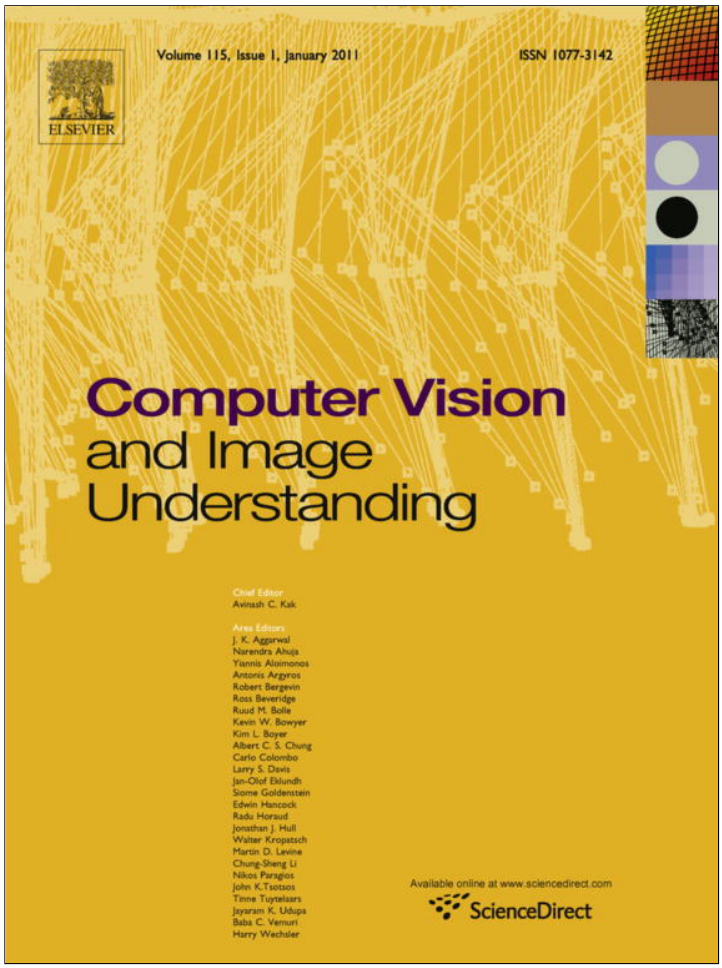


(This is a sample cover image for this issue. The actual cover is not yet available at this time.)

Computer Vision and Image Understanding 115 (2011) 1099–1111

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Detecting anomalies in people's trajectories using spectral graph analysis

Simone Calderara [a], Uri Heinemann [c], Andrea Prati [b,*], Rita Cucchiara [a], Naftali Tishby [c]

[a] Department of Information Engineering, University of Modena and Reggio Emilia, Via Vignolese, 905/b - 41122 Modena, Italy
[b] Department of Engineering Science and Methods, University of Modena and Reggio Emilia, Via Amendola, 2, 42122 Reggio Emilia, Italy
[c] School of Engineering and Computer Science, The Hebrew University, Jerusalem 91904, Israel

ARTICLE INFO

Article history:
Received 12 July 2010
Accepted 15 March 2011
Available online 31 March 2011

Keywords:
Spectral graph theory
Trajectory analysis
Video surveillance
Anomaly detection

ABSTRACT

Video surveillance is becoming the technology of choice for monitoring crowded areas for security threats. While video provides ample information for human inspectors, there is a great need for robust automated techniques that can efficiently detect anomalous behavior in streaming video from single or multiple cameras. In this work we synergistically combine two state-of-the-art methodologies. The first is the ability to track and label single person trajectories in a crowded area using multiple video cameras, and the second is a new class of novelty detection algorithms based on spectral analysis of graphs. By representing the trajectories as sequences of transitions between nodes in a graph, shared individual trajectories capture only a small subspace of the possible trajectories on the graph. This subspace is characterized by large connected components of the graph, which are spanned by the eigenvectors with the low eigenvalues of the graph Laplacian matrix. Using this technique, we develop robust invariant distance measures for detecting anomalous trajectories, and demonstrate their application on real video data.

1. Introduction

Modern society must face threats from many different sources: natural disasters, wars, lack of natural resources (such as water), micro- and macro-criminality, etc. Terrorism is one of today's major security threats. Terrorist attacks are very difficult to predict and do not follow specific known patterns. There is no universally recognized technology or procedure to alert people or prevent terrorist attacks. This is particularly true when the terrorist is a single individual in a crowd, unfortunately a frequent event we try to address in this paper.

