Video Event Segmentation and Visualisation in Non-linear Subspace

Ioannis Tziakos [a] Andrea Cavallaro [a] Li-Qun Xu [b]

[a] *Multimedia and Vision Group, Queen Mary, University of London, UK*
[b] *BT Research & Venturing, British Telecommunications Plc, UK*

Abstract

We introduce the use of dimensionality reduction for video event detection without explicitly using motion estimation or object tracking. Raw data from video sequences are used to construct a low dimensional mapping representing the input frames. We compare Principal Component Analysis, Multidimensional Scaling, Isomap, Maximum Variance Unfolding and Laplacian Eigenmaps and implement an approach based on local, non-linear dimensionality reduction. We propose an approach with a graph based on the similarity of frames and enriched with the temporal information from the sequence processed by Laplacian Eigenmaps. This makes it possible to visualise the manifold of motion in the scene and to detect unusual events in a low dimensional space. We demonstrate the approach on standard traffic surveillance test sequences.

Key words: unusual event detection, dimensionality reduction, laplacian eigenmaps

1. Introduction

Continuous remote monitoring for automated scene understanding is becoming an increasingly studied problem for its expected benefits in applications such as proactive telecare (Skubic, 2005), distributed surveillance (Valera and Velastin, 2005) and, in general, ambient intelligence. Such systems can provide real-time alarms and warnings for security or medical personnel to act and intervene. In the long term, statistical information gathered can prove useful to redesign safety procedures and improve services.

Automatic video surveillance systems aim to provide reliable answers to questions, like how many and what kind of objects are in the scene and what actions are performed. An important goal for automated scene understanding is to identify unusual events. An unusual event corresponds to a set of subsequent frames where an action (motion) produces a deviation from a known pattern. A simple taxonomy of frameworks to detect unusual events could divide them into on-line and off-line approaches. On-line approaches process incoming streams of video. Where as off-line approaches process the entire video sequence (batch processing). Off-line (unusual) event detection is important for training data preparation, sequence segmentation and video indexing.

Email addresses: ioannis.tziakos@elec.qmul.ac.uk (Ioannis Tziakos), andrea.cavallaro@elec.qmul.ac.uk (Andrea Cavallaro), li-qun.xu@bt.com (Li-Qun Xu).

In this paper we show that events can be visualised and unusual events can be identified in low dimensional projections of the video frames. The proposed approach is based only on the raw video frame data from the camera using a dimensionality reduction algorithm, Laplacian Eigenmaps (Belkin and Niyogi, 2003), adapted to include temporal information. The main contribution of this work is that we do not need object extraction and tracking. Moreover, temporal information is incorporated in the neighbouring graph of frames with an elegant neighbourhood thresholding scheme.

The rest of the paper is organised as follows. Section 2 presents previous work on unusual event detection and Section 3 describes the proposed approach. Experimental results are shown in Section 4. Finally, conclusions and future work are described in Section 5.

2. Related Work

Event detection is mainly performed in two stages, feature extraction and event classification. The initial stage uses either features: object based and non-object based. The first consists in extracting features of moving objects from the video sequence such as colour histograms or trajectories. The second relies on video content descriptors like ensemble of patches and frame similarity. The extracted information is used in the final stage to classify the current actions in the scene as usual or unusual using Hidden

Markov Models (HMM), neural networks, graphs and spectral clustering methods.

Several approaches address this challenge based on features retrieved from an object tracking module. The trajectories from the object tracker are used by the event detection module that categorises them into "usual" and "unusual" trajectories. A set of neural networks can be used to achieve this goal by performing vector quantisation (VQ) to define prototypes that approximate the spatial and temporal distributions of the trajectories in sequence (Johnson and Hogg, 1996) or in parallel (Mecocci et al., 2003). After training, the new trajectories are characterised as unusual based on the learnt probability densities. A different approach uses tree structures to represent trajectories (Piciarelli and Foresti, 2006). Each node describes a cluster of similar partial paths. Unusual events are detected based on statistical analysis of the path the object follows through these trees. Hybrid approaches (Porikli and Haga, 2004), use Hidden Markov Models (HMM) followed by spectral clustering to classify patterns of trajectories and discover unusual events based on conformity scores.

The above mentioned techniques are hindered by the performance of the object detection and object tracking modules, thus being sensitive to scene changes (illumination variations, rain and wind) that produce clutters which ultimately lead to false detection and misclassification. Nevertheless, they have a real-time or close to real-time performance.

Zhong et al. (2004) extract object features from video segments in a high-dimensional space and co-embeds them with prototypes in a low-dimensional space to discover patterns of events in an off-line process. Zhang et al. (2005) investigate the use of a semi-supervised iterative adapted HMM to cluster data as common and uncommon. The algorithm is tested both on video and (synthetic) audio sequences. Xiang and Gong (2005) propose a relevance learning algorithm to cluster features representing video segments. Multi-observation HMMs are trained on these classes to detect patterns of behaviour. Other approaches detect foreground objects in the scene and convert them to graph representations of actions with techniques similar to those used in document analysis. For simple scenes in a stationary indoor environment, Hamid et al. (2006) convert moving object detections into event-motifs and classify actions as usual and unusual using spectral clustering methods.

There are a few approaches that do not rely on object features. Ensembles of patches, which provide a relaxed similarity measure between actions in video sequence and actions in a database are used for unusual event detection (Boiman and Irani, 2005). This is an on-line approach that combines a space-time video descriptor with heuristic database search techniques.

A summary of the methods described above is presented in Table 1. The methods are grouped based on the features used to detect unusual events and by the type of detection algorithm used.

Table 1
Summary of Unusual event detection methods

Features	Approach	Reference
Trajectories	Neural Net	Johnson and Hogg (1996)
		Mecocci et al. (2003)
	HMM	Porikli and Haga (2004)
	Trees	Piciarelli and Foresti (2006)
Objects	Spectral	Zhong et al. (2004) *
	HMM	Zhang et al. (2005)
		Xiang and Gong (2005)
	Graphs	Hamid et al. (2006)
Patches	Database	Boiman and Irani (2005)

(*) Off-line algorithm

3. Proposed approach

Dimensionality reduction techniques applied directly to video frames usually aim to create a shot summary of the sequence for video clustering and indexing applications (Xu and Luo, 2007; Li et al., 2006). Unlike commercial movies, the sequences acquired from static CCTV cameras do not present scene cuts and rapid changes in camera view point or scenery. This generally results in a smooth change of motion in the frames and as a consequence there exists a smooth manifold of motion over time.

