orthant represents an unique cluster. In other words, clusters can be encoded with binary indicators as expressed in Eq. (5). Consequently, we can obtain crisp values from the cluster indicators as the rows $\widetilde{\boldsymbol{e}}_i$ $(\forall i, i \in \{1, \ldots, N\})$ of matrix $\widetilde{\boldsymbol{E}}$ can be directly converted from binary to decimal numbers. Nonetheless, here it is preferred to constraint such a conversion as the maximum resulting number will be the expected number of clusters. Then, binary codes are converted into decimal numbers upon order of appearing, from 1 to K to reach the encoded tracking vector $\widetilde{\boldsymbol{\eta}} \in \mathbb{R}^d$.

So, to exemplify our encoding approach, let us consider the following example with $n_e = 4$:

$$\boldsymbol{E} = \begin{pmatrix} 2.7 & 2.1 & -0.4 & 4.1 \\ 4.3 & 2.5 & -0.5 & -1.3 \\ 2.3 & 1.5 & -0.5 & 4.3 \\ 1.3 & -1.5 & -0.5 & 2.3 \\ 1.3 & 2.5 & -0.5 & 4.3 \end{pmatrix},$$

yielding an encoded matrix in the form:

$$\widetilde{\boldsymbol{E}} = \begin{pmatrix} 1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & -1 & 1 \end{pmatrix}, \tag{8}$$

and therefore its $\widetilde{\boldsymbol{\eta}}$ will correspondingly be given by:

$$\widetilde{\boldsymbol{\eta}} = \begin{pmatrix} 1 \\ 1 \\ 2 \\ 3 \\ 1 \end{pmatrix}.$$

3.3 KSC-Based Tracking Algorithm

The steps for calculating the proposed KSC-based tracking (KSCT) vectors are summarized in Algorithm 2.

Algorithm 2. KSCT: $[\boldsymbol{\eta}, \widetilde{\boldsymbol{\eta}}] = \text{KSCT}\left(\{\boldsymbol{X}^{(0)}, \ldots, \boldsymbol{X}^{(N_f-1)}\}, K\right)$

Input: Number of clusters K, a frame sequence $\{\boldsymbol{\mathcal{X}}^{(1)}, \ldots, \boldsymbol{\mathcal{X}}^{(N_f)}\}$, a kernel function $\mathcal{K}(\cdot, \cdot)$

1. Form the frame matrix $\boldsymbol{X} = [\boldsymbol{x}_1^\top, \ldots, \boldsymbol{x}_{N_f}^\top]$ such that $\boldsymbol{x}_t = \text{vec}(\boldsymbol{\mathcal{X}}^{(t)})$
2. Apply KSC over $\boldsymbol{X}$ with K to get the eigenvalues $\boldsymbol{\Lambda} = \text{Diag}(\lambda_1, \ldots, \lambda_{\tilde{n}_e})$ and eigenvectors
   $\boldsymbol{A} = [\boldsymbol{\alpha}^{(1)}, \cdots, \boldsymbol{\alpha}^{(\tilde{n}_e)}]$: $[\boldsymbol{A}, \boldsymbol{\Lambda}, \widetilde{\boldsymbol{E}}] = \text{KSC}(\boldsymbol{X}, \mathcal{K}(\cdot, \cdot), K)$
3. Compute $\boldsymbol{\eta} = \sum_{\ell=1}^{\tilde{n}_e} \lambda_\ell \boldsymbol{\alpha}^{(\ell)} \circ \boldsymbol{\alpha}^{(\ell)}$ with $\tilde{n}_e = K - 1$
4. Normalize $\boldsymbol{\eta}$ as $\boldsymbol{\eta} \leftarrow \boldsymbol{\eta} / \max |\boldsymbol{\eta}|$
5. Obtain $\widetilde{\boldsymbol{\eta}}$ by encoding into decimal numbers $\widetilde{\boldsymbol{E}}$

Output: Tracking vectors $\boldsymbol{\eta}, \widetilde{\boldsymbol{\eta}}$

4 Experimental Setup

4.1 Database

For experiments, we use an object of the well-known database `COIL 20` introduced in [7], which is an image bank consisting of 72 gray-level images of 20 different objects placed at different angles (72) - rotated at every 5 degrees. Specifically, we pick the `object # 4` as shown in Fig. 1. The 72 images (one per angle/pose) $\mathbb{I}$ are in size 128×128 pixels, which are firstly re-scaled at a 50 %, yielding then final RGB images as $\boldsymbol{\mathcal{X}}^{(t)} \in \mathbb{R}^{64\times64}$, being $t \in \{0, \ldots, 71\}$. Subsequently, a data matrix is formed by vectoryzing the RGB images. Therefore, the number of data points is $N = 64 \times 64 \times 3 = 12288$, as well as the number of variables is $d = 72$ (being the same number as N_f), which means that the data matrix to be clustered is $\boldsymbol{X} \in \mathbb{R}^{12288\times72}$.

