Event monitoring via local motion abnormality detection in non-linear subspace

Ioannis Tziakos[a,1], Andrea Cavallaro[a,1], Li-Qun Xu[b]

[a] *Queen Mary University of London, School of Electronic Engineering & Computer Science, Mile End Road, London E1 4NS, UK*
[b] *BT Research & Venturing, British Telecommunications Plc, Adastral Park, Ipswich IP5 3RE, UK*

Abstract

We present a computational approach to abnormal visual event detection, which is based on exploring and modeling local motion patterns in a non-linear subspace. We use motion vectors extracted over a Region of Interest (ROI) as features and a non-linear, graph-based manifold learning algorithm coupled with a supervised novelty classifier to label segments of a video sequence. Given a small sample of annotated normal motion vectors, the non-linear detector ranks segments in a sequence as a function of abnormality. We evaluate the proposed method and compare its performance against the use of other low-level features such pixel appearance and change detection masks. Our choice of feature space compares favorably to the alternatives in terms of classification performance, sensitivity to noise as well as computational complexity.

Key words: Abnormal video event detection, Manifold learning, Laplacian Eigenmaps, Novelty classifier

1. Introduction

Dynamic visual scene understanding is a key research area in computer vision and signal processing. Its aim is to filter high volumes of video data and to extract essential event and behavior descriptors characterizing the semantics in a scene. Subspace learning methods that have been used for document analysis and content-based image retrieval have been recently applied in video analysis domain to map the spatial-temporally redundant, high-dimensional video stream into a low-dimensional feature subspace while retaining the desired structural information.

In the field of video abnormality detection, one can model normal action patterns that take place in the scene through subspace learning and then discover abnormal instances that considerably deviate from the observed normal patterns. However, to address multi-object and unrestricted scene one needs to consider an appropriate mathematical modeling method that represents relevant information at the signal level. This signal-level representation and processing has to cope with the non-linearity, spatial locality of patterns and noise that is inherent of such high-volume video data. In this work we present a non-linear subspace learning detector to address the abnormality detection problem.

1.1. Background

Related video stream analysis work for scene understanding and abnormality detection is mainly performed in conjunction with an object tracking [13, 20, 19, 11, 12, 25]. These methods are generally hindered by factors such as initialization and scene clutter. Moreover, their application to crowded scenes is not straightforward as the computational complexity increases considerably with the number of objects while the performance degrades due to occlusions and observation ambiguities.

Sampling support	Features	Modeling	Reference
Pixel	Object detection	Density estimation	Ermis et al. [9]
Region	Appearance	Database search infinite HMM	Boiman and Irani [5] Pruteanu-Malinici and Carin [21]
	Object detection	pLSA	Li et al. [15]
	Motion vectors	Phase Locked Loop Probability distribution	Russell and Gong [23] Adam et al. [1]
Frame	Object detection	Graph co-clustering Probabilistic manifolds Multi-Observation HMM Coupled HMM PCA + SVM	Zhong et al. [34] Cui et al. [8] Xiang and Gong [31] Zhou and Kimber [35] Sudo et al. [26]
	Object + motion Motion vectors	Mixture of Gaussian HMM Semi-supervised adapted HMM	Andrade et al. [2] Zhang et al. [33]

Table 1: Abnormality detection using low-level features

In order to detect abnormal events in complex scenes (crowded, unrestricted scenarios) where object tracking is unreliable, recent attempts have been directed towards exploring or modeling directly the low-level features extracted from video streams. These frameworks resort to methods from other fields, such as document analysis, manifold learning, data mining, and data compression, to learn the normal action patterns in the scene so as to detect abnormal events.

The methods can be grouped according to the dimensions of the sampling support, the features and the mathematical modeling technique used (see Table 1). The sampling support defines the input that is used to infer a decision about the abnormality. The smallest sampling support is a single pixel. Ermis et al. [9] learn the statistical properties of each pixel as an independent sensor and detect abnormality based on its on/off (foreground/background status) patterns. Frame-based meth-

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)

[1] I. Tziakos and A. Cavallaro are with the Multimedia and Vision Group.

ods model normal behaviors of the full scene within the field of view of a camera. The scene descriptor information is commonly extracted based on moving objects detection (blobs), and the models are trained to detect unusual events. Li et al. [15] use document analysis techniques together with object detection to find abnormal actions taking place in large segmented regions of the scene.

Cui et al. [8] model action by probabilistic manifolds in a linear subspace of object-based features and utilize a particle filtering to detect abnormal events. Zhong et al. [34] extract object features and then cluster them to produce a dictionary of prototype classes. Features and prototypes are co-embedded in a low dimensional space, where spatially isolated clusters labeled unusual. Based on a linear subspace (PCA) of object features Zhang et al. [33] train semi-supervised adapted HMMs on the normal action patterns and detect abnormal ones. PCA is also utilized by Sudo et al. [26] and coupled with an SVM novelty classifier. Xiang and Gong [31] segment the video based on activity from object features and model patterns using Multi-Observation HMM on a subspace provided by spectral clustering. Abnormality is ranked based on conformity of the observed patterns to the learned models.

At the moment single pixels can only detect simple localized appearance patterns; thus are incapable of describing complex motion patterns from articulated objects which may span over a large region. Frame-based methods are better suited for such a task instead. However, the complex interactions between multiple objects (e.g. crowded scene) that move in an unrestricted real scene often create complex action patterns. Further, more using global frame features, localized abnormal patterns will be averaged among all the other actions taking place and thus difficult to detect.

Mid-sized regions offer a compromise in describing more complex patterns than single pixels, and because they contain fewer objects the complexity of observed patterns is tractable. Adam et al. [1] make inference on single pixel abnormality, but the decision taking account of its motion characteristics in the local neighborhood (region). Boiman and Irani [5] extract patches in grids overlaid on each frame to calculate local descriptors and compare against a database of normal patterns. Pruteanu-Malinici and Carin [21] apply Invariant Subspace Analysis (ISA) to appearance features and train infinite-HMMs for each block in a grid region. Russell and Gong [23] use a Phase Locked Loop (PLL) topology to learn the fundamental change frequency in blocks and detect abnormal changes.

Patterns in the selected feature spaces are normally modeled by probabilistic methods such as Hidden Markov Models. The high-dimensional spaces spanned by low-level features are usually impractical to model with such probabilistic methods because of training and complexity limitations, thus several methods have been proposed to incorporate dimensionality reduction techniques before pattern modeling to produce a more manageable subspace of features and avoid the curse of dimensionality. The majority of techniques are based on spectral clustering variations or linear methods such as PCA, which are incapable of representing the non-linear manifolds even in simple scenarios [29].

1.2. Contribution

The proposed method builds on the authors' previous work [29] which investigated the problem of representing features extracted from video frames and compared Laplacian Eigenmaps [3] with linear (PCA and MDS) and non-linear approaches [28, 30] in representing appearance-based features. Laplacian Eigenmaps (LE) proved to provide better projections (subspace) at representing the internal structure of the appearance-based patterns in the sequence.

This work extends the framework by introducing a novelty classifier trained in the non-linear subspace to detect abnormal events. We compare features of motion vectors with pixels and change detection masks based on detection performance, sensitivity to noise and computational complexity. Finally, we discuss results on a real-world surveillance video sequence of an underground train station and demonstrate the ability of the proposed approach in detecting a variety of motion patterns that deviate from the dominant regular motions.

