Unsupervised Learning of Behavioural Patterns for Video-Surveillance

Nicoletta Noceti, Matteo Santoro, Francesca Odone

Unsupervised Learning of Behavioural Patterns for Video-Surveillance

Nicoletta Noceti, Matteo Santoro, and Francesca Odone

DISI, Via Dodecaneso, 35 — 16146 Genova, ITALY
{noceti,santoro,odone}@disi.unige.it

Abstract. Unsupervised learning is a way to extract knowledge from noisy and complex sets of unlabeled data. The video-surveillance setting provides a potentially huge amount of unlabeled information on a given scene. In this paper we explore the use of spectral clustering to learn common behaviours from sets of dynamic events from a video-surveillance system. In particular we discuss how temporal data, characterized by variable lengths and an internal ordering, may be exploited effectively by means of appropriate representations and kernel functions. An experimental assessment on synthetic and real data guides us to an effective solution based on the use of strings.

1 Introduction and Background

Classification based on dynamic information is a challenging problem in computer vision and plays a central role in a number of present-day applications including visual surveillance, human-machine interface, and semantic retrieval from videos [1,2,3,4]. Although the urging need for effective systems in all these contexts is fuelling an intense research activity, several crucial issues are still either not solved or even not addressed. Specifically in the field of visual surveillance, with which we are mainly concerned in this paper, an important issue is the efficient analysis of *behavioural patterns* of people activities in the scene. Given the huge variety of possible behaviours in moderately complex scenarios, a fully supervised classification approach is likely to fail to produce systems that are both *robust* – i.e. able detect and recognise effectively many classes of events – and *flexible*, i.e. easy to be modified to recognise new (classes of) events. The peculiarity of video-surveillance is a great help in this direction. Usually a system observes the same scene for very long periods. Thus, such system could *learn* common behaviours in that scene from long-time observations, carried out in an initial or intermediate calibration stage. This aim calls for some form of unsupervised data analysis that does not depend on data labelling and does not require any a priori decision on the classes of interest. An analysis of available literature makes it clear that unsupervised methods for behaviour understanding or, more in general, to time series analysis are quite limited - see [5,6,7,8].

Many successful methods have been proposed to discover classes of similar pattern in a dataset. Unfortunately most of them are not tailored for clustering sequential data, such as temporal series. In fact, while analysing temporal

data, the *temporal dimension* usually induces specific, inherent structure to the sequence of feature vectors that is likely to be disregarded by traditional clustering methods. Furthermore, sequences of different lengths cannot be compared in a unique way. Previous works on clustering temporal series approached the problem from two distinct viewpoints (see [6] and references therein). The first approach relies on modifying suitable existing algorithms for clustering static data in such a way that time series data can be handled. An alternative is to convert temporal series data into the form of static data so that the existing algorithms for clustering static data can be directly used, trying to keep information on the structure of data. Our work is more related to this latter approach.

One of the main contributions of this paper is to set out an analogy between behaviour analysis for visual surveillance systems and the unsupervised classification of temporal/sequential series. Since our primary objective is to develop efficient strategies for analysing the behaviour of people moving in a scene, this analogy should allow for a profitable deployment of successful approaches to temporal series analysis. The underlying idea is that recent advances in unsupervised learning technologies may play a major role in developing robust yet flexible visual surveillance systems. In particular, we argue that spectral clustering-based algorithms achieve good performances in analysing the behaviours once suitable similarity measures have been chosen for comparing temporal series. Within this general framework we discuss an algorithmic architecture, based on unsupervised learning, for building a module of a video-surveillance system that learns automatically usual behaviours in a scene of interest from the observation of the scene itself. Understanding what is common or usual in a scene sets the basis for the design of anomaly detection systems.