The novelty of this work is the use of manifold learning algorithms, like Laplacian Eigenmaps, to discover the abstract manifold of motion in the scene, without performing object detection and tracking. We aim to acquire a summary of the video sequence in order to understand the action patterns and thus detect those that correspond to interesting events. In general, the motion in a video sequence will lie close to a low dimensional non-linear manifold. If the scene or region of interest (ROI) has the view of a single object, then the manifold will describe its motion. If the scene includes more than one object, then the discovered manifold will approximate the general flow of movement/change in the scene over time. With Laplacian Eigenmaps we unfold the manifold of motion in the scene and we expect to generate a representation of the events and also provide clues about unusual and uncommon patterns of frames.

3.1. *Laplacian Eigenmaps*

Graph dimensionality reduction algorithms rely on metrics defined on a neighbourhood graph. Based on these metrics and under certain constraints, the mapping is produced by solving an eigenvalue problem to find the solution that minimises the projection error. For example, Isomap (Tenenbaum et al., 2000) is an extension to multidimensional Scaling (MDS) where the distance metric is the geodesic distance defined on a neighbouring graph constructed from the input data. Using this metric, Isomap can

achieve adequate results to learn (unfold) manifolds that have a non-linear global structure. Another global dimensionality reduction algorithm is Maximum Variance Unfolding (MVU) (Weinberger and Saul, 2006) (previously known as Semi-Definite Embedding), which solves an optimisation problem that maximises the distance between the nodes in a neighbouring graph while preserving the distances along the edges and the angles between the edges. This optimisation problem is solved using Semi-definite Programming techniques.

Locally Linear Embedding (LLE) (Roweis and Saul, 2000) and Laplacian Eigenmaps (LE) (Belkin and Niyogi, 2003) use similar graphs to embed data accounting for the local data structure around each point in the high dimensional space. Specifically, LE is based on the commute times between the graph nodes, which takes into account all the paths from one node to another, and not just the shortest path, thus preserving local structure.

In the following, to improve readability of the description we consider the terms vector and node of a graph as alternative representations of the same entity, i.e. the multivariate observation acquired from the sequence of frames. Special notice is provided when these terms are not any more compatible.

We also assume that the graph has the following characteristics: (i) it is *connected*, which guarantees that there is always a path (not necessarily a direct connection) to travel between all the possible node pairs; (ii) it is *undirected*, which guarantees that the representing matrix is symmetric; (iii) the graph-matrix is semi-definite, so that the eigenvalues are also composed of real positive values.

We exploit the non-linear manifold learning locality preserving characteristics of LE. The LE-technique is an application of spectral graph theory for the Graph Laplacian. Given a set of N multivariate observations embedded as vectors $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$ in $\mathbb{R}^l (l \gg 1)$, a weighted graph $\boldsymbol{G}$ is built over the endpoints of these vectors. It consists of N nodes, one for each point and a set of edges connecting neighbouring points. Consider the problem of mapping the weighted graph $\boldsymbol{G}$ to a map of m dimensions so that connected points stay as close as possible. If two points are close enough, then there is an edge between them. Let $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$ be such a map. A reasonable criterion for choosing a good map is to minimise the following objective function:

$$\sum_{ij} \left\| \boldsymbol{y}_i - \boldsymbol{y}_j \right\|^2 W_{ij} \text{ with } i, j = \{1, \ldots, N\} \tag{1}$$

where $\boldsymbol{W}$ is the weight matrix defined as follows:

$$W_{ij} = \begin{cases} \exp \frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2} & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \text{ are connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The scale parameter σ is a free parameter that defines the importance of the neighbouring points. The objective function (Eq.(1)) with our choice of weights W_{ij} incurs a heavy penalty if neighbouring points $\boldsymbol{x}_i, \boldsymbol{x}_j$ are mapped far apart. It turns out that the minimisation problem reduces to finding:

$$\boldsymbol{Y}_{opt} = \underset{\boldsymbol{Y}^T \boldsymbol{D} \boldsymbol{Y} = \boldsymbol{I}}{\operatorname{argmin}} \; tr(\boldsymbol{Y}^T \boldsymbol{L} \boldsymbol{Y})$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$ is the Graph Laplacian matrix. $\boldsymbol{D}$ is the diagonal weight matrix such that its entries are column (or row, since W is symmetric) sums of $\boldsymbol{W}$, $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ij}$. Standard methods show that the solution is provided by the matrix of eigenvectors corresponding to the lowest, non-zero, eigenvalues of the generalised eigenvalue problem $\boldsymbol{L}\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}$.

The procedure to perform LE is formally stated below:

(i) Create the neighbouring graph matrix $\boldsymbol{G}$ from the multidimensional vectors $\boldsymbol{x}_i$.

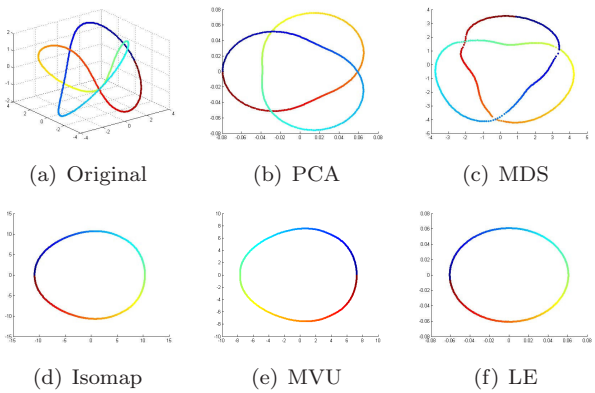
(ii) Compute the combinatorial graph Laplacian $\boldsymbol{L}$.