Although manifold learning has been used for action recognition [16], the adaptation of such methods to the abnormality detection problem is not trivial. Methods that perform action recognition [32] are trained on datasets [24] that describe specific actions. However, the knowledge about a predefined set of action patterns is *not* available in abnormality detection. The lack of this knowledge also inhibits the use of supervised classification techniques that solve the "c-1" class problem [27]. In fact, in abnormality detection the concepts of what is *normal* and what is *abnormal* are inferred by the data itself based on frequency of occurrence.

The novelty of this work is to use graph based non-linear dimensionality reduction as the core module of the abnormality detection framework. In particular, we propose a self-tuned weighted graph combined with Laplacian Eigenmaps that allow to reduce the dimensionality and to separate normal from abnormal instances in one single step. The resulting abnormality detector *i*) detects abnormal events without prior knowledge of normal patterns, *ii*) does not require high-level information filtering steps (e.g., object tracking), *iii*) accounts for the non-linear correlation of the features, *iv*) is generic and can be used with a variety of low-level features.

While the proposed detector is not directly comparable with alternative proposals in related work [1] as it is currently an offline semi-supervised method, on a qualitative basis our detector is able to learn non-linear action patterns and work with a wide range of original dimensions and features. On the other hand, related works are restricted to use one dimensional inputs and this limitation restricts the type of events that can be detected.

The paper is organized as follows. Section 2 describes the proposed framework in detail, including the motion features, the graph-based manifold learning algorithm, the novelty classifier, and discuss the limitations. Section 3 discusses the experimental results on both a synthetic and real-world videos in a challenging video surveillance scenario. Finally, conclusions and future work are given in Section 4.

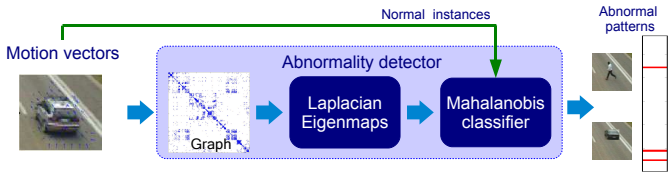


Figure 1: Diagram of the proposed abnormality detector. The motion vector space is mapped into a non-linear subspace using LE and the projection is labeled by the novelty classifier

2. Proposed approach

2.1. Preliminaries

Given a video sequence $\mathbf{S} = \{I_t : t = 1, \ldots, T\}$ and a Region of Interest (ROI) $R_t \subset I_t$, the goal is to detect abnormal instances R_a that deviate from the common action patterns in the set $\mathbf{A} = \{R_t : t = 1, \ldots, T\}$. We address this problem as a binary classification performed by the function f.

$$f : \mathbf{A} \rightarrow \{normal, abnormal\} \tag{1}$$

Ideally f should be able to learn normal patterns that take place in the ROI and detect abnormal instances without supervision. In practice, this is not feasible because of the complexity of the action patterns and the interaction between objects in the ROI. For this reason, we provide as training few annotated samples for f. Based on these samples we then detect abnormality. To further simplify the problem we aim to extract information that better describes the actions we are interested. Examples of such actions are *running* and *wrong way*. We calculate a set of feature vectors $\mathbf{M}$ to represent the ROI over time ($R_i \rightarrow M_i$).

We assume that there is a primary motion pattern in the sequence and that the abnormal events are rare and of short duration. We detect abnormal motion patterns and implement f by using non-linear manifold learning on the extracted feature set $\mathbf{M}$ to produce low-dimensional representation. A novelty classifier is trained in this subspace to provide the abnormality labeling.

Motion features are better suited to describe events of interest in video surveillance analysis than pixel-based appearance features [29] as they are more descriptive, while being less computationally complex for the construction of the graph.

Note that separating the abnormality detectors from the selection of the ROI is a common strategy in the literature [1, 23, 21]. The ROI is defined in the current work manually (only once) by the operator. As an alternative, it could also be selected on a grid similar to the procedure described in [1].

2.2. Low-level features

The proposed framework is flexible and can use a set of different low-level features. In the specific implementation of this paper, we analyze motion vectors produced by block matching, as these can be extracted directly from the compressed video stream.

The vectors are calculated over a grid ($w \times h$) of non-overlapping patches defined in the ROI:

$$R_i \rightarrow M_i = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,w} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,w} \\ \vdots & \vdots & \ddots & \vdots \\ m_{h,1} & m_{h,2} & \cdots & m_{h,w} \end{pmatrix} \tag{2}$$

At each iteration we search in a neighborhood of the patch (k,l) to find the horizontal and vertical displacement vector $m_{k,l} = (u_{k,l}, v_{k,l})$. M_i is reshaped by concatenating all the scalar values into a column vector $\vec{x}_i$:

$$M_i = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,w} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,w} \\ \vdots & \vdots & \ddots & \vdots \\ m_{h,1} & m_{h,2} & \cdots & m_{h,w} \end{pmatrix} \rightarrow \vec{x}_i = \begin{pmatrix} u_{1,1} \\ v_{1,1} \\ u_{1,2} \\ v_{1,2} \\ \vdots \\ u_{h,w} \\ v_{h,w} \end{pmatrix} \tag{3}$$

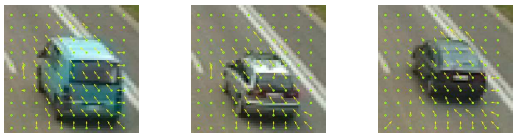
The similarity is calculated using the Euclidean distance between the motion feature vectors $\vec{x}_i$.

The dimensionality of this feature space is much lower than the pixel-based region representation. For example, if we assume a 100×100 ROI the color pixel dimensionality is 30000, while a 20×20 grid of motion vectors (5×5 block) generate a dimensionality of only 800. This results in a considerable reduction of the overall complexity. The parameters in this process are the block (patch) size, which defines the minimum object size, and the search window, which defines the maximum speed.

2.3. Manifold learning

Linear manifolds can be discovered by Principal Components Analysis (PCA) and Multi-Dimensional Scaling (MDS) that provide a good approximation of the high-dimensional feature space under linearity and Gaussian assumptions of the feature values. However, they cannot provide correct mappings when the feature vectors lie on a non-linear manifold.

Non-linear manifolds can be discovered by graph dimensionality reduction algorithms which rely on metrics defined on a neighborhood graph. Based on these metrics and under certain constraints, the mapping is produced by solving an eigenvalue problem to find the solution that minimizes the projection error. For example, Isomap [28] is an extension to MDS where the distance metric is the geodesic distance defined on a neighboring 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) [30] (previously known as Semi-Definite Embedding), which solves an optimization problem which by maximizing the distance between the nodes in a neighboring graph while preserving the distances along the edges and the angles between the edges. This optimization problem is solved using Semi-definite Programming. Locally Linear Embedding (LLE) [22]



(a) (b) (c)

Figure 2: Motion vectors represent objects that follow the same path in a similar manner, independently of their appearance

and Laplacian Eigenmaps 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 take into account all the paths from one node to another, and not just the shortest path, thus preserving local structure.