The processing pipeline we refer to in this paper has been effectively described in [1]: first, moving objects are detected (motion segmentation) and possibly classified in the scene. Second, single entities/objects previously detected, are tracked along the video sequence in order to build a model of the dynamic evolution of the trajectory as long as the *identity* of the object can be preserved. Third, the trajectories are analysed in order to understand and describe the behaviours. Finally, information coming from multiple cameras, if available, is registered and combined. In this paper we focus on the third step, hypothesising that video inputs have been conveniently pre-processed insofar that a number of feature vector sequences are computed and then associated to objects moving in the scene. The choice of the specific feature map by means of which to *represent* a specific moving object at each time instant is not crucial to our analysis: we only require that the sequences are described by means of $d \times T$ matrices, where d is the number of components of the feature map and T is the number of elements in the sequence. In general, T may change from sequence to sequence.

In order to fully exploit spectral clustering algorithm, we focus on two basic issues: (1) how data should be represented, and (2) which is a proper similarity measure to compare them. In Section 2 different representation schemas are described and appropriate kernel functions are plugged into the general spectral clustering algorithm. An experimental analysis on synthetically generated data

(Sec. 3) shows that the combination of a language-based representation and string kernels are the most appropriate for the analysed application domain. This combination is then applied to find behaviour patterns in datasets of real image sequences used as a benchmark by the video-surveillance community. Preliminary results on such datasets speak in favour of the appropriateness of our approach, as discussed in Section 4.

2 The Approach

In this section we describe an algorithmic schema for the unsupervised analysis of behavioural patterns associated to image sequences. After reviewing some popular approaches to the *representation* of temporal series, we discuss the spectral clustering-based algorithm to discover behavioural patterns and describe suitable kernel functions to deal with data characterised by a variable length and an internal ordering due to temporal coherence. Finally, we motivate empirically the choices.

2.1 Representation of temporal series

We denote $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N}$ the dataset of N temporal series, where each datum $\mathbf{x}_i$ is a sequence of k_i vectors in some Euclidean space $\mathbb{R}^d$, i.e. $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,k_i})^T$ and $x_{i,t} \in \mathbb{R}^d$. A natural way to interpret the index t is as the temporal index. The most popular approaches to describe information contained in $\mathbf{X}$ are the following:

1. *Non-Parametric Representation of Temporal Series.* If a non-parametric approach is adopted, then the representation of a set of temporal series is simply $\mathbf{X}$ itself. For instance, in the special case where all the series have the same length - $k_i = k \quad \forall i$ - it is sometimes convenient to think of a $k \times (d \cdot N)$ matrix $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)$.
   This allows for a compact formulation of some structural properties of the data in terms of vector matrix calculations. When the lengths k_i are not all the same, zero-padding or resampling [9] techniques may be applied or, more commonly, parametric representations may be considered. In the following we review a set of possible parametric approaches.
2. *Parametric representation of temporal sequences by means of curve fitting.* In a parametric setting, one aims at representing an event by means of a fixed-length feature vector in order to overcome two crucial issues related to temporal series: (1) the fact that the events to be represented do not last the same amount of time, and (2) they do not have the same spatio-temporal evolution (sometimes even within events of the same class).
   A popular and easy-to-implement parametric approach is based on *curve fitting*, in which a regression function is built for each sequence (see [10] and references therein). Here an important step is to choose a specific family of fitting functions. Such choice depends on the distribution of the points in the

scene, which is strongly connected to the complexity of observed behaviours. Common alternatives are polynomials for moderately complex series [11] or a family of *spline* models for more general and complex scenario [10]. As described below, in our experiments we used second-degree polynomials fitted by means of a simple least squares-based method.