(iii) Solve the generalised eigenvalue problem of the graph Laplacian

$$\boldsymbol{L}\boldsymbol{y} = \lambda \boldsymbol{D}\boldsymbol{y}. \tag{3}$$

(iv) Embed into m-dimensional space using the m eigenvectors in ascending order of eigenvalues, starting from the first non-zero eigenvalue:

$$\begin{gathered} \text{with } \lambda_0 = 0 < \lambda_1 < \lambda_2 < \ldots < \lambda_m \\ \boldsymbol{x}_i \rightarrow (\boldsymbol{y}_1(i), \boldsymbol{y}_2(i), \ldots, \boldsymbol{y}_m(i)) \end{gathered} \tag{4}$$



(a) Original (b) PCA (c) MDS

(d) Isomap (e) MVU (f) LE

Fig. 1. Examples of dimensionality reduction of the trefoil manifold (a) using linear (b,c), global- nonlinear (d,e) and local-nonlinear (f) dimensionality reduction algorithms

An example of dimensionality reduction is presented in Figure 1. The data consist of 539 points (Weinberger and Saul, 2006) sampled from a trefoil knot in three dimensions. In this case, the underlying manifold is a one-dimensional curve. Due to the fact that the loop is closed, it can only be represented in a two-dimensional space (by a two-dimensional closed loop). The application of LE to this manifold produces the correct projection. The graph was created by the k-nearest neighbours rules (two nearest neighbours) and $\sigma \rightarrow \infty$. The same graph was used for Isomap and MVU, which also perform adequately. On the other hand, PCA and MDS, fail to find the internal structure of the data.

Algorithm 1 minimum k-nearest neighbours

procedure MINKNN($data$) ▷ Find connected graph of the data with the minimum k
  $k \leftarrow 1$ ▷ Initial number of neighbours
  $g \leftarrow k$-nn$(data, k)$ ▷ Create graph
  while $g \neq connected$ **do** ▷ Loop while unconnected
    $k \leftarrow k + 1$ ▷ Increase neighbours
    $g_{previous} \leftarrow g$ ▷ Save previous graph
    $g \leftarrow k$-nn$(data, k)$ ▷ Create new graph
  end while
  $g \leftarrow g_{previous}$ ▷ Restore the last connected graph
  return g ▷ Return the graph
end procedure

3.2. *Neighbour graphs*

The neighbouring graph is crucial to the success of the manifold learning process. The graph neighbour parameters define what is considered "local". An inappropriate selection of these values produces a distorted embedding. In general, a smaller number of neighbours better represents the local structure. However, the solution becomes more sensitive to the selected weighting scheme.

The most commonly used neighbouring graphs are the k-nearest neighbours. The Laplacian Eigenmaps formalisation is not restricted to these type of relational graphs only. Due to its simple definition and effective approximation of the manifold structure, they are usually preferred over other more complex graphs, e.g. Delaunay Graphs.

The k-nearest neighbour graph is based on the rule that each node is connected to at least k neighbouring (closest) nodes sorted by a similarity measure, usually the Euclidean distance between vectors. As a set of connection rules, node i is connected to node j if node j is among the k closest neighbours of i or node i is among the k closest neighbours of j.

The main advantages are that it usually provides a connected graph and low average number of connections per node, which gives a very sparse matrix such that the numerical eigensolver executes faster. To automatically choose the number of neighbours we follow the iterative process (Algorithm 1) that provides a connected graph with the minimum possible k.

The graph creation completes with the selection of weights. The simple scheme is to use a binary representation and to assign "1" where there is an edge between two nodes and "0" otherwise. We can achieve this result by setting $\sigma \to \infty$ in Equation (2). In this way it preserves the general information about the local structure in the proximity of each vector, but discards the relative importance between them. By setting the value of $\sigma \in (0, \infty)$, we can scale the influence that neighbouring (connected) nodes have among them. In theory, the latter weighting strategy holds more information about the manifold. Both weighting schemes produce similar results when the number of nodes increases. However, the latter needs careful selection of the scale parameter σ to avoid errors in the numerical solution of the generalised eigenvalue problem. For our work we use the binary scheme or the σ value selected as:

$$\sigma = \frac{\sum_{ij} G_{ij}}{N_v}, \tag{5}$$

where N_v is the number of vertexes in the graph $\boldsymbol{G}$.

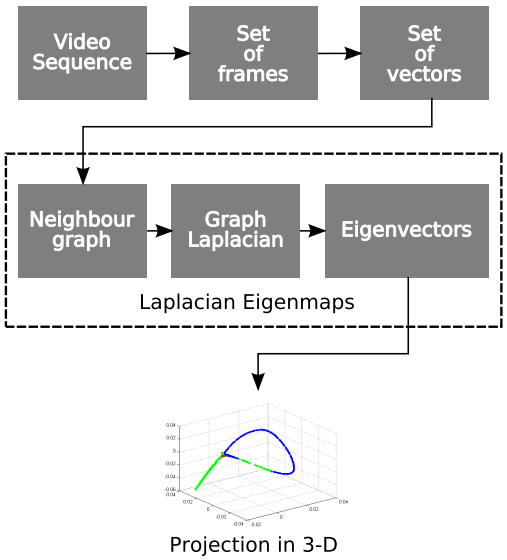


Fig. 2. Block diagram of the proposed approach

3.3. *Algorithmic steps*

The proposed framework to visualise a video sequence in a low dimensional space is shown in Figure 2. Given a video sequence, we extract the raw image frames and then convert each frame into a high-dimensional feature vector. This is performed by concatenating each pixel of the frame as a new dimension in the vector (i.e, a frame of 100x100 pixels will reshape to a vector of 10000 dimensions if the original frame is grey-scale or 30000 dimensions if the original is colour RGB). The above step results in a set of vectors whose number is equal to the number of frames in the sequence.

Laplacian Eigenmaps is applied to this set of vectors and provides a mapping to a 3-dimensional space. The intermediate steps are also presented in Figure 2. From left to right, we create the neighbouring graph and then compute the graph Laplacian. Finally we solve the generalised eigenvalue problem (Eq. (3)) to find the first three, smallest non-zero eigenvalues and their corresponding eigenvectors. The concluding step is to use these eigenvectors and create a new set of endpoints in a three-dimensional space (Eq. (4)). Each vector represents a frame in the original video sequence. This visualisation of reduced dimensionality provides us with cues about the characteristics of the entire video sequence and an abstract description of the events.

3.4. *Frame similarity*

In order to construct the neighbouring graphs and to establish a similarity measure between the high-dimensional

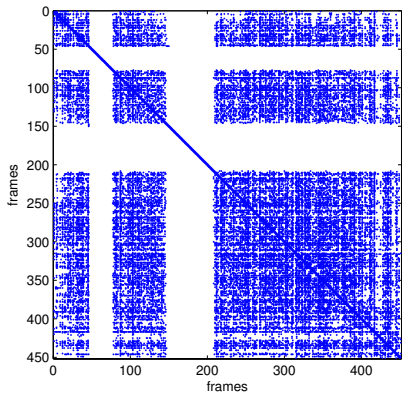


Fig. 3. A sample matrix of a graph constructed with the temporal ε-graph rules. Nodes (frames) are divided into *popular* and *lonely*. The regions of low connectivity (frames: 45-75 and 146-207) correspond to frames with slow changing actions in the sequence

feature vectors (frames), we selected the squared Euclidean distance. This index of similarity provides reasonable results without increasing the complexity of the overall algorithm. Nevertheless a good amount of spatial information is lost since the vector distance provides only an average index of the total distortion, due to motion, between two frames. This loss is acceptable since we are only interested in an abstract description of the scene and do not aim toward objects detection explicitly.