LE is an application of Spectral Graph Theory of the Graph Laplacian. Given a vector set $\mathbf{X} = \{\vec{x}_i = 1, \ldots, N\}$ the procedure to perform LE is formally stated below:

1. Create the weighted neighboring undirected graph matrix $\mathbf{W}$ from the multi-dimensional vectors $\vec{x}_i$.

2. Compute the combinatorial graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Where $\mathbf{D}$ is the diagonal weight matrix such that its entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$, $d_{ii} = \sum_j w_{ij}$.

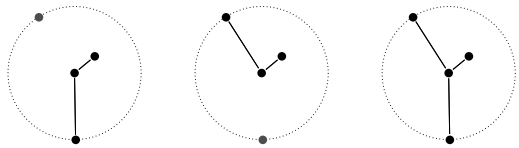
3. Solve the generalized eigenvalue problem of the graph Laplacian.
$$\mathbf{L}\vec{y} = \lambda \mathbf{D}\vec{y} \tag{4}$$

4. Embed into s-dimensional space using the first m eigenvectors in ascending order of eigenvalues, starting from the first non-zero eigenvalue (with $\lambda_0 = 0 < \lambda_1 < \ldots < \lambda_s$).
$$\vec{x}_i \rightarrow (\vec{y}_1(i), \vec{y}_2(i), \ldots, \vec{y}_s(i)) \tag{5}$$

2.4. Graph Construction

The neighboring graph is crucial to the success of the manifold learning process. The graph neighbor parameters define what is considered to be *local*. An inappropriate selection of these values produces a distorted embedding. In general, using a small number of neighbors better represent the local structure. However, the solution becomes more sensitive to the selected weighting scheme.

The k-nearest neighbors is a commonly used neighboring graph that is based on the rule that each node is connected to at least k neighboring (closest) nodes sorted by a similarity measure (e.g. 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 neighbors of i or node i is among the k closest neighbors of j. We apply an exception to these rules that takes place when there are multiple nodes sharing the same similarity with the k-th neighbor. We force all these nodes to connect to the reference node. The probability of such duplicate similarity is high in discrete feature spaces like those generated by a change detection mask and by motion vectors. Without



(a) (b) (c)

Figure 3: K-nn rules example (K = 2): (a,b) Original rules only one of the two equidistant nodes will be connected at random and (c) correct nearest neighbor graph using our extension

Algorithm 1 K-nearest neighbors with duplicates

```
function K-NN(data, K)                          ▷ construct the K-nn graph from the distance matrix data
    n ← number of rows in data
    Create an empty graph g
    for i = 1 to n do                           ▷ iterate for every row in the matrix
        row ← data[i]                           ▷ get the i-th row
        index ← argsort(row)                    ▷ get the index that would sort the row
        for j = 2 to K + 1 do                   ▷ connect the k nearest neighbors
            g(i, index[j]) ← connected
            g(index[j], i) ← connected          ▷ undirected graph
        end for
        while row[index[j + 1]] = row[index[k + 1]] do   ▷ connect equidistant nodes
            j ← j + 1
            g(i, index[j]) ← connected
            g(index[j], i) ← connected          ▷ undirected graph
        end while
    end for
    return g                                    ▷ Return the graph
end function
```

this exception, the reference node would randomly connect to one of the candidate nodes and the resulting structure would not be symmetrical (Fig. 3). Algorithm 1 describes the brute force implementation of the extended k-nn graph construction. To automatically choose the number of neighbors, we follow the iterative process (Alg. 2) that provides a connected graph with the minimum possible k.

The weighting of the graph is provided by the following function:

$$f(\vec{x}_i, \vec{x}_j) = \begin{cases} \exp^{\frac{-\|\vec{x}_i, \vec{x}_j\|}{\sigma}} & \text{if } \vec{x}_i, \vec{x}_j \text{ are connected,} \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

The parameter σ is effectively separating the local neighborhood into *close* and *distant relatives*. A good value can be provided by applying Otsu's method [18] to the edge histogram of the minimum-kNN graph. The complete process of creating the neighborhood graph is effectively parameter-less since K and σ are estimated from the data.

The proposed extended minimum-kNN graph is iteratively increasing the number of nearest neighbors, thus there is a pos-

Algorithm 2 minimum K-nearest neighbors

```
procedure MINKNN(data)                          ▷ Find connected graph of the data with the minimum K
    K ← 1
    g ← K-NN(data,K)                            ▷ Create graph
    while g ≠ connected do
        K ← K + 1                               ▷ Increase neighbors
        g_previous ← g                          ▷ Save previous graph
        g ← K-nn(data, K)                       ▷ Create new graph
    end while
    g ← g_previous                              ▷ Restore the last connected graph
    return g                                    ▷ Return the graph
end procedure
```

(a) (b) (c)

Figure 4: Limitations of motion vectors and change detection from sequence S1: (a,b) Uniform color, (d) Duplicate detection of a running man (change detection only)

sibility of increasing k so much that the graph is dense. While the algorithm will be able to provide a projection even under these circumstances the solution will be equivalent to spectral clustering methods.

To investigate these extreme cases, let us assume that we have a set of N multi-dimensional vectors $\mathbf{X} = \{x_i\}$ and a set of n outlier vectors $\mathbf{A} = \{a_i\}$. Let these vectors sets follow Gaussian distributions $(\vec{\mu_x}, \Sigma_x)$, $(\vec{\mu_a}, \Sigma_a)$ and be very far apart from each other. We combine $(X \cup A)$ these vector sets and construct the graph $\mathbf{G}$ based on the proposed rules. The minimum-k-NN (Alg. 2) algorithm will start from $k = 1$ and will continue to increase until $k = n - 1$. At that point the subgraph of the outlier vectors $\mathbf{G}_A \subset \mathbf{G}$ will be fully connected, while the main graph remains unconnected. However, when $k = n$, the rules will force all of the vectors in $\mathbf{A}$ to connect to the closest $\vec{x_i} \in \mathbf{X}$ point making the graph connected. Thus the worst case scenario gives $k = n$. In practice, the minimum-k-NN becomes fully connected using just a few nearest neighbors. This happens because the features in a video sequence without scene cuts generally evolve smoothly over time and they evolve incrementally from normal to abnormal. This is apparent in Figure 6, where each moving object follows a loop and the abnormal motion patterns gradually deviate from the normal instances.

2.5. Novelty classifier

The manifold learning process discovers the internal structure of the input feature space. In this non-linear subspace, the natural local clustering of the vectors emerges. Statistically common vectors are grouped together whereas rare, unexpected instances are mapped as outliers.

To detect the outlier, we train a novelty classifier based on the Mahalanobis distance. The Mahalanobis distance provides a measure of conformity for the represented frames. Given a set $\mathbf{C} = \{\vec{c}_1, \vec{c}_2, \ldots, \vec{c}_N\}$ representing training samples (*normal* instances) in the reduced space. We calculate the covariance matrix $\mathbf{P}_C$ and the mean $\bar{c}$ that correspond to $\mathbf{C}$. The abnormality score of a vector $\vec{y_i}$ in the lower dimensional space is calculated as:

$$\mathbf{SC}_C(\vec{y_i}) = (\vec{y_i} - \bar{c})^{\mathbf{T}} \mathbf{P}_C^{-1} (\vec{y_i} - \bar{c}), \tag{7}$$

Based on this metric we calculate the abnormality score and we label frames in the sequence that score above the threshold as abnormal.