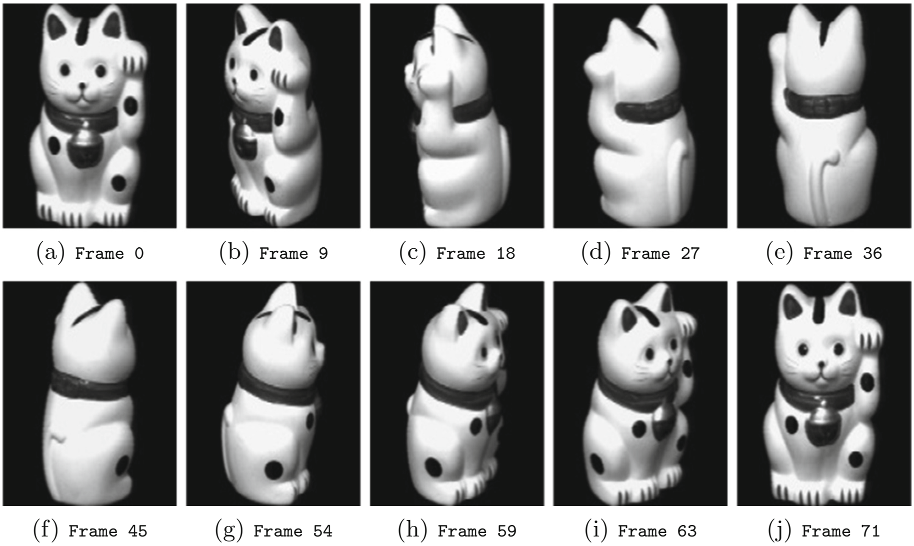


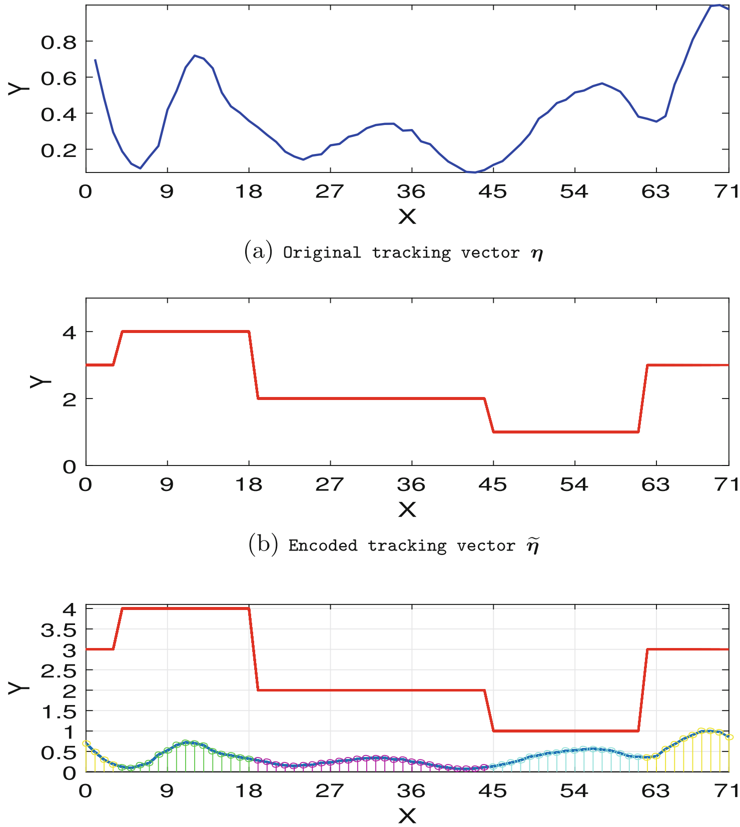
Fig. 1. Some instances of `object # 4` frames from `COIL 20` database.