3. *Parametric representation of temporal sequences as probability models.* In a probabilistic setting, temporal sequences modelling may consist of building a probabilistic model of the present observation given all past observations: $p(x_{i,t}|x_{i,1}, y_{i,2}, \ldots, y_{i,t-1})$. Because the history of observations can grow arbitrarily large it is necessary to limit the complexity of such a model. A common approach to overcome this problem is to limit the window of past observations. The simplest model, $p(x_{i,t}|x_{i,t-1})$, is known as a first-order Markov model. One popular choice of probabilistic models that make this assumption, are the hidden Markov models (HMMs), of which it is always possible to write down explicitly the likelihood function [12]:

$$p(\mathbf{x}|\theta) = \sum_{q_0,\ldots,q_k} p(x_0|q_0)p(q_0) \prod_{t=1}^{k} p(x_t|q_t)p(q_t|q_{t-1}) \tag{1}$$

where the $\mathbf{q} = \{q_1, \ldots, q_k\}$ are the hidden states corresponding to each element of $\mathbf{x}$ and can take on one of the discrete values $\{1, \ldots, M\}$. In order to specify an HMM, one has to specify the parameter vector $\theta = (\pi, \alpha, \mu, \Sigma)$, where $\pi \in \mathbb{R}^M$ represents the initial state probability distributions $\pi_i = p(q_0 = i)$ for all $i = 1, \ldots, M$. $\alpha \in \mathbb{R}^{M \times M}$ is the state transition probability distribution, i.e. $\alpha_{ij} = p(q_t = j|q_{t-1} = i)$. $\mu = \{\mu_1, \ldots, \mu_M\}$ and $\Sigma = \{\Sigma_1, \ldots, \Sigma_M\}$ represents the mean and covariance of the Gaussian distributions for each state i, i.e. the emission density $p(x_t|q_t = i)$ is a normal distribution $\mathcal{N}(x_t|\mu_i, \Sigma_i)$.

4. *Parametric Representation of Temporal Sequences as Strings.* A sequence $\mathbf{x}$ can be also considered a concatenation of symbols from a finite *alphabet* $\mathcal{A}$. As for the choice of the alphabet we consider a partition of the $d-$dimensional input space. Fully relying on an unsupervised approach to the problem we estimate an appropriate partition from the available dataset. To this purpose we cluster data in the $d-$dimensional input space, disregarding the temporal component - i.e. $\{\{x_{i,t}\}_{t=1,\ldots,k_i}\}_{i=1,\ldots,N}$.
In our experiments spectral clustering is used to cluster data. As for the kernel choice, off-the-shelf kernels, such as the Gaussian kernel, may be used. Out-of-sample extension produces a full partition of the space. Once the alphabet is built, a temporal series $\mathbf{x}$ may be translated into a string s with an association of each element $x_j \in \mathbb{R}^d$ to the partition it belongs to. To obtain compressed descriptions that capture the peculiarities of each behaviour we consider only transitions between states, skipping the replicates of the same symbol, i.e., if a string contains replicates of the symbol $u \in \mathcal{A}$ it can be compressed as: $\alpha u^{\lambda} \beta \rightarrow \alpha u \beta$, for every arbitrary substrings α and β.

2.2 Clustering of temporal series

Spectral clustering refers to a class of techniques to partition points into disjoint clusters by analysing the eigenstructure of the Laplacian matrix of a similarity graph associated to the dataset. In [13], the first few eigenvectors of the graph's Laplacian are shown to carry information about the *optimal cut* to partition the graph. Different algorithms have been proposed in the last years that exploit this result (see [14] and [15] for two popular examples). For the experiments, we implemented an efficient recursive algorithm for spectral clustering following [15]. With this approach we do not have to decide the number of the clusters beforehand, but we have only to define the minimum size allowed for the small clusters. The level of granularity is specified through a threshold (CT) on the value of the *cut*. The partitioning is stable across multiple runs of the algorithm because we do not need a final k-means step as in [5] (where the authors slightly adapted the algorithm in [14]).