2.6. Feature spaces: limitations

We compare motion vectors with the pixel-based representation and the change detection mask. The change detection mask is calculated by thresholding the absolute frame difference in the ROI. The threshold value (c_τ) is selected such that it maximizes the detection of foreground pixels that belong to moving objects against those that are attributed to noise.

Each feature space has intrinsic limitations. The color or gray-level *pixel* images are very sensitive to appearance changes. We can see this in the projections of Figure 6 where common events have a wide spread over the three-dimensional space. In fact, although these objects follow the same path, they do not share similar appearance. This sensitivity leads to performance degradation in abnormal motion detection.

The *change detection mask* is less sensitive to appearance changes (common motion patterns are mapped in a compact ring in Fig. 6). However, it may not detect objects that move very slowly, stop or have areas of uniform appearance. In addition to this, duplicate imprints may be generated by the object in the correspondence of the previous and next frame. Because of these problems, crowded scenes will not be represented effectively by change detection masks. Some of the issues can be avoided by using adaptive methods to learn the background and thresholds, along with increasing the sampling rate of the video capture.

Finally, *motion vectors* suffer from similar limitations in the case of non-moving objects and objects with uniform appearance, but there are no issues with duplicate imprints and they behave more reliably in crowed scenes. Also due to the nature of the motion vectors calculation, the measurements at the edges of the image (where an object enters or leaves the scene) are not reliable since the search area is not any more symmetrical.

These limitations (Fig. 4) combined with the inherent ambiguity of the Euclidean similarity measure will cause a number of incorrect connections between vectors in the graph. Such errors are known to hinder dimensionality reduction algorithms that are based on the calculation of the geodesic distances. However, as LE considers the overall connectivity of each node and its local neighborhood, it is less sensitive to short-circuiting. To further reduce the probability of occurrence of such a case, we weight the graph to reduce the influence of long distance connections and use minimum neighborhood size (minimum k-NN) graph.

3. Experimental results

We compare the three feature spaces, namely pixel, change mask and motion vectors, to find the best compromise between classification performance and computational complexity and we demonstrate the performance of the proposed framework based on motion vectors against the alternative approach based on change detection mask.

3.1. Evaluation Measures

We evaluate the classification performance based on separation index score, ROC curves and Precision-Recall curves. A

(a) (b) (c)

Figure 5: Sample frames for S1: (a) man running (crossing), (b) car, (c) van

	Pixels		Change detection mask Using previous frame		Motion vectors	
Colour space	RGB	gray	RGB	gray	gray	gray
Low pass	5×5	5×5	5×5	5×5	5×5	5×5
Threshold	8	5	10	8	–	–
Block size	–	–	–	–	5	5
Maximum search	–	–	–	–	5	10
Label	A1	A2	B1	B2	C1	C2
Dimensionality	8424	2808	2808	2808	200	200

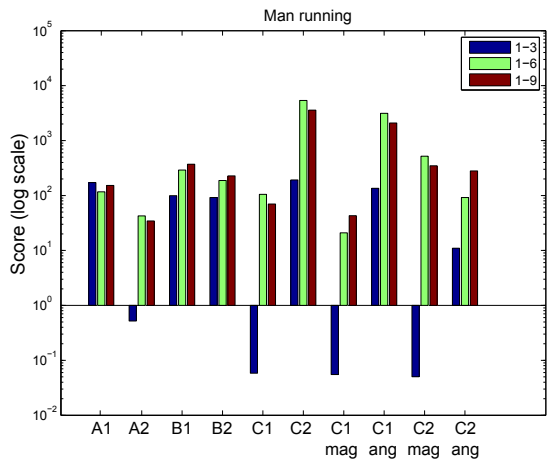
Table 2: Experimental setup parameters for S1

good projection should map the normal vectors (i.e., motion vectors corresponding to a normal event) in a compact area. The abnormal events should be mapped far away from these structures. The *separation index score* is calculated by using (7) and by averaging the score of the highlighted frames when the classifier is trained with all the normal vectors. To compare the scores from different number of dimensions, we divide each result with the average score distance produced, thus generating the relative performance for each highlighted event. Values greater than one are an indication that the highlighted event is mapped far away from the normal motion pattern. Finally, the classification results are evaluated based on the *Receiver Operating Characteristic* (ROC) [10] and Precision and Recall (PR) curves. ROC and PR curves provide complementary information especially when highly unbalanced datasets are classified.

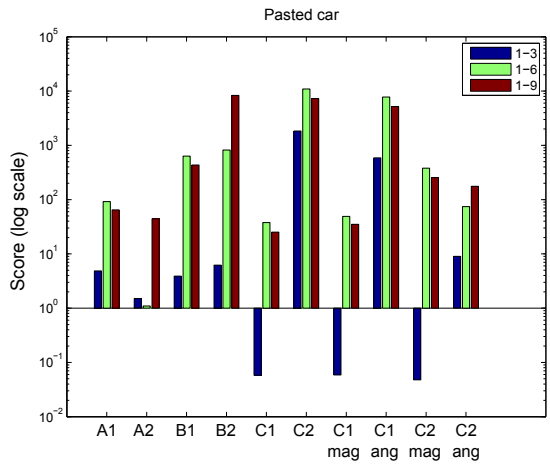
3.2. Feature space comparison

The synthetic sequence S1 (Fig. 5, 2900 frames, 15 fps) is created by concatenating events from a small ROI (54×52 pixels) in a highway. The sequence has a variety of moving vehicles like cars, vans and trucks numbering 20 occurrences in total, thus there is wide variety of moving objects speeds and appearance. Common events correspond to vehicles that pass along (from bottom-right to top-left). The uncommon events are: (i) a man crossing the highway (two instances) and (ii) a car moving horizontally (synthetic). Between the events there are several frames of inactivity.

The sequence is first preprocessed by a lowpass filter applied in each frame (5-pixel square sliding window) to remove noise. Then we compare the results using the following features: (i) pixels in color and grayscale, (ii) change detection in color and grayscale, and (iii) two setups of motion vectors. Table 2 lists the various setups and the corresponding labels used in the figures. In the projections of the applied manifold learning algorithm each event is projected in a loop that starts and ends at the empty (no activity) frames. Figure 6 presents the mapping of the first three dimensions provided by Laplacian Eigenmaps.



(a)



(b)

Figure 7: Separation scores of abnormal events calculated for the projections of sequence S1: (a) man running event and (b) pasted car event. (Legend: dimension range used)

The main loop is produced by the normal events of the passing cars (blue color). It is better defined (more compact) by the B1, B2 and C2 (but not C1) setups when compared with the pixel-based ones (A1, A2). This can be attributed to the fact that the feature spaces produced by the raw pixels is more sensitive to noise and the appearance difference between cars, while the change detection and motion are less sensitive to such differences. It is also visible that B1, B2 and C2 depict a better separation between the normal and abnormal events. The C2 projection however shows a few abnormal frames very close to the no-action frames. This is attributed to very few perceivable motion vectors in these frames since the object is not fully visible in the scene (entry and exit frames) and the motion vectors calculated at the edges of the ROI are unreliable.