3.5. *Temporal ε-graph*

To further enhance the graph representation we add prior knowledge about the video sequences that come from stationary surveillance cameras. Hence there is consistency (similarity) between subsequent frames.

There are two ways to incorporate prior knowledge into the graph. It is possible to add the information as new dimension to the input vectors and weight accordingly to increase or decrease the influence in the total similarity measure. A fine example of such an approach can be found in *normalized cuts* for image segmentation (Shi and Malik, 2000) where the spatial information is combined with the colour information of the pixels for the graph generation.

There is also a more elegant and simpler way to achieve the same effect without introducing another free parameter to the framework. We can use the intra-frame relation to threshold the local neighbourhood of the nodes. As a result, the graph is formed in compliance to the following connection rules. Given a video sequence of N frames and their conversion to N vectors, vector $\boldsymbol{x}_i$ is always connected to the next vector $\boldsymbol{x}_{i+1}$. For the rest of the vectors, $\boldsymbol{x}_j$, where $j \neq \{i, (i+1)\}$, we threshold the radius of allowed connections based on the rule $d(\boldsymbol{x}_j, \boldsymbol{x}_i) \leq d(\boldsymbol{x}_{i+1}, \boldsymbol{x}_i)$, where d is the similarity measure.

This graph gives a strong affinity between subsequent vectors. The nodes of the graph are divided into *popular* and *lonely* nodes (Fig. 3). Frames nodes that have a large difference from adjacent frames, have their neighbouring threshold relaxed *popular*. Frames with small differences in subsequent frames are hardly connected to any other nodes *lonely*, except the previous and next frame. These rules will make slow changing actions to be considered outliers and thus stand out in the projection.

Table 2
Data-set description

Seq.	Number of frames	ROI Size (pixels)	Events	Frames
SEQA	451	54x51	Car passing	45-75
			Man crossing	146-207
SEQB	6000	54x51	Car reversing	762-881
			Car entering highway	1917-1961
			Man crossing	3303-3353
SEQC	2900	54x51	Man crossing	211-261
			Car crossing (synthetic)	415-496
			Man crossing	2194-2246
SEQD	1057	54x51	Car using auxiliary lane	173-848

Table 3
Parameter variations used for LE in 3D projections

Set	Graph	Weight
LE-C	k-nn	binary
LE-CW	k-nn	Eq. (5)
LE-CT	temporal	binary
LE-CTW	temporal	Eq. (5)

4. Experimental results

In this section we compare the results obtained with state-of-the-art dimensionality reduction algorithms on the highway surveillance videos from the MPEG-7 data-set. These videos have a variety of moving vehicles (cars, vans and trucks): Figure 4 shows samples from the region of interests (ROIs), whereas Table 2 summarises the contents of the sequences.

SEQA (Fig. 4: row 1) covers a small area of the scene, that is comparable to the size of an object. SEQB (Fig. 4: row 2) has three unusual events, namely a car reversing in the auxiliary lane, the same car moving forward in the auxiliary lane and a man crossing the road. SEQC (Fig. 4: row 3) contains two instances of the scene of a man crossing the road. The third unusual event is generated with an artificial scene of a car that is moving from left to right. The car has been scaled to 70% of the original size to produce less distortion, thus creating a slow moving artefact scene similar to the one of the pedestrian. Finally, SEQD (Fig. 4: row 4) contains multiple objects including a car using the auxiliary lane before it returns to the highway.

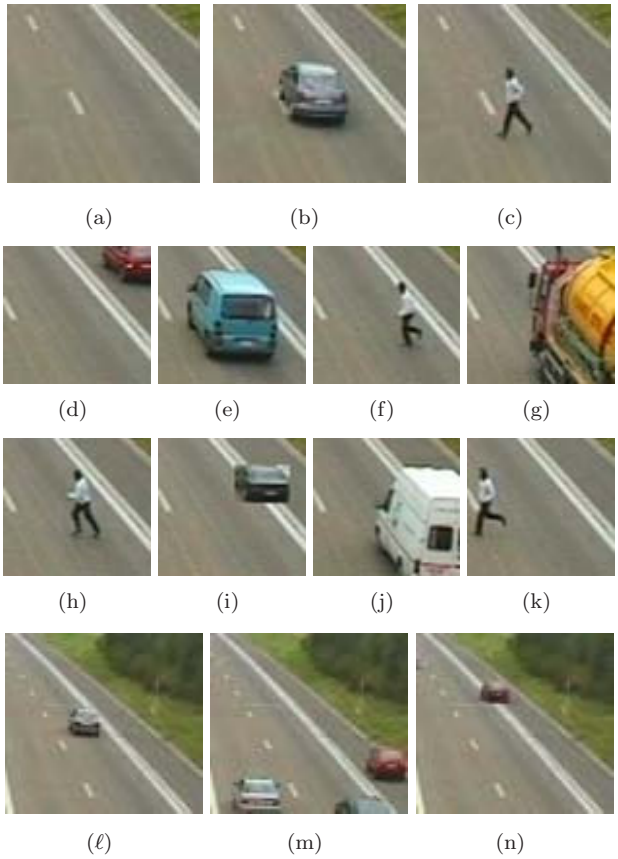


Fig. 4. **SEQA:** (a) Null event (frame 12) , (b) Car passing (frame 56) and (c) man crossing (frame 175). **SEQB:** (d) Car reversing (frame 808), (e) Blue van passing (frame 1348), (f) Man crossing (frame 3323) and (g) Truck passing (frame 5357). **SEQC:** (h) Man crossing (frame 235), (i) Car crossing "sideways" (frame 440), (j) White van passing (frame 912) and (k) Man crossing (frame 2235). **SEQD:** (l) Car passing (frame 1), (m) Car parked on the auxiliary lane (frame 220) and (n) Car leaving the auxiliary lane (frame 795)

The projection results are compared with PCA, MDS and Isomap along with variations (Table 3) of LE. For the nearest neighbour graph in Isomap and MVU, we used the same graph created as input to the LE-C variation (other parameters are set to default values). MVU was used in combination with the CSDP solver (Borchers, 1997). The projections for SEQB and SEQC were acquired using the alternative incremental approach of MVU. For PCA and MDS we defined the number of dimensions to three.