It is well assessed that, in the spectral approach, clustering results strongly depend on the ability to capture the underlying notion of metric - or the similarity structure - over the input space. In turn, the similarity structure is related to the choice of a proper kernel function on the data. A function to be used as a kernel for spectral clustering should be *symmetric* and *positive*. Since in our experiments we compare different representation schemes we have to choose appropriate kernels. Both in the non-parametric and in the polynomial-based setting we refer to well established kernels such as the Gaussian kernels. The other two approaches require a more careful choice of the kernel able to deal with variable length data [16]. We considered the following:

1. *Probability Product Kernel (PPK).* An efficient measure of similarity among temporal series represented by means of HMMs, introduced in [17], and used in [5] in tasks similar to ours.
   Given two probability distributions representing two pairs of data sequences over the space of all potential observable sequences X, the generalised inner product can be computed by integrating the product of the distributions:

$$K(p(x|\theta), p(x|\theta)) = \int_X p^\beta(x|\theta) p^\beta(x|\theta) dx, \tag{2}$$

   where β is a free parameter which allow for the specification of some properties of the kernel. For instance, if $\beta = 1/2$, the PPK becomes the classic Bhattacharyya affinity metric between two probability distributions, which is a Mercer kernel and can be conveniently used in our context. In addition, it is computable in closed form for a variety of distribution families. For the HMMs, we used the guidelines proposed in [5].
2. *P-spectrum Kernel.* If one interprets the temporal series as strings over a finite alphabet, then it is possible to borrow from studies on text and biological data manipulation [18,19]. The most natural way to compare two series is to count how many transition they have in common from one symbol of the alphabet to a different. This is tightly related to the concept of

spectrum of order P (or P-specturm) of a sequence s to be the histogram of frequencies of all its (contiguous) substrings of length P. In the case of simple transitions between atomic symbols of the alphabet, we are considering basic substrings of length 2. Formally, the kernel may be defined as a feature map of strings followed by an appropriate dot product [16]. The map makes explicit all sub-strings of length P of string s:

$$\phi_u^P(s) = |\{(v_1, v_2) : s = v_1 u v_2\}|, u \in \mathcal{A}^P. \tag{3}$$

The associated kernel between two strings s and t is defined as:

$$K_P(s,t) = \langle \phi^P(s), \phi^P(t) \rangle = \sum_{u \in \mathcal{A}^P} \phi_u^P(s) \phi_u^P(t). \tag{4}$$

String length independence is achieved with by a normalisation [16].

3 Experimental Assessment

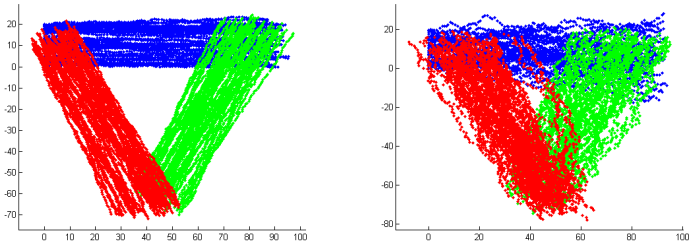
In the previous section we introduced a general algorithm for unsupervised analysis of temporal series and proposed a number of variants for data representation. By leveraging on some preliminary experimental results, we can now discuss more thoroughly the properties of each specific choice and validate the effectiveness of the overall approach

For the experiments, we generated a number of temporal series of $2D$ points on a plane - see Figure 1. Such data are evocative of a simple yet realistic video surveillance scenario, as shown in Fig. 2), where positions of objects moving along the image plane are used to describe dynamic events. In this case the input space is so that $d = 2$, The sequences are divided into 5 groups with an increasing level of noise (0.1, 0.3, 0.5, 0.7, 0.9 respectively), so to model situations where a smooth object tracking is increasingly more difficult (e.g., because of scene clutter or bad illumination).

Each group of series comprises 3 distinct behavioural models (represented with different colours in Figure 1). In order to make some statistical analysis of the results, we resampled 20 different versions of each group of data. Since in principle some of the proposed kernels should be able to capture more complex properties of temporal series - such as to distinguish between opposite directions - we generated a second database (named $DS6B$) where each of the 3 behaviours is splitted in two subgroups by choosing randomly the direction of the points. Note that in our setting this is equivalent to switch the temporal index.