Intelligent video surveillance (IVS) is one of the terrorist prevention technologies which has attracted the most interest, due to the increasing presence of surveillance cameras and potential algorithms for analyzing complex scenes that provide visual cues for high-level automatic reasoning. Generally speaking, video cameras provide the richest and most promising source of information about potential terrorist threats. This abundance comes at the cost of very complex data processing and still unreliable automatic procedures, which make completely automated video surveillance impractical. Nevertheless, significant advances in computer vision and machine learning have been made since the time camera feeds were simply sent to monitors in a control center and were scrutinized by human operators. Today, intelligent video surveillance can tap increasingly sophisticated algorithms capable of detecting and tracking moving objects from still cameras in challenging crowd scenarios [1–4].

Reliable methods for fusing the results obtained from multiple cameras, either with overlapping fields of view [5–7] or with disjoint views [8–10], have been developed. These methods make it possible to keep a person's movement under surveillance for longer periods of time, and hence to achieve more complete and continuous monitoring.

Among the numerous features that can be extracted from video, the trajectories of people moving in a scene provide useful and informative cues for detecting suspicious and anomalous behaviors. In fact, there is no simple way to categorize and label people's trajectories such that a "standard" supervised machine learning approach can be utilized. This is mainly because there is no simple model for what can be identified as anomalous or suspicious behavior, and no simple measure of similarity for such trajectories. Conversely, without any a-priori context knowledge, trajectory patterns can be classified as normal or abnormal only by considering their statistical occurrence. Moreover, in real-world contexts, especially in outdoor scenarios with high scene variability, people's trajectories can be very different and therefore very difficult to model.

In this paper we adopt an approach that has proved successful in other unsupervised tasks with similar difficulties. The idea is to represent people's trajectories in space, as detected by video tracking, as sequences of transitions between nodes on a graph. The nodes are determined by vector quantization of short motion

* Corresponding author.
E-mail address: andrea.prati@unimore.it (A. Prati).

1077-3142/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2011.03.003

segments in the video field of view. Using this discrete representation it is possible to statistically characterize "normal trajectories" by the algebraic properties of the graph generated by accumulated trajectories. It is shown that "normal" trajectories, when long enough, capture only a tiny subspace of all the possible sequences of node transitions. This small subspace can be determined by the dominant connected components of the graph, or equivalently, by eigenvectors with small eigenvalues of the graph Laplacian matrix [11–13].

The paper is organized as follows. We first describe the background and related work. Then we detail the three steps of our combined approach: (i) from video to discrete time series (ii) from time series to graph (iii) the anomaly detection method on graphs using the Laplacian filtering. We then describe our experimental results and application of the algorithm to real data, following a conclusion.

2. Related works

The problem of anomaly detection in video-surveillance scenarios has been addressed by several researchers. Partial surveys on the topic by Buxton [14] and Hu et al. [1] give an overview of the vast range of techniques and approaches that can be applied to the problem. In the literature, the problem of anomaly detection has been addressed in two main ways: the first considers an anomaly as a deviation from a pre-learned set of normal events [15–17]; whereas the second approach directly models a set of desired activities for querying and mining [18,19]. Although the latter approach requires the formalization of the set of anomalies that may occur in the scene, the former exploits typical machine learning techniques for inferring infrequent and anomalous events.

Another important factor that influences the problem of detecting anomalies in video surveillance is the choice of features that correctly reveal infrequent events in a scene. Many methods adopt a holistic approach that considers the scene as a whole rather than specific features of it. In [20] an anomaly is detected on a compact representation of elementary video events relative to the motion information of the foreground objects by using a generative model of behaviors trained on a small collection of examples, with an online likelihood ratio test to detect anomalies in new sequences. Basharat et al. [21] and Saleemi et al. [22] successfully analyzed the motion patterns of objects in the scene using non-parametric and parametric probability distribution functions (pdfs), that statistically capture the common motion models and track features, respectively. Anomalies are detected either by adopting first-order dynamic models, Monte Carlo Markov Chains, or by clustering similar distributions using the Expectation Maximization framework. The holistic approaches share the invaluable advantage of detecting deviations from usual and frequent events without any prior knowledge, but rely heavily on the discriminate power of the selected features that typically catch only low-level changes in the recorded scene (i.e., pixel-level illumination changes, objects appearing and disappearing, etc.).