4.2 Clustering and Kernel Settings

The number of clusters is set to be $K = 4$. The considered kernel function is the conventional Gaussian kernel defined as: $\Omega_{ij} = \exp(-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2/(2\sigma^2)$, where $\|\cdot\|$ denotes the Euclidean norm and the scale parameter σ is set empirically as 30.

5 Results and Discussion

For analyzing the sequence of frames arranged into matrix $\boldsymbol{X}$, we first apply KSC. Then, with the KSC outcomes, the vector $\boldsymbol{\eta}$ is calculated using the Eq. (7). From Fig. 2, we can observe the process of the dynamic behaviour captured by the KSC-based tracking, as follows: Fig. 2(a) and (b) shows the plotting of the original tracking vector $\boldsymbol{\eta}$ and the encoded version $\widetilde{\boldsymbol{\eta}}$, respectively. In Fig. 2(c), the reference labelling vector is shown, which is obtained directly from the values of $\widetilde{\boldsymbol{\eta}}$.



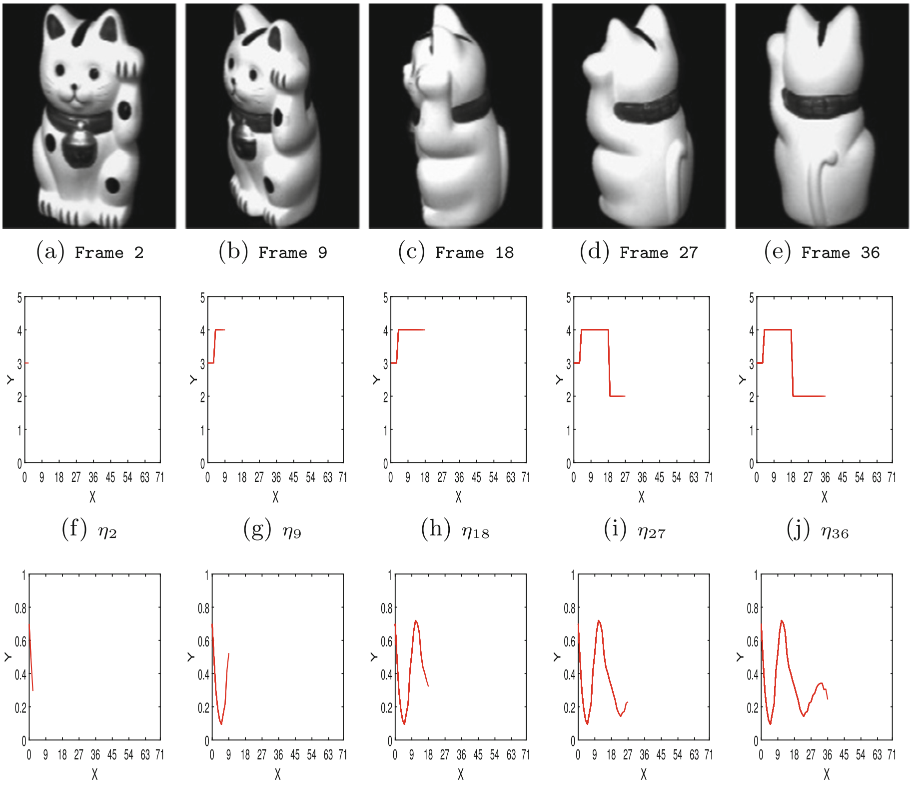
(a) `Original tracking vector` $\boldsymbol{\eta}$

(b) `Encoded tracking vector` $\widetilde{\boldsymbol{\eta}}$

(c) `Encoded-vector-based labels`

Fig. 2. Original and encoded tracking vector plotting. It is depicted the plotting of vectors $\boldsymbol{\eta}$ and $\widetilde{\boldsymbol{\eta}}$ along the 72 frames for `Object # 4` from `COIL 20` database in Figs. 2(a) and (b), respectively. Figure 2(c) is the overlapped representation of the vectors $\boldsymbol{\eta}$ and $\widetilde{\boldsymbol{\eta}}$, while the area under the curve is colored to highlight the motion-stage-based labelling regarding $\widetilde{\boldsymbol{\eta}}$.