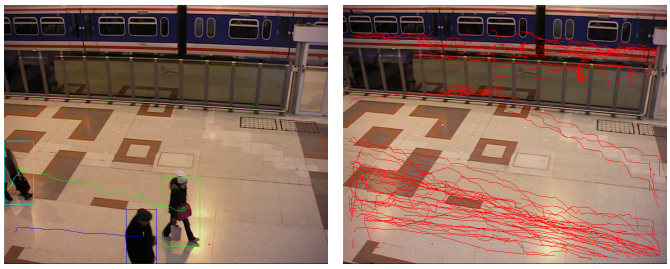
3.1 Data representation and kernel choice

Parametric methods described in Section 2.1 lead to intermediate descriptions of temporal series which enhance only specific characteristics of the data and necessarily loose some others. Moreover, each method poses a number of practical issues that have to be solved adequately.



(a) (b)

Fig. 1. Examples of series in the *DS3B* dataset. On the left the noise level is low and the trajectories are smooth and almost linear. On the right the noise level increases and the classification of the series is far more difficult.



(a) Blob tracking (b) Blob trajectories

Fig. 2. The output of a simple people detection and tracking system from the database PETS06 (see later for details). The figure shows that our synthetic dataset can be conveniently used as a testbed for the proposed method.

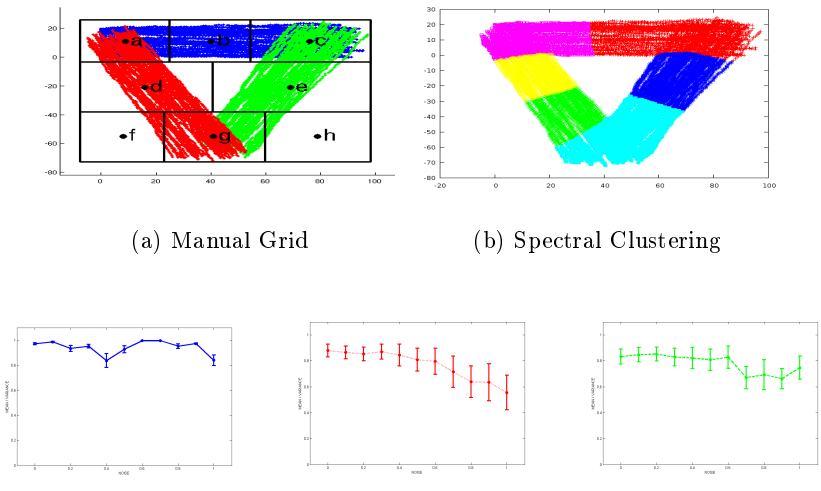
- *Gaussian Kernels in the Parameter Space of the Polynomials.* The preliminary step in the curve fitting approach for representing temporal data is the choice of the family of curves to approximate. Since we decided to use polynomials, the problem becomes the choice of curve degree. The lesser the degree, the smaller is the dimensionality of the parameter space and the faster is the fitting algorithm. However, if the degree is too low, the approximation of the actual points of the series may be too loose and the method may fail to capture important properties of the data. Therefore one always needs to balance between these two important aspects.
  In order to avoid arbitrariness at this stage, we adopted the following heuris-

tic strategy. For each dataset, we first selected randomly a small fraction of series and fit polynomials with increasing degree. The ratio between the number of non zero coefficients and the total number of coefficients represents how complex the data are, and can be used to choose the proper degree for that specific dataset. We observed experimentally that straight lines are sufficient if the level of noise is low (Fig. 1 left), while noisy data can be represented by means of second (at most third) degree curves (Fig. 1 right).