The projection provided by using the C1 setup is failing to show the expected structure. The main cause to this effect is the maximum search radius parameter value used. As a result the high speed objects (cars) are not represented properly in the features and thus cannot be distinguished from the abnormal

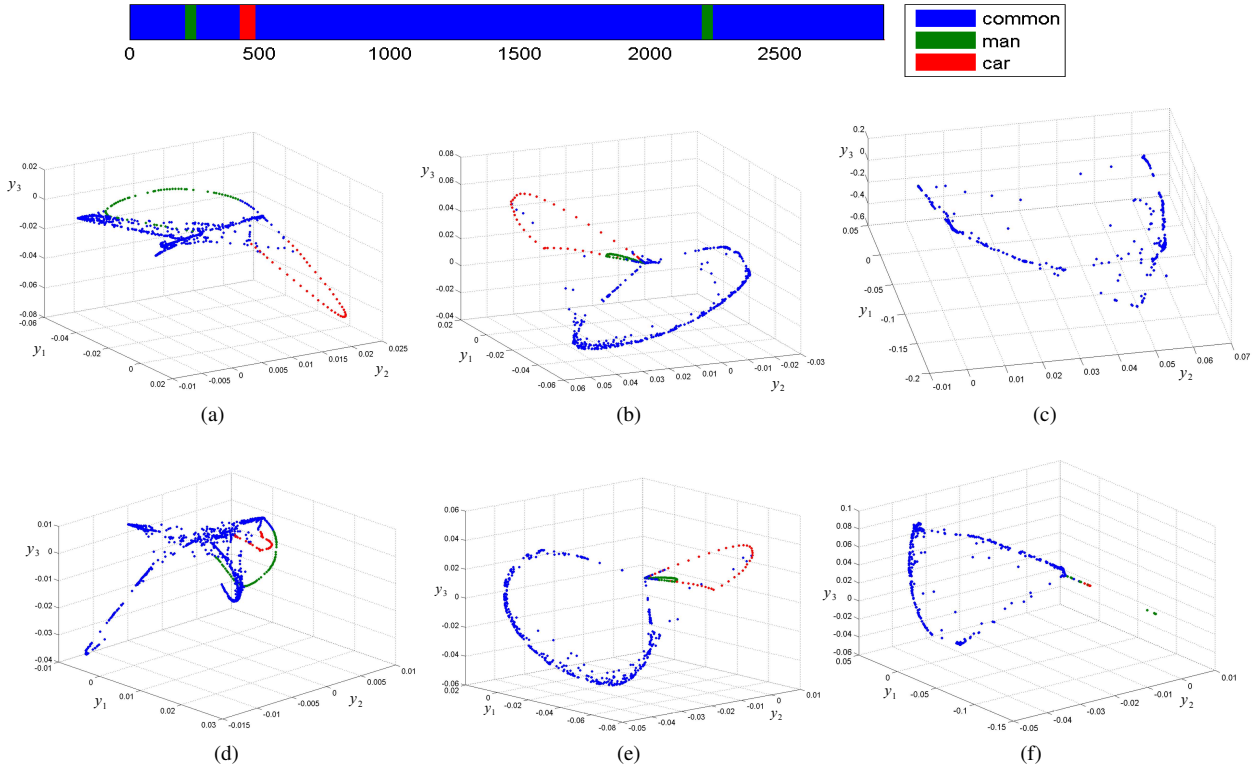


Figure 6: First three dimensions of the mapping produced by Laplacian Eigenmaps: (i) Pixels: (a) color (A1) and (d) gray (A2) , (ii) Change detection mask: (b) color (B1) and (e) gray (B2) and (iii) Motion vectors: (c) C1 setup (f) C2 setup

events.

Figures 7(a,b) show the resulting scores using an increasing number of dimensions (3, 6 and 9) from each feature space. To be able to compare the scores from different number of dimensions, we divide each result with the average score distance produced from the **M** set, thus we have the relative score for each highlighted event. This allows us to establish when an event is mapped outside the area of the main path of motion (values >> 1).

Since the novelty classifier is based on the Mahalanobis distance we are restricted by the Gaussian assumption for the distribution of the normal vectors in the projected space. Figure 6 shows the projection of motion features in three dimensions where the main motion pattern is projected in a torus. As presented in Figure 7 the first three dimensions provide only marginal separation since the average distance is between $0-10$. Nevertheless the torus, while not Gaussian, is projected on a plane almost perpendicular to the plane formed by the abnormal vector loops thus an ellipsoid is enough to provide detection of the abnormal events. However, in higher dimensions the trend to pack the normal vectors in a small area is magnified to provide an average distance close to 10^4. Distances of this magnitude allow us to use the Gaussian assumption as a good approximation for the distribution of the common vectors in higher dimensions.

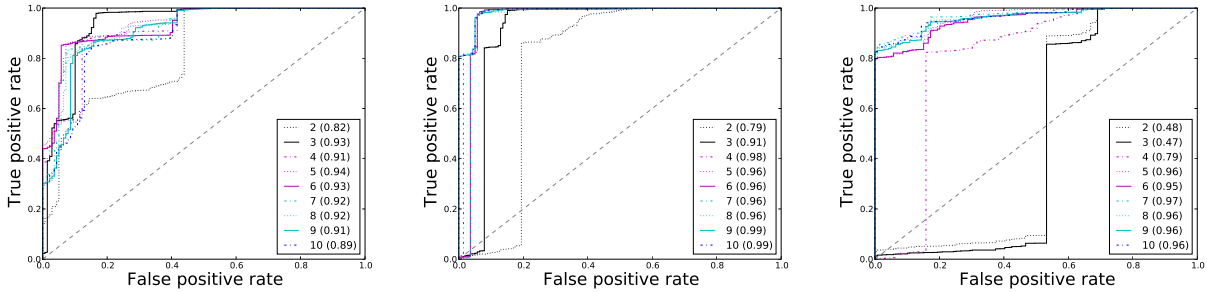
The values confirm our conclusions from the visual inspection where change detection and motion vectors give better separation results. We can also see the effect of increasing the search radius for the motion vectors calculation, where the correct calculation magnitude accounts for most of the improvement. The best score comes by using the C2 feature set, because it gives consistent results for both events.

Further to the separation index we also compute the ROC curves produced by using the novelty classifier trained using the frames that of the passing car in Figure 5c (frames: 1211-1238, 1% of the sequence). The B2 and C2 setups provide better results than those using setup A2 (Fig. 8). B2 is marginally better, but the cost of higher feature space dimensionality (ratio 14:1 compared to the motion vectors) favors the motion vectors.

3.3. *Real-world sequence*

We apply the framework in a video sequence (S2) from a dataset [1]: it is a scene from an underground train station (64900 frames of size 512x384, compressed) and the camera oversee the exit turnstiles of the platform. During the normal motion flow people arriving with the train are exiting the platform through the turnstiles and turn left or right on the corridor. The abnormal actions are performed by actors going the wrong way and entering the platform (Fig. 9).