From the plotting of $\boldsymbol{\eta}$, it can be seen that its shape is multimodal-like. By comparing vector $\widetilde{\boldsymbol{\eta}}$ with $\boldsymbol{\eta}$, it can be readily noticed that each mode of the $\boldsymbol{\eta}$ plotting corresponds to a different cluster, i.e. a motion stage in the context of video analysis. Such correspondence can be attributed to the fact that the eigenvectors $\boldsymbol{\alpha}^{(l)}$ point out the direction where samples exhibit the most variability measured in term of the generalized inner product $(\boldsymbol{\Phi}^\top \boldsymbol{\Phi})$. In this connection, kernel functions take place and enable the estimation of the inner of high-dimensional representation spaces, wherein resulting clusters are assumed to be linearly separable. The direct connection between the tracking vector $\boldsymbol{\eta}$



(a) `Frame 2` (b) `Frame 9` (c) `Frame 18` (d) `Frame 27` (e) `Frame 36`

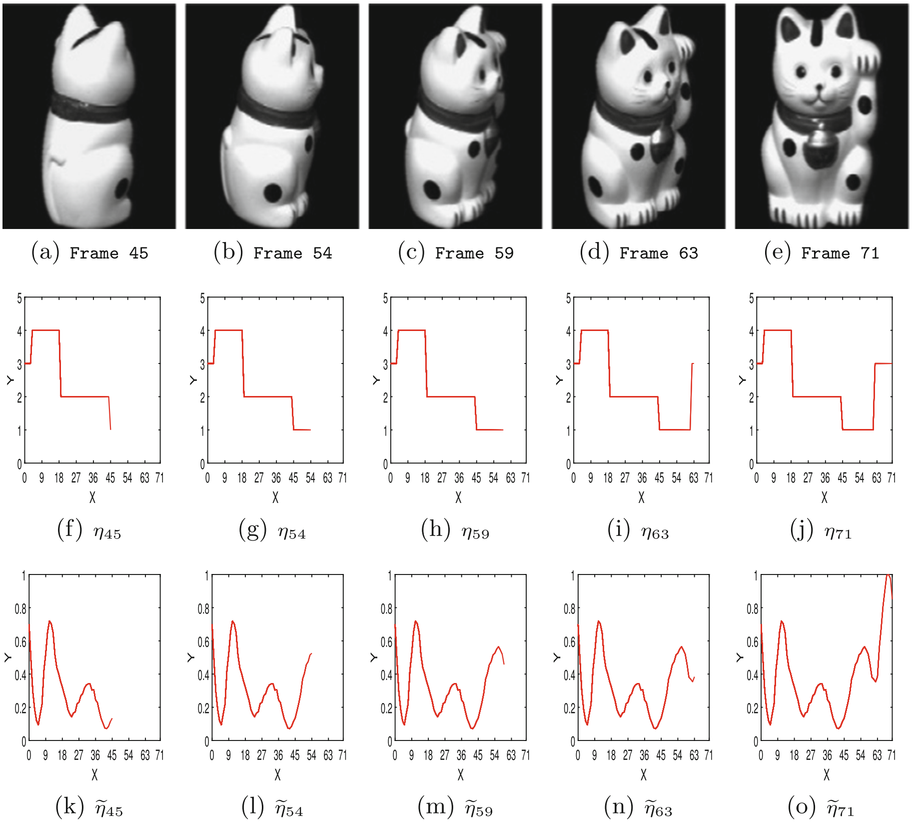
(f) η_2 (g) η_9 (h) η_{18} (i) η_{27} (j) η_{36}

(k) $\widetilde{\eta}_2$ (l) $\widetilde{\eta}_9$ (m) $\widetilde{\eta}_{18}$ (n) $\widetilde{\eta}_{27}$ (o) $\widetilde{\eta}_{36}$

Fig. 3. `Object #4` tracking original frames (2, 9, 18, 27 and 36) and tracking vectors.

and the partition of natural movements from `Object #4` can be plainly appreciated in Figs. 3 and 4, where the top row shows representative frames per cluster while middle row and bottom row depicts the corresponding evolution of the $\boldsymbol{\eta}$ and $\widetilde{\boldsymbol{\eta}}$ curve, respectively.

As noticed, each mode between inflections forms a concave curve in the plotting, which means that another natural cluster within the sequence has appeared. Such cluster splitting can even be determined by simple inspection. Besides, the encoding vector allows then for validating the premise that vector $\boldsymbol{\eta}$ is able to divide the sequence of frames into natural motion stages (clusters), when the clustering settings are appropriate. An instance of the motion segmentation effect is depicted in the video available at: https://sdas-group.com/gallery/.