- *Probability product kernels for HMM-based representation.* The inizialisation stage is the very critical for the HMM-based method. First, we note that a major source of concern is the fit of a multiparametric probabilistic model (such as the HMMs) to a single time series, and therefore the effectiveness of an abstract representation of two series in terms of probability distributions. According to [17], since the HMM is just an intermediate step in forming the kernel and to capture the similarity among series, it is not necessary to build a very accurate model. In principle, we agree with this viewpoint, but in our application scenario temporal series corresponding to the same behavioural pattern may be have very different lengths and structure due to errors in the tracking system. These may affect negatively the the algorithm that "learns" the HMM model.

  In order to limit these concerns and reduce the effect of previous errors in the algorithmic pipeline, we test both a common initialisation for all the series and a fully independent model fitting strategy. The results show that the original fully independent approach is not competitive in our scenario.
- *P-spectrum kernels for string-based representations.* When temporal series are described as strings over an alphabet, the first issue is the choice of the alphabet symbols. Above, we already mentioned an automatic quantisation approach in which the feature space is partitioned by means of spectral clustering of all the points in the dataset. The association of a symbol/state to each point in the sequence is shown in Figure3(b), where the colours represent different symbols and each string is a concatenation of symbols. If the feature vectors are defined in a low dimensional space ($d \leq 3$) - which may be the case in our context - then a viable alternative to automatic selection could be manual partitioning - where a user decides how to cluster "reasonably" the feature space (fig. 3(a)).

  For a quantitative assessment of the *quality* of the alphabet definition, we evaluated the *stability* of the representation in terms of how similar the corresponding strings are when two similar sequences are considered. Since the series in dataset $DS3B$ and $DS6B$ are known to belong to a fixed set of behavioural classes, we evaluated the intra-class stability of the different methods. Similarity between strings is computed with the 2-spectrum. Figs. 3(c), 3(d) and 3(e) show mean and variance for different datasets arranged with increasing noise levels over the 20 replicates of the entire dataset. Blue plot refers to the automatic partition, red plot to the definition of a fixed manual partition specified - and kept fixed - at the beginning of all the experiments. Finally, in the green plot an ad-hoc partition is defined for each experiment: even if the results are slightly better that those obtained with



(a) Manual Grid (b) Spectral Clustering

(c) Automatic (d) Manual (e) Manual

Fig. 3. The choice of a proper alphabet. The top-left figure shows the states assigned by manual choice, compared with the result of our automatic procedures (top-right). The botton row shows the stability of the representations while the noise level increases.

the fixed manual partition, this approach is not suitable for real applications.

3.2 Experiments on temporal series categorisation

For each representation scheme, we computed the corresponding feature map, built the similarity matrix of all the series in the dataset and run the spectral clustering algorithm to evaluate the results.
As a first analysis, we inspected the similarity matrices obtained with each kernel function and noise level, see Figure 4. In order to highlight the block structure of the matrices and make the non-quantitative analysis easier, we ordered the series beforehand insofar that the "optimal" method should produce exactly three distinct blocks along the diagonal of the similarity matrix. 2-spectrum kernel combined with string based-representation is very close to get this optimal result when the noise level is low, and it is also the closest one on more noisy data. These results were confirmed also in the subsequent quantitative analysis.

In general, quantitative performance evaluation on the unsupervised domain is difficult to achieve, and no well assessed - and theoretically ground - methods have been proposed so far. The most popular solution to this problem is to analyse labelled datasets for which the ground truth is known - i.e. to use the clustering method as a classification algorithm - and to measure the classifica-