The events are colour coded in all the plots. The events of low interest are coloured with the same colour as the null event frames, except in SEQA where for clarity both motion actions (car and man) have different colouring.

4.1. *Event projection*

When the region of interest has comparable size to the objects, then the projections are expected to hold information about their actions. Sequence SEQA contains two events. All projections (Fig. 5) are expected to hold meaningful information about these events. The PCA projection

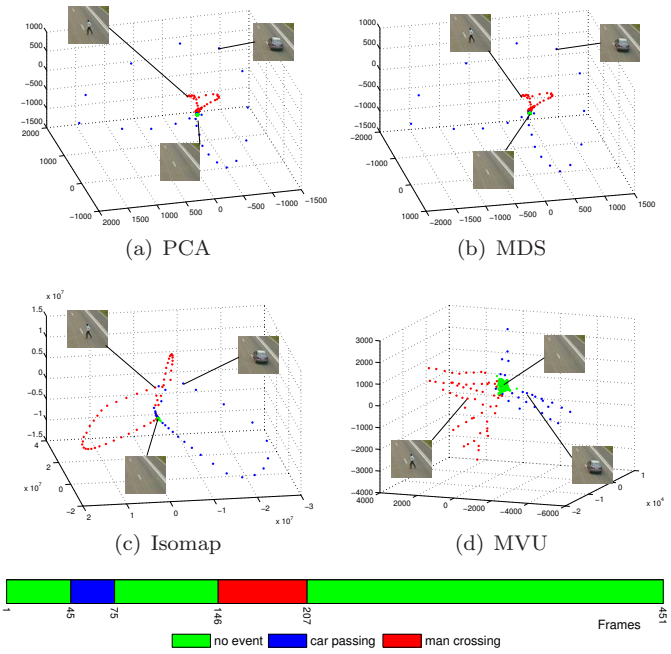


Fig. 5. Comparison of dimensionality reduction algorithms applied on the video sequence SEQA using colour frames

separates the two events. The man generates a small loop inside the bigger loop of the car. MDS also provides a similar embedding. Isomap results in an projection where different events are mapped in different loops of points. MVU projects the two events in opposite directions but the path that the objects follow in this space is not as clear as in the case of the other algorithms. Changes in angle of the trace of the man correspond to changes in the shape (size) of the man while he runs across the highway.

When we compare with the LE variations (Fig. 6), Isomap and LE-C generate projections with similar appearance. The events are placed in loops that start and end in the *null event*. The weighted LE-WC variation mapping has a topology similar to the PCA projection, but reveals additional interesting features. The pedestrian performs a periodical movement, while crossing the road from right to left and periodically change his shape and size, thus the sequential frames have maxima and minima in distance between them. In this projection the periodic movement of the man generates angles like in MVU and density differences.

The LE embedding gives a more natural explanation of the scene. If the temporal information (LE-CT) is included in the graph, then the two events are better separated. Since both events have a different scale of distances between sequential frames, they are not connected by the temporal ε-graph (Fig. 3). Additionally when the weighting changes to the average scheme, LE-CTW, the car frames collapse to a small area close to the background frame, while the *pedestrian* frames generate a loop with density and angle changes.

This ability of LE to separate the two events in the se-

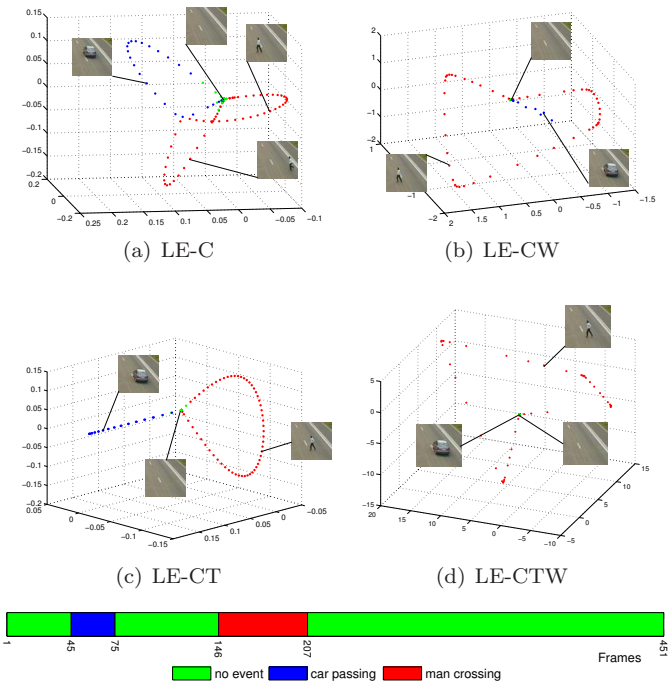


Fig. 6. Comparison of four LE variations on the video sequence SEQA using colour frames

quence (the car and the man) justifies our choice in a manifold learning technique to visualise the events on the video sequence. Although the other dimensionality algorithms perform well on this simple scenario, LE combines their distinct characterises under a single framework. It is true that PCA (Fig. 5(a)) provides a descriptive summary of the actions in the video. Isomap while discovering the underlying manifold forces each action to project in a distinct loop. LE gives a fusion of these properties in various degrees depending on the graph and weighting schemes used, from the *Isomap* style (Fig. 6(a)) to the combination of *PCA* and *MVU* style projection (Fig. 6(b)) and even further to extreme and *selective* projections (Fig. 6(c and d)) using the temporal ε-graph.

4.2. *Unusual event separation*

To further explore the advantages and limitations of LE as a means to acquire frame projections for event detection, we apply the LE variations to SEQB and SEQC.

Figure 7 compares the dimensionality reduction algorithms against a longer and more complex video, SEQB. The linear dimensionality reduction algorithms (PCA and MDS) fail to understand the local nonlinear structure of the frame manifold as they use only the variance or distance information in the high dimensional space. The internal structure is also not discovered by Isomap, while only MVU is able to project the two highlighted events involing the car away from the main *null events*.

When LE-C (Fig. 7(e)) is used, it is possible also to notice the loops that describe the car (green and light blue dots) but not the man crossing the road. The remaining frames are distributed on a triangular surface. The lower corner holds the frames that show a car (or truck) in the centre of the frame. The two top corners correspond to frames under different illumination conditions and some small movement of the camera. The area in between is occupied by the frames where the cars move in or move out of the view. The trace associated with each vehicle follows similar path: It starts at the top corner, moves down to the low corner and returns to the original position when it is out of the view.