Given the ROI **A** (Fig. 9) we apply our abnormality detector. The video is preprocessed using a sliding low pass filter (averager) with a square window of seven pixels. We calculate motion vectors (5×5 grid) using a 16×16 block and a search area of 16×16 pixels. We the motion features with the change



(a) (b) (c)

Figure 8: ROC curves of the abnormality detection and classification varying the number of dimensions, trained using one instance of a car passing (1% of sequence length). Setups: (a) A1, (b) B2 and (c) C2

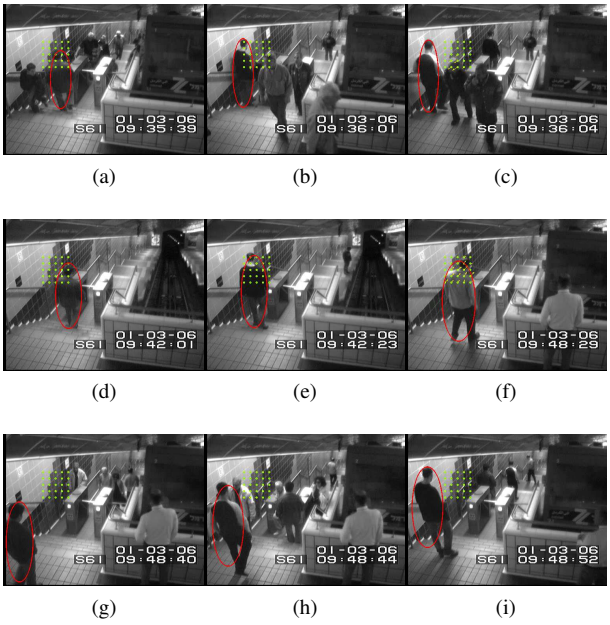


Figure 9: Sample images of the abnormal events in S2 and the ROI **A**

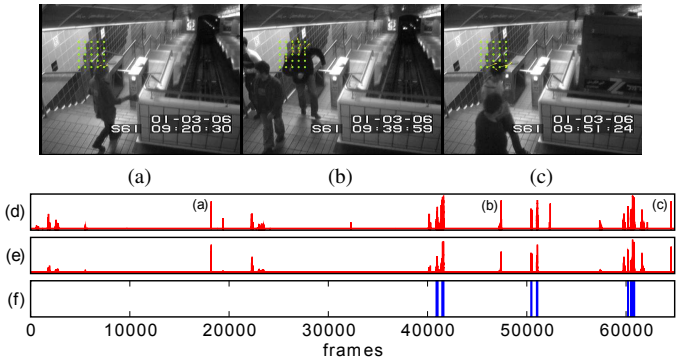


Figure 10: Ranking results for the S2 sequence. Sample frames of additional detected events: (a) Cleaning Lady, (b) jumping and (c) passing along. Timelines: (d) ranking abnormality, (e) filtered ranking and (f) actor abnormal motion ground truth

detection mask for the same region (112×112 pixels), produced by a threshold of 6. To further reduce complexity we also (automatically) remove the null frames of inactivity except, one frame before and one after each such segment. The process reduces the size of the feature space to 5090 and to 3764 vectors for change detection mask and motion respectively. Training also takes place in these subsets, thus given that the first 7500 frames of the sequence as normal, the training vectors for motion vectors consist of the first 584 and for change detection mask of the first 784 of the corresponding feature subsets. The two prominent features motion vectors and change detection mask are compared using ROC and Precision - Recall curves calculated for the labeling of the *complete* sequence of frames.

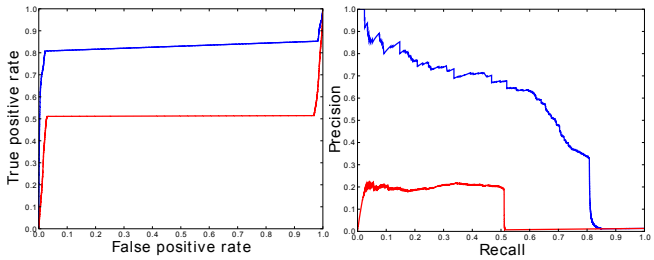
3.4. *Motion vs Change-detection*

Figure 10 shows the ranking results before and after the application of an one dimensional median filter (window of 5 pixels) in the output to remove inconsistent detections. The events in the ground truth (blue timeline) are defined based on the entry and exit of the object in region **A**. The results show consistency with the ground-truth. However, there three peaks that are persistent, even after filtering. These peaks correspond to real abnormal events that deviate from the main motion in the region but are not present in scripted events. In more detail: (a) a cleaning lady moves and *enters* the exit platform an action that complies with the *wrong way* event but is not in the scripted ground truth. (b) One of the actors while exiting the platform *jumps* forward. Because we consider both magnitude and angle we can also detect these events at the same time. (c) A person is visible in **A** walking fast along the main corridor effectively causing a false alarm.

It is also important to notice that the detected abnormal events are caused by moving objects in the ROI that vary in scale. This ranges from the head (small area) of the cleaning lady to the torso (large area) of the jumping person. Partial occlusions can also cause short lived errors in the labeling. These

(a) (b) (c)

Figure 11: Sample boundary frames of abnormal events in region (A)



(a) (b)

Figure 12: Classification performance comparison for ROI **A**: (a) ROC curves, (b) Precision and Recall. Legend: (blue) Motion vectors and (red) change detection mask

errors are addressed in two steps: i) the locality preserving characteristics of Laplacian Eigenmaps filter small deviations on the motion features so the projections are temporally smoothed, ii) we remove inconsistent labeling by post-processing (temporal filtering) the results (Fig. 10).

We compare (Fig. 12) the motion vector classification against the change detection mask. Using ten dimensions selected based on the highest AUC value. The ground truth used is the extended one that includes the unscripted abnormal events (Fig. 12 (a,b)). In these graphs, the motion vectors show a lower false alarm rate with high true positive rate, while the change detection mask is no better than a random detector. The PR curves are more informative and show that motion vectors are offering a better classification performance for the specific event (4 times better precision) and change detection mask cannot offer more than ~ 0.5 recall. Since the methodology used is exactly the same for both features, we can conclude that the problem lies on the descriptive performance of the features. As expected, a small region has a higher likelihood to become crowded which combined with shadows, illumination changes and slow apparent motion (due to perspective) reduce the performance of the change detection mask.

3.5. Alternative methods

Novelty detection can be achieved by using classifiers such as one-class SVM [17] and nearest neighbors. The first method (SVM) maps the training class of positive (normal) instances into a higher dimensional space using a predefined kernel function. In this space the algorithm learns the optimal boundary lines that enclose the single normal class. Based on these boundaries the test vectors are labeled as normal or abnormal. In the experiments we use the implementation provided by SVMLIB [6]. Results are presented using a Radial-based-Function (RBF) and linear kernel respectively. The second method is an adaptation of the multi-class nearest neighbor classifier to one-class novelty problems. The k nearest neighbors of the test vectors are discovered among the training samples and their average (mean) Euclidean distance is thresholded to label for abnormality.