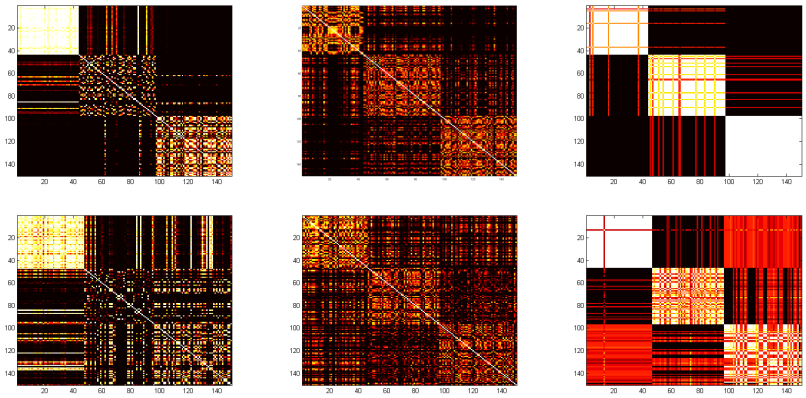


Fig. 4. Similarity matrices: from left to right, gaussian kernel on polynomial coefficients, probabilistic product kernel on HMMs and 2-spectrum kernel on strings. The top row refers to moderately noisy data (0.3), the bottom row the noise level is increased (0.9).

Table 1. Percentages of correct behaviour associations on datasets $DS3B$ and $DS6B$.

	Dataset DS3B					**Dataset DS6B**				
Noise Level	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**
Strings (autom. alphabet)	90.0	80.7	80.0	81.0	79.0	97.8	96.2	87.0	84.1	78.8
Strings (manual alphabet)	92.8	94.0	79.0	79.0	78.0	99.8	98.6	96.2	84.0	67.4
HMMs (common init.)	87.7	81.5	70.4	63.1	58.8	48.2	44.6	39.4	35.4	31.7
HMMs (indep. init.)	60.0	57.4	51.8	48.6	46.8	36.2	34.8	34.2	27.9	30.6
Polynomials	91.0	72.0	56.0	40.0	43.0	-	-	-	-	-
Normalised polynomials	62.0	57.2	52.0	50.5	47.1	-	-	-	-	-

tion performances. Indeed, it is worth pointing our that in many applications of unsupervised learning one is not necessarily interested in assigning a class to "each" datum but to cluster coherent data, discarding noisy or uncommon patterns, see for instance [20].

In our experiments, we rely on the circumstance that for synthetic data it is possible to know the a-priori model by means of which they are generated. Therefore, we can measure the stability of intra-class and inter-class distances, and compare the performances of the methods. Table 1 reports a comparative analysis of the results obtained as the noise level grows. The performances are expressed in percentages and refer to the correct "classification" of the clusters.

The above results provide experimental evidence that, in general, the approach based on the use of string kernel gives better results than the others. Manual partition seems to be more accurate than the automatic method for

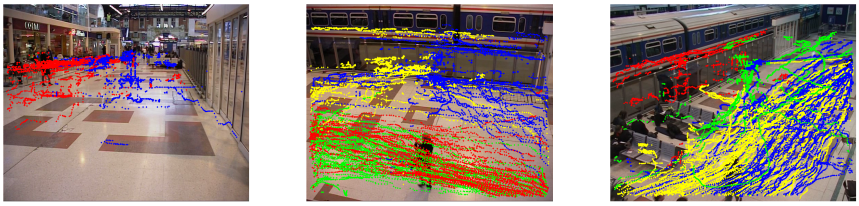


Fig. 5. Preliminary results on real data extracted from the dataset PETS06.

moderate noise levels. However, as the clutter increases, data-driven states estimate provides better and better performances (this is coherent with our conclusions in Section 3.1). Moreover they can deal with bidirectional trajectories maintaining similar accuracy.
As anticipated, in this specific experimental scenario, HMM-based technique depends strongly on the initialization strategy of the HMM density estimation: an independent inizialisation tends to map series of the same cluster to dissimilar feature vectors, and therefore the spectral clustering algorithm is not able to *cut* the graph appropriately. Furthermore, with the probability product kernel seem to fail to capture correctly the directionality of the series, if compared with the performances of 2-spectrum.
Polynomials give very poor results for higher levels of noise and do not consent to distinguish among bidirectional events: in curve fitting we disregard the order in which the points are generated and, thus the direction of the sequence.