The temporal ε-graph embedding (Fig. 7(f)) changes the mapping in a drastic way. This behaviour is expected since the temporal ε-graph creation rules penalise only slow changing sequential frames. Both the man and the car in reverse events are slow, thus in the graph structure they are *lonely* nodes and embedded as outliers. But when the car is moving forward the movement is not slow and there are multiple objects in the scene. The average distortion of sequential frames is large and the nodes are *popular*, which results in high connectivity.

SEQC is another example that demonstrates the effect of the temporal ε-graph for slow changing scenes. In this case the sequence that combines an artificial and two real-life unusual events. The alternative projections (PCA, MDS, Isomap) cannot provide any visual clues about the highlighted events, while LE-C projection (Fig. 8(e)) shows the highlighted events in separate loops, with the loop of the man more easily identified. Since the sequence has usual events similar to SEQB, there are features that are present also in Figure 7(e and f). The usual events are visualised in a triangular shaped surface and the unusual events form a loop moving out of the surface. A large portion of this loop is still is very close to the cloud of *no event* frames. MVU provides a clearer distinction for the events. Clearly the "car crossing" event is easier to spot.

The results are improved by applying LE-CT (Fig. 8(e)). The two scenes of the man crossing the highway are mapped together, but now the points follow a straight line. The same applies to the synthetic event. Although these events are both slow, they are not embedded close to each other in the embedded space. In fact, the difference between them is large enough so that the frames are not connected in the graph. The LE-CT variation projects slow changing scenes into *lonely* points in the projections, but those points are not going to be close (Euclidean distance) to each other, unless they are also similar.

4.3. *Objective evaluation*

The existence of visible structures in the projections, as discussed in the previous subsection, provides us with clues about the characteristics of the motion/action in the scene. Another property is the separation of the highlighted events from the *no event* set. This can be evaluated by calculating the average squared Mahalanobis distance of these frames against the *no event* frames in the projection.

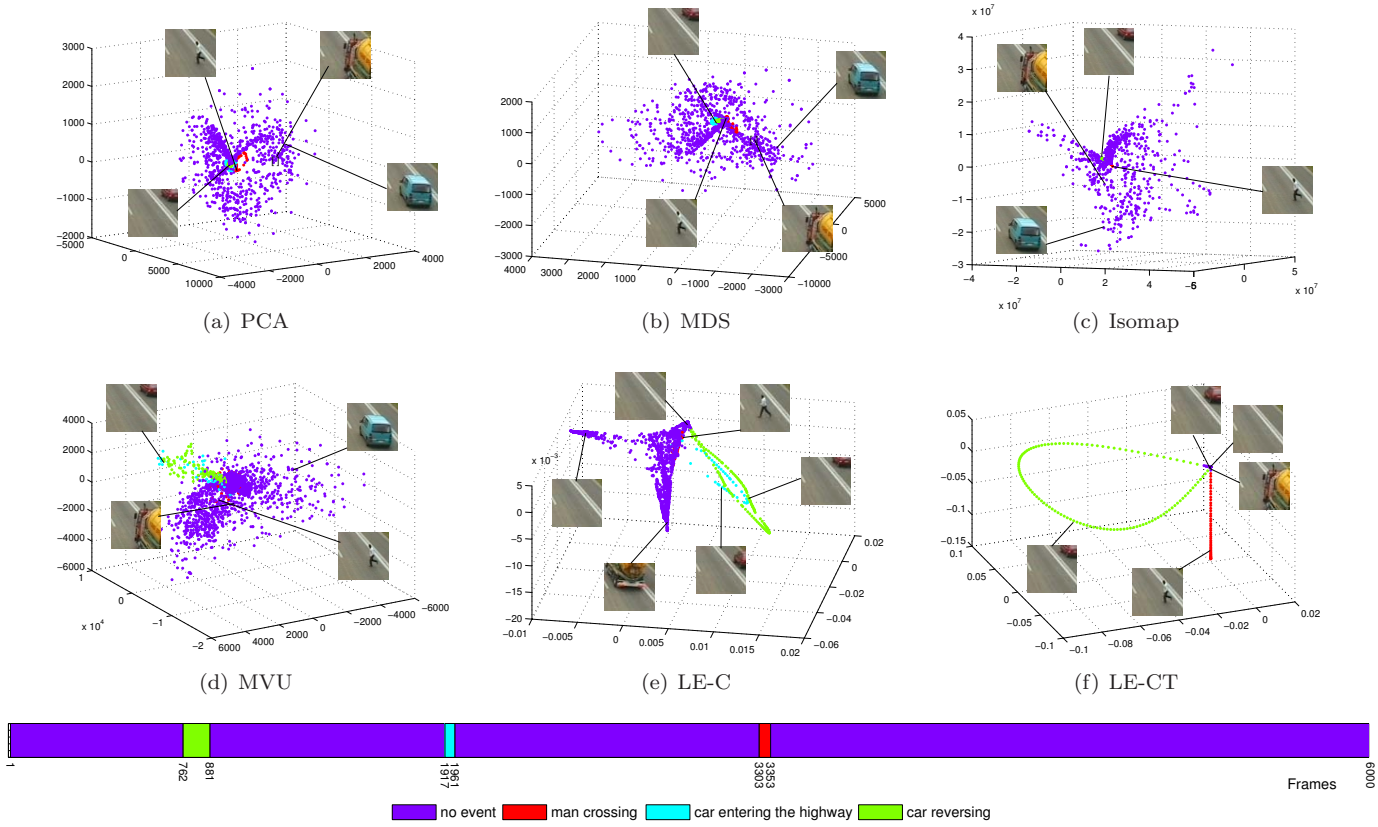


Fig. 7. Comparison of projections for the video sequence SEQB, using colour frames

Given the projections $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N\}$ of the N frame and the set $\boldsymbol{S}$ of the "no-event" frames, we compute the distance D_i of every highlighted frame $\boldsymbol{y}_i \notin \boldsymbol{S}$ as

$$D_i = (\boldsymbol{y}_i - \boldsymbol{\mu})^T \boldsymbol{P}^{-1} (\boldsymbol{y}_i - \boldsymbol{\mu}), \tag{6}$$

where $\boldsymbol{P}$ is the covariance matrix of $\boldsymbol{S}$ in the projected space and $\boldsymbol{\mu}$ is the corresponding mean. We compute the average frame distance of each highlighted event from $\boldsymbol{S}$. The quality score is computed as logarithm of the distance, so that negative values denote a poor separation (e.g., when interesting event frames are inside the cloud of the *no event* frames). Values close to zero account for event projections that have a good amount of overlapping with the *no event* set. Finally, values greater that zero correspond to projections where the highlighted events are further away from the *no event* frames. Score values larger than 1.50 are usually enough to visually identify the event in the projections.