Both k-NN and the proposed method need only two parameters to be manually defined (neighbors/dimensionality and threshold), while the more complex one-class SVM requires the selection of the kernel function (plus the parameters associated with it), the learning parameters for the training and the threshold constant. The classification threshold for the one-class SVM is provided by the SVMLIB implementation with default parameters applied for training. For the k-NN classifier we use as threshold the maximum score given to the training vectors from the algorithm, the same method is adapted also for the proposed approach threshold. For the one-class SVM and k-NN we use the high-dimensional motion vector features. Training is based on the subset described in 3.4. We also median filter the sequence using a 5-frame window to filter out non-persistent labels.

The results (Fig. 13) using the one class SVM are inferior to the proposed approach. This can be attributed to high-dimensionality [14] and temporal (usually non-linear) correlation between input vectors. In addition, one-class SVMs have been found to be very sensitive to the parameter selection in the work of Manevitz and Yousef [17]. The nearest neighbor classifier shares the characteristics of the nearest neighbor graph. However, the labeling is more sensitive to the selection of the neighborhood size. Using a small number of neighbors (Fig. 13) the classifier provides a labeling that misses a few of the abnormal events. The increase in neighborhood size increases the global structure influence and thus the abnormal events are smoothed out. In contrast, the proposed dimensionality reduction using the minimum-kNN graph constructed on the complete sequence discovers a mapping which compresses the normal (common events) far away from the outliers. This allows the novelty detector to label the abnormal events correctly.

We calculate precision PR and recall RC values of the labeling based on per frame and event abnormality respectively. Precision describes how certain we are that the detected abnormal frame/event is truly abnormal. Recall measures the probability that the abnormal events are detected. These measures are given by the formulas in (8).

$$PR = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \qquad RC = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{8}$$

When considering frames we evaluate the detectors ability to detect the abnormal frames, in this case *i*) True positive frames TP_f are those that are correctly labeled abnormal, *ii*) False positive frames FP_f are incorrectly labeled abnormal and *iii*) False negative frames FN_f are incorrectly labeled normal. However, abnormality detectors in real world situations should detect events and provide warnings of abnormality. Thus we need

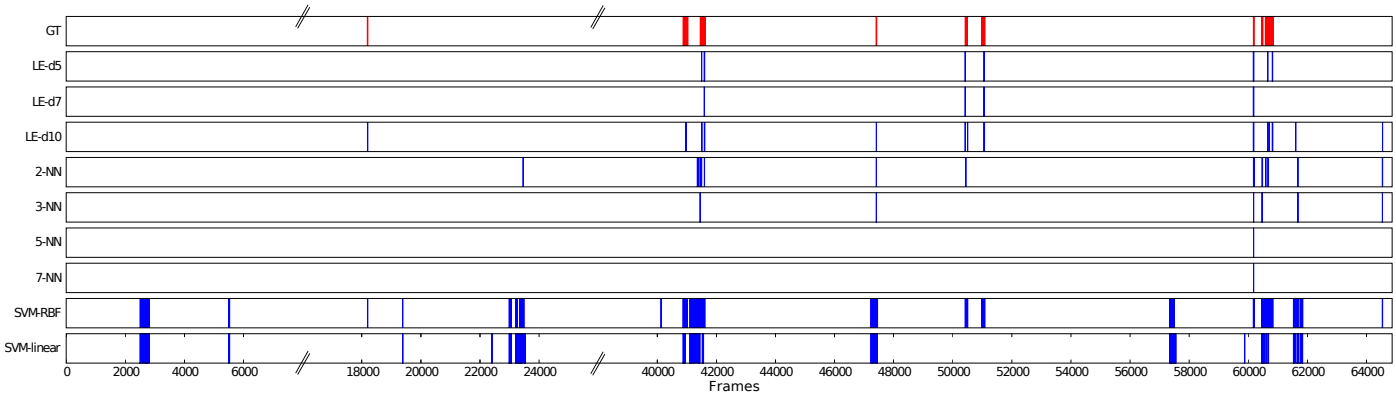


Figure 13: Frame labeling results for the *(i)* proposed approach using 5,7 and 10 dimensions (LE-d5,LE-d7 and LE-d10), *(ii)* novelty nearest neighbor classifier (k-NN) using 2,3,5 and 7 neighbors and *(iii)* one-class SVM using the RBF and linear kernels. (GT: Ground truth)

to see how the detectors are performing in detecting the abnormal events in the sequence. We define a simple set of rules for evaluating the detection of these events *i*) True positive detections TP_e exist when at least one frame labeled abnormal overlaps with an event in the ground truth, *ii*) False positive detections FP_e are frame segments that are incorrectly labeled abnormal (i.e. they do not overlap with the ground truth) and *iii*) False negative detections FN_e are assigned where abnormal segments in the ground truth are not abnormal in the labeling. These event detection guidelines will give high values to detectors which have very extended abnormality segments, For example a detector could return all the sequence as abnormal and have high PR_e and RC_e values. However, *frame* precision and recall values will show that such a detector is performing poorly. Thus the combination of these measures provided a complimentary view in the performance of the methods.

Table 3 presents the frame and event evaluation based on precision and recall values where overall the proposed approach performs better in both evaluation modes *frame* and *event* evaluations. Precision is high ($\bar{PR}_f \sim 0.903$, $\bar{PR}_e \sim 0.923$) and degrades slowly while increasing the dimensionality. The higher the dimensionality that we use the more noise that we introduce in the projections resulting in more false positive instances (thus the event precision degrades faster). Recall is low since not all the frames are detected however the detector correctly detects most of the abnormal events using ten dimensions ($RC_e = 0.909$). Using the k-NN novelty classifier is not able to detect all the abnormal events and the precision and recall values degrade by the increasing k. The SVM classifier labels a lot of frames as abnormal and achieves high *event* Recall value using the RBF kernel. However precision shows that this labeling is not reliable and there is a good amount of false positives.

3.6. Computational complexity

Although Laplacian Eigenmaps does not require to store and solve a dense matrix of size n (number of vectors) it is still more expensive than direct application of nearest neighbor or one-class SVM methods. The bottleneck in the process is the graph construction. The brute force method, while being easily parallelized, requires $O(dn^2)$ distance calculations (d is the vector dimensionality). However recent work [7] has proved that it is possible to get approximate k-nearest neighbors with a lower than quadratic complexity $O(dn^t)$ where t is a small number close to 1. Additional speed-ups can be achieved by using out-of-sample extensions [4] to the Laplacian Eigenmaps methodology. In such a case, the on-line complexity of the algorithm will be comparable to alternative methods like one-class SVM.

4. Conclusions

We demonstrated in this paper the usefulness of non-linear subspace learning as core component of a local motion abnormality detector. The detector is based on first discovering the motion manifold and then constructing an off-line detector to label abnormal motion patterns within a ROI, without the use of object detection and tracking. We showed that we can achieve comparable performance but lower sensitivity to noise while utilizing a more manageable feature space of lower initial dimensionality than the image pixels or change detection masks. We discussed the results based on separation scores and ROC curves. The performance of the framework was demonstrated in a real-world scene of an underground train station. We compared the proposed approach with alternative popular novelty detection methods and showed that the proposed detector performs better in detecting abnormal events.

Future work includes the extension of the proposed method to an unsupervised module to automatically detect the majority class. We also aim to adapt the method to on-line processing.

Acknowledgments

This work was jointly supported by the Engineering and Physical Sciences Research Council (UK) and British Telecommunications Plc (UK).