In terms of features, people's trajectories constitute a better choice for representing behaviors in a scene since they can be robustly extracted from complex scenarios by most modern video-surveillance systems [23]. Searching for anomalies in people's trajectories involves defining a measure for comparing trajectories and the ability to cluster common behaviors. Anomalies are spotted by observing the clusters' cardinality and directly learned from unlabeled data [24]. For example, in [25] a modified version of the edit distance was adopted as the similarity measure and anomalies were detected using spectral clustering on distance matrices. Panozzo and Mecocci in [26] modified the iterative *Altruistic Vector Quantization* algorithm to robustly cluster trajectories by pure spatial observations and obtained representative prototypes. Their anomaly detection was based on fitting a spatial Gaussian on each prototype and statistically checking the fit of new trajectory samples. In [27] Junejo et al. applied graph cuts to cluster trajectories using the Hausdorff distance measure. In [7] a system for learning statistical motion patterns was presented. Trajectory clustering was performed using a two-stage fuzzy k-means. First, trajectories were clustered in the spatial domain, then each cluster is sub-clustered in the temporal domain. New trajectories were checked against cluster centers using Bayes rule and anomaly detection was performed by thresholding the resulting probability. Similarly, Sillito and Fisher [28] adopted a semi-supervised scheme to correctly label normal trajectories where the human operator plays a key role in defining the concept of normal paths; conversely, Piciarelli et al. [29] trained an SVM using a corpus of normal trajectories exploiting the SVM novelty detection capabilities to detect anomalous behaviors online.

All the above works compare each trajectory to the previous ones. We adopted a different approach, looking at the scene as a whole, i.e. accumulating the trajectories to create perception of the normal behavior in the scene. For this purpose, we describe the scene as a weighted graph. Note that in our setting we can handle many trajectories at a time. Moreover, the need of the trajectories to be complete is removed, allowing more continuous detection.

Detecting anomalies in graphs (networks) in an efficient and precise way is still a challenge. The works of Lakhina et al. [30,31] try to detect anomalies by the fact that normal graph activity resides in low dimension space. PCA is used in order to define the abnormal space, then each new activity is projected into this space. A projection with high volume can be considered anomalous, i.e. it does not reside in the restrict normal space. This method demands heavy computational load, and a possible solution can be found in [32]. Ide and Kashima [33] try to detect anomalies in computer network by accumulating the first eigenvector for each sample of network activity. They generate a matrix from these eigenvector then the left eigenvector of this matrix is taken and the new sample is projected to it. The sample is considered anomalous if the value resulting from the above process is higher then some threshold, chosen in a sophisticated way.

Here we adopt an unsupervised method for anomaly detection that combines several known and novel techniques. First, we quantize the motion using a partition of the scene into statistical Voronoi cells, and then quantify the trajectories as weighted undirected graph by representing each node as n adjacent cells boundary-crossings. To the best of our knowledge this is a novel representation of trajectories for such applications. The idea underpinning our anomaly detection approach is to analyze the spectral fluctuation of the graph built on cell transitions spanned by people trajectories. Since the first pioneering works [34,35] presented the chance of analyzing graph topology by studying the underlying embedding eigenspace, several applications in computer vision and pattern recognition have been proposed to tackle, by Laplacian analysis, heterogeneous problems ranging from image segmentation [36], to people tracking [37], to image enhancing [38] and retrieval [39].

Several measures exist for comparing graphs of the same size (exact matching) [11], while inexact graph matching is still an intensively studied problem, adopted to cluster similar images [40], 3D meshes [41] and objects shapes [42]. A technique for comparing structural similarities in graphs with the aim of clustering is presented in [43]. It relies on the embedding of the graph topology in a vector-form pattern space that is robust to the permutations of the graph nodes and preserves more informative content than the Laplacian eigenvalues feature vector. The pattern vector embedding is achieved by constructing symmetric polynomials using

the graph Laplacian values. Alternative graph representations, instead of operating directly on the Laplacian matrix, have been used for image clustering. For example, the heat kernel trace on the Laplacian matrix [44] has been proposed as a graph-node permutation-invariant measure for effective inexact graph comparison and clustering.

Although many methods exist for deeply analysing graph spectra, in our proposal we then extend the spectral embedding method used in spectral clustering [45,13], by projecting the graph on the low-frequency eigenvector of its Laplacian matrix. This is analogous to lowpass filtering in Fourier representation of signals, where the Laplacian eigenvectors generalize the standard orthonormal Fourier basis to a general graph. The rational for this representation is that anomalous trajectories will locally deviate from the most frequent cell-transitions and introduce a high-frequency fluctuation in this spectral representation. In the final step we repeat the same projection on powers of the Laplacian matrix, to capture anomalies on larger time scales. This can be performed efficiently using the graph diffusion kernel [46], or the exponential of the Laplacian matrix.

The anomaly detection is done using the divergence measure defined by the canonical angle between the subspaces defined by the k-low eigenvectors of the test and train Laplacian matrices [11,33,32,47]. This measure is an invariant measure that is empirically found to be sensitive to the anomalies we seek in this work.

3. From videos to time series

In this section we describe our trajectory tracking and segmentation algorithms, used for the graph embedding and novelty detection method.

3.1. Single camera processing

The first important processing step in an automatic surveillance system is the extraction of objects of interest. For our application, "objects" means "people" and "interest" means "movement", since we aim to model people's behaviors. In theory, stationary people can be potential threats to security as well. However, completely stationary people are more difficult to detect and less likely to be found