Table 4 reports the event separation scores of the projections (PCA, MDS, Isomap, MVU, LE-C and LE-CT) for sequences SEQA, SEQB and SEQC. Note that since for LE-CW and LE-CTW the application of the weighting scheme results in an unstable solution of the eigenvalue problem for the longer sequences, we do not report their results. We observe that the projections of SEQA (Fig. 5 and 6) are equivalent in terms of separating the selected events, with the exception of LE-CT, which outperforms the other approaches achieving a score of 9.58 for the separation of the "man crossing". In SEQB, PCA, MDS and Isomap give poor results (below zero), while MVU is the only alternative dimensionality reduction algorithm that has positive scores for this sequence (good scores for the events involving the car). On the other hand, LE-C performs well (better than MVU) for the events with the car in the scene but the score for the man event is still low (i.e., it will not be easily visible). Similarly LE-CT has good results for two out of three uncommon events in the sequence. The last set of scores describe the projections of SEQC. In this case, PCA and MDS fail to provide good scores, especially for the synthetic event. Isomap improves the results by having

Table 4
Separation scores for projections on SEQA, SEQB and SEQC

	SEQA		SEQB			SEQC	
Algorithm	man crossing	car passing	man crossing	car reversing	car entering highway	man crossing	car crossing
PCA	3.31	4.46	−0.20	−0.93	−0.40	−0.25	0.45
MDS	3.31	4.46	−0.20	−0.93	−0.40	−0.25	0.45
Isomap	2.45	4.15	−0.98	−0.52	−0.46	0.16	0.57
MVU	4.62	4.19	0.57	2.50	2.44	0.51	1.75
LE-C	4.44	3.42	1.15	4.48	4.14	0.43	0.79
LE-CT	9.58	4.13	6.38	7.32	−0.39	2.62	8.05

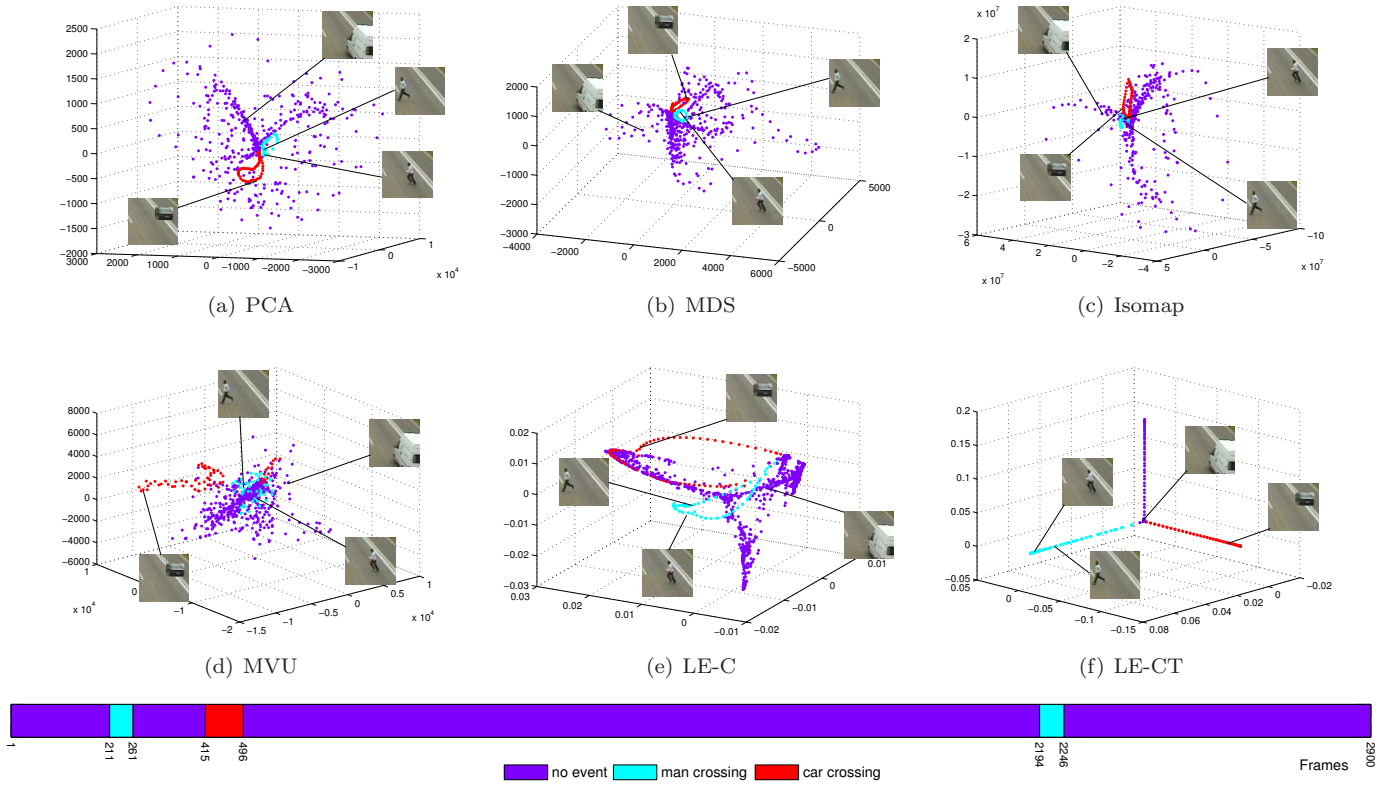


Fig. 8. Comparison of projections for the video sequence SEQC, using colour frames

a positive score for that event. MVU performs better and achives score higher that LE-C. A large part of the traces of the projected events is very close to the *no event* frames, a fact that can be verified also by the visual inspection of the LE-C projection. Nevertheless LE-CT provides the best results for both events in this sequence.

To summarize, the objective evaluation (Table 4) confirms that Laplacian Eigenmaps is more suitable for this application and can provide clues about the interesting events in the sequence, as previously discussed based on visual evaluation of the projections.