4 Discussion

Automatic behaviour understanding may be effectively addressed by unsupervised learning on temporal series. To our knowledge the available literature is lacking both on theoretical and experimental view-points. This paper proposes an analysis of a selection of methods to represent temporal series and analyse them via spectral clustering. All these methods can be cast in a common algorithmic scheme, although they highlight different aspects of the data at hand.

We evaluated the different choices on a set of synthetically generated data at different noise levels. This analysis is motivated by the intent of obtaining a robust method for temporal series analysis to be applied in the video-surveillance domain. Preliminary results, obtained on the PETS06 dataset, available at `http://www.cvg.rdg.ac.uk/PETS2006/data.html`, are promising. Trajectories have been obtained with a change detection module, followed by a Kalman tracker. A colour-based data association limits the effects of occlusions. Trajectories model the person's centroid position evolving in time. Figure 5 shows the results obtained using a state-based representations with 2-spectrum kernel — overlapping sets of trajectories with different colors describe bidirectional patterns.

Future works will be devoted to exploit the effects of long observations on a more complex environment and explore the effects of adopting more complete feature maps and kernels.

References

1. Hu, W., Tan, T.N., Wang, L., Maybank, S.J.: A survey on visual surveillance of object motion and behaviors. IEEE Tran. on Sys., Man and Cyb. **34**(3) (2004)
2. Zhang, Y., Ji, Q.: Active and dynamic information fusion for facial expression understanding from image sequences. IEEE Trans. on PAMI **27**(5) (2005) 699–714
3. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: IEEE Int. Conf. on Computer Vision. (2003) 726–733
4. Robertson, N. Reid, I.: Behaviour understanding in video: a combined method. In: IEEE Proc. on ICCV. Volume 1. (2005)
5. Jebara, T., Song, Y., Thadani, K.: Spectral clustering and embedding with hidden markov models. In Kok, J.N., Koronacki, J., de Mántaras, R.L., Matwin, S., Mladenic, D., Skowron, A., eds.: ECML. Volume 4701 of Lecture Notes in Computer Science., Springer (2007) 164–175
6. Liao, T.W.: Clustering of time series data: A survey. Pattern Recognition **38**(11) (2005)
7. J C Niebles, H Wang, L.F.F.: Unsupervised learning of human action categories using spatial-temporal words. In: Proc. of BMVC. (2006)
8. Makris, D., Ellis, T.: Learning semantic scene models from observing activity in visual surveillance. IEEE Trans. on Sys. Man and Cyb. - Part B' **35**(3) (2005)
9. Johnson, N., Hogg, D.: Learning the distribution of object trajectories for event recognition. IVC **14**(8) (1996)
10. Pittore, M., Campani, M., Verri, A.: Learning to recognize visual dynamic events from examples. IJCV (2000)
11. Lin, W., Orgun, M.A., Williams, G.J.: Temporal data mining using multilevel-local polynominal models. In: Int. Conf. IDEAL. Volume 1983 of Lecture Notes in Computer Science. (2000)
12. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE **77** (1989) 257–286
13. Chung, O.: Spectral graph theory (reprinted with corrections). In: CBMS: Conference Board of the Mathematical Sciences, Regional Conference Series. (1997)
14. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: NIPS 14. (2002)
15. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. on PAMI **22**(8) (2000) 888–905
16. Taylor, J.S., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press (2004)
17. Jebara, T., Kondor, R.I., Howard, A.: Probability product kernels. Journal of Machine Learning Research **5** (2004) 819–844
18. Leslie, C., Eskin, E., Noble, W.: The spectrum kernel: a string kernel for svm protein classification. Bioinformatics **1**(1) (2003)
19. Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., Watkins, C.: text classification using string kernels. JMLR **2** (2002)
20. Camastra, F., Verri, A.: A novel kernel method for clustering. IEEE Trans. on PAMI **27**(5) (2005) 801–804