References

[1] A. Adam, E. Rivlin, I. Shimshoni and D. Reinitz, Robust Real-Time Unusual Event Detection using Multiple Fixed-Location Monitors, IEEE

Method	Frames					Events				
	TP_f	FP_f	FN_f	PR_f	RC_f	TP_e	FP_e	FN_e	PR_e	RC_e
LE-d5	46	3	823	0.939	0.053	7	0	4	1	0.636
LE-d7	25	3	844	0.893	0.029	4	0	7	1	0.364
LE-d10	121	17	748	0.877	0.139	10	3	1	0.769	0.909
2-NN	65	45	804	0.591	0.075	7	5	4	0.583	0.636
3-NN	17	18	852	0.486	0.020	3	3	8	0.500	0.273
5-NN	4	0	865	1	0.005	1	0	10	1	0.091
7-NN	1	0	868	1	0.001	1	0	10	1	0.091
SVM-RBF	757	1430	112	0.346	0.871	11	54	0	0.169	1.0
SVM-linear	122	1381	747	0.081	0.140	6	54	5	0.100	0.545

Table 3: Precision and Recall values and detection information for the abnormality detection methods on ROI **A**.*(i)* proposed approach using 5,7 and 10 dimensions (LE-d5,LE-d7 and LE-d10), *(ii)* novelty nearest neighbor classifier (k-NN) using 2,3,5 and 7 neighbors and *(iii)* one-class SVM using the RBF and linear kernels

Transactions on Pattern Analysis and Machine Intelligence 30(3) (2008) 555–560.

[2] E. Andrade, O. Blunsden and R. Fisher, Performance Analysis of Event Detection Models in Crowded Scenes, in: Proc. IET International Conference on Visual Information Engineering (IET, London, 2006) 427–432.

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

[4] Y. Bengio, J.-F. Paiement and P. Vincent, Out-of-Sample Extensions for LLE, Isomap, MDS, Eigenmaps, and Spectral Clustering, in: Proc. Advances in Neural Information Processing Systems 16 (NIPS, Vancouver, 2003) 177–184.

[5] O. Boiman and M. Irani, Detecting Irregularities in Images and in Video, International Journal of Computer Vision 74(1) (2007) 17–31.

[6] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[7] J. Chen, H. R. Fang and Y. Saad, Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection, Technical Report, Minnesota Supercomputer Institute, Minneapolis, 2008.

[8] P. Cui, L.-F. Sun, Z.-Q. Liu and S.-Q. Yang, A Sequential Monte Carlo Approach to Anomaly Detection in Tracking Visual Events, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition. (IEEE, New York, 2007) 1–8.

[9] E. B. Ermis, V. Saligrama, P.-M. Jodoin and J. Konrad, Abnormal behavior detection and behavior matching for networked cameras, in: Proc. ACM/IEEE International Conference on Distributed Smart Cameras, (IEEE, New York, 2008) 1–10.

[10] T. Fawcett, An introduction to ROC analysis, Pattern Recognition Letters 27(8), (2006) 861 – 874.

[11] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan and S. Maybank, A System for Learning Statistical Motion Patterns, IEEE Transactions on Pattern Analysis and Machine Intelligence 28(9) (2006) 1450–1464.

[12] F. Jiang, Y. Wu and A. K. Katsaggelos, Abnormal Event Detection from Surveillance Video by Dynamic Hierarchical Clustering, in: Proc. IEEE International Conference on Image Processing. Vol. 5, (IEEE, New York, 2007) 145–148.

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

[14] S. Klement, A. Madany Mamlouk and T. Martinetz, Reliability of Cross-Validation for SVMs in High-Dimensional, Low Sample Size Scenarios, in: Proc. International Conference of Artificial Neural Networks, Lecture Notes in Computer Science, Vol. 5163 (Springer, Berlin, 2008) 14-50.

[15] J. Li, S. Gong and T. Xiang, Global Behaviour Inference using Probabilistic Latent Semantic Analysis, in: Proc. British Machine Vision Conference (BMVA, Malvern, 2008).

[16] Z. Li, Y. Fu, T. Huang and S. Yan, Real-time human action recognition by luminance field trajectory analysis, in: Proc 16th ACM International Conference on Multimedia (ACM, New York, 2008) 671–676.

[17] L. M. Manevitz and M. Yousef, One-class svms for document classification, Journal of Machine Learning Research 2 (2002) 139–154.

[18] N. Otsu, A Threshold Selection Method from Gray-Level Histograms, IEEE Transactions on Systems, Man and Cybernetics 9(1) (1979) 62–66.

[19] C. Piciarelli, G. Foresti, L. Snidara, Trajectory clustering and its applications for video surveillance, in: Proc. IEEE Conference on Advanced Video and Signal based Surveillance. (IEEE, New York, 2005) 40–45.

[20] F. Porikli and T. Haga, Event Detection by Eigenvector Decomposition Using Object and Frame Features. in: Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshop, (IEEE, New York, 2004) 114–114.

[21] I. Pruteanu-Malinici and L. Carin, Infinite Hidden Markov Models for Unusual-Event Detection in Video, IEEE Transactions on Image Processing 17(5) (2008) 811–822.

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

[23] D. Russell and S. Gong, Exploiting Periodicity in Recurrent Scenes. In: British Machine Vision Conference (BMVA, Malvern, 2008).

[24] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: Proc. 17th International Conference on Pattern Recognition ICPR 2004. Vol. 3. (IEEE, New York, 2004) 32–36.

[25] R. R. Sillito and R. B. Fisher, Semi-supervised Learning for Anomalous Trajectory Detection, in: Proc. British Machine Vision Conference (BMVA, Malvern, 2008).

[26] K. Sudo, T. Osawa, K. Wakabayashi, H. Koike and K. Arakawa, Estimating Anomality of the Video Sequences for Surveillance Using 1-Class SVM. IEICE Transactions on Information and Systems E91-D (7), 2008 1929–1936.

[27] D. Tao, X. Li, X. Wu and S. J. Maybank, Geometric Mean for Subspace Selection, IEEE Journal of Pattern Recognition and Machine Inteligence 31(2) (2009) 260–274.

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

[29] I. Tziakos, A. Cavallaro and L.-Q. Xu, Video event segmentation and visualisation in non-linear subspace, Pattern Recognition Letters 30(2) (2009) 123–131.

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

[31] T. Xiang and S. Gong, Activity based surveillance video content modelling. Pattern Recognition 41 (7) (2008) 2309 – 2326.

[32] J. Yuan, Z. Liu and Y. Wu, Discriminative subvolume search for efficient action detection. in: Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshop, (IEEE, New York, 2009) 2442–2449.

[33] D. Zhang, D. Gatica-Perez, S. Bengio and I. McCowan, Semi-Supervised Adapted HMMs for Unusual Event Detection, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1 (IEEE, New York,2005) 611–618.

[34] H. Zhong, J. Shi and M. Visontai, Detecting unusual activity in video, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, (IEEE, New York, 2004) 819–826.

[35] H. Zhou and D. Kimber, Unusual Event Detection via Multi-camera Video Mining, in: Proc. IEEE Conference on Computer Vision and Pat-

tern Recognition, Vol. 3 (IEEE, New York, 2006) 1161–1166.