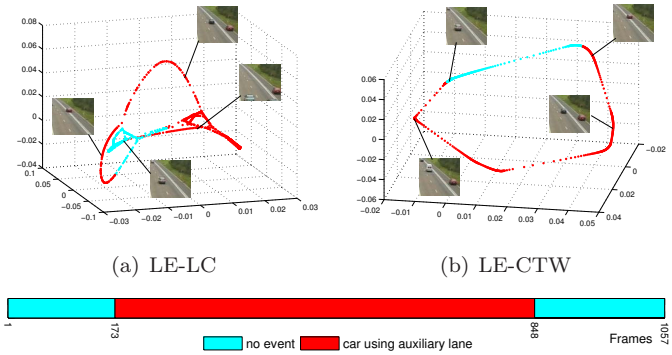


Fig. 9. Comparison of LE-C and LE-CTW on a multiple object scene (SEQD)

4.4. *Multiple objects in the ROI*

When the scene involves several moving objects we expect the produced three-dimensional visualisation to provide an abstract summary of the total motion flow and not that of individual objects. SEQD presents such a case. Every movement in the video creates a new path in the projected space (Fig. 9(a)). These paths cross each other in various places corresponding to frames with similar content.

When the weighted temporal ε-graph (LE-CTW) is used, the complete sequence is projected as one loop (Fig. 9(b)). The successive events of the car passing appear in the sequence one after the other without *null event* frames to separate them. Each moving object generates a spatio-temporal difference in the sequential frames. Since neighbouring graphs, k-nearest neighbours and ε-nearest neighbours are created based on the total Euclidean distance between frames, they provide a manifold of this index. The projection improves by adding the temporal information into the graph construction phase. The slow changing frames form a separate chain that is mapped as a loop or a line. The *popular*, fast changing frames have more nodes connected to them, since they enjoy a relaxed (larger) threshold. This is the effect of the dynamic threshold imposed by the temporal ε-graph rules.

The forced temporal connection leads to concatenating the events together. Thus we are able to visualise the repetitive nature of the general motion. The points of higher den-

sity correspond to the frames where a car is far and moves slowly out of the scene. The frames where a new car appears in the scene are placed on the sparse areas of the loop.

5. Conclusions and Future work

We proposed an unsupervised framework based on LE to produce a three dimensional visualisation of surveillance video sequences using dimensionality reduction. The proposed framework enables the separation of actions without the need of an object detector or object tracker. We compared the performance of five standard dimensionality reduction approaches using visually inspection as well as an objective score on a standard Highway surveillance data-set. We showed that the proposed approach based on LE is able to map the video sequence in a low-dimensional space such that the general characteristics of motion can be preserved. The framework is also capable to provide cues about the existence of statistically unusual events in the sequence and outperforms alternative dimensionality reduction algorithms in this task. Furthermore LE was naturally extended to take into account temporal information, thus giving improved results in the visualisations and the scores.

Our future work includes the exploration of more elaborate graph rules to apply in more complex scenarios and the definition of an unsupervised algorithm to analyse the projected mapping.

References

Belkin, M., Niyogi, P., Jun. 2003. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. Neural Computation 15 (6), 1373–1396.

Boiman, O., Irani, M., 2005. Detecting irregularities in images and in video. In: Proceedings of the Tenth IEEE International Conference on Computer Vision. Vol. 1. pp. 462–469.

Borchers, B., 1997. CSDP : A C library for semidefinite programming. Tech. rep., Socorro, NM, USA.

Hamid, R., Maddi, S., Bobick, A., Essa, I., 2006. Unsupervised analysis of activity sequences using event-motifs. In: Proceedings of the 4th ACM International Workshop on Video Surveillance and Sensor Networks. pp. 71–78.

Johnson, N., Hogg, D., Aug. 1996. Learning the distribution of object trajectories for event recognition. Image and Vision Computing 14 (8), 609–615.

Tenenbaum,J. B., de Silva, V., Langford, J. C., Dec. 2000. A Global Geometric Frame-work for Nonlinear Dimensionality Reduction. Science 290 (5500), 2319–2323.

Li, Z., Li, Z., Gao, L., Katsaggelos, A., 2006. Locally Embedded Linear Subspaces for Efficient Video Indexing and Retrieval. In: Proceedings of the IEEE International Conference on Multimedia and Expo. pp. 1765–1768.

Mecocci, A., Pannozzo, M., Fumarola, A., 29-31 July 2003. Automatic detection of anomalous behavioural events for advanced real-time video surveillance. In: Proceedings of the IEEE International Symposium on Computational Intelligence for Measurement Systems and Applications. pp. 187–192.

Piciarelli, C., Foresti, G., Nov. 2006. On-line trajectory clustering for anomalous events detection. Pattern Recognition Letters 27 (15), 1835–1842.

Porikli, F., Haga, T., June 2004. Event Detection by Eigenvector Decomposition Using Object and Frame Features. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshop. pp. 114–114.

Roweis, S. T., Saul, L. K., Dec. 2000. Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290 (5500), 2323–2326.

Shi, J., Malik, J., Aug. 2000. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8), 888–905.

Skubic, M., Nov. 2005. Assessing Mobility and Cognitive Problems in Elders. In: AAAI Fall Symposium, Workshop on Caring Machines: AI in Eldercare.

Valera, M., Velastin, S., 8 April 2005. Intelligent distributed surveillance systems: a review. In: IEE Proceedings on Vision, Image and Signal Processing. Vol. 152. pp. 192–204.

Weinberger, K. Q., Saul, L. K., 2006. Unsupervised learning of image manifolds by semidefinite programming. International Journal of Computer Vision 70 (1), 77–90.

Xiang, T., Gong, S., 15-16 Sept. 2005. Relevance learning for spectral clustering with applications on image segmentation and video behaviour profiling. In: Proceedings of the IEEE Conference on Advanced Video and Signal Based Surveillance. pp. 28–33.

Xu, L.-Q., Luo, B., 2007. Appearance-based video clustering in 2D locality preserving projection subspace. In: Proceedings of the 6th ACM international conference on Image and Video Retrieval. pp. 356–362.

Zhang, D., Gatica-Perez, D., Bengio, S., McCowan, I., 2005. Semi-Supervised Adapted HMMs for Unusual Event Detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 1. pp. 611–618.

Zhong, H., Shi, J., Visontai, M., 27 June-2 July 2004. Detecting unusual activity in video. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Vol. 2. pp. 819–826.