(a) Frame 45 (b) Frame 54 (c) Frame 59 (d) Frame 63 (e) Frame 71

(f) η_{45} (g) η_{54} (h) η_{59} (i) η_{63} (j) η_{71}

(k) $\widetilde{\eta}_{45}$ (l) $\widetilde{\eta}_{54}$ (m) $\widetilde{\eta}_{59}$ (n) $\widetilde{\eta}_{63}$ (o) $\widetilde{\eta}_{71}$

Fig. 4. `Object #4` tracking original frames (45, 54, 59, 63 and 71) and tracking vectors.

6 Conclusions

The dynamic point of view of the greatly wide field of data analysis entails a complex and difficult issue to tackle, since the input data vary along the time. Even more, the intrinsic dynamics - involved during the movement itself - adds more complexity to the subsequent data processing task. On this regard, one of the challenging open issues is the automatic motion segmentation - which can be readily evaluated over rotating objects. In this sense, we have proved that KSC method represents a powerful, suitable tool.

In this work, the use of non-supervised approaches is preferred since, in real-world video applications, an enough amount of labelling is infeasible or prohibitive. Notwithstanding, the disadvantage of working on rotating objects analysis within unsupervised settings is that no automatic motion segmentation can directly be generated by means of a tracking function (here-called tracking vector). At this point, to overcome this obstacle, we have introduced a clustering-indicators-based encoding procedure, so that the quality of the original multimodal tracking vector can be measured.

Acknowledgments. Authors acknowledge the SDAS Research Group (www.sdas-group.com) for its valuable support.

References

1. Sandhu, M., Upadhyay, S., Krishna, M., Medasani, S.: Motion segmentation using spectral clustering on Indian road scenes. In: The European Conference on Computer Vision (ECCV) Workshops, September 2018
2. Huang, W., Zhang, P.: A novel framework to localize moving targets in video surveillance systems via spectral clustering. Proc. Comput. Sci. **147**, 480–486 (2019). 2018 International Conference on Identification, Information and Knowledge in the Internet of Things. http://www.sciencedirect.com/science/article/pii/S1877050919302996
3. Aamer, B., et al.: Self-tuning spectral clustering for adaptive tracking areas design in 5G ultra-dense networks. arXiv e-prints, arXiv:1902.01342, February 2019
4. 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, 2006. IJCNN 2006, pp. 138–144. IEEE (2006)
5. Oña-Rocha, O.R., et al.: Automatic motion segmentation via a cumulative kernel representation and spectral clustering. In: Yin, H., et al. (eds.) IDEAL 2017. LNCS, vol. 10585, pp. 406–414. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68935-7_44
6. Peluffo Ordoñez, D.H., Lee, J.A., Verleysen, M., Rodriguez, J.L., Castellanos-Dominguez, G.: Unsupervised relevance analysis for feature extraction and selection. A distance-based approach for feature relevance. In: 3rd International Conference on Pattern Recognition Applications and Methods (ICPRAM 2014) (2015)
7. Nene, S.A., Nayar, S.K., Murase, H.: Columbia object image library, COIL-20, Technical report (1996)
8. Langone, R., Mall, R., Alzate, C., Suykens, J.A.K.: Kernel spectral clustering and applications. In: Celebi, M.E., Aydin, K. (eds.) Unsupervised Learning Algorithms, pp. 135–161. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-24211-8_6
9. Diego Peluffo-Ordóñez, E. M.-O.: Theoretical developments for interpreting kernel spectral clustering from alternative viewpoints. Adv. Sci. Technol. Eng. Syst. J. **2**(3), 1670–1676 (2017). https://astesj.com/v02/i03/p208/
10. Alzate, C., Suykens, J.A.K.: Multiway spectral clustering with out-of-sample extensions through weighted Kernel PCA. IEEE Trans. Pattern Anal. Mach. Intell. **32**(2), 335–347 (2010)
11. 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
12. Alzate, S.J.C.: Highly sparse kernel spectral clustering with predictive out-of-sample extensions (2010)
13. Peluffo-Ordóñez, D.H., García-Vega, S., Álvarez-Meza, A.M., Castellanos-Domínguez, C.G.: Kernel spectral clustering for dynamic data. In: Ruiz-Shulcloper, J., Sanniti di Baja, G. (eds.) CIARP 2013. LNCS, vol. 8258, pp. 238–245. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41822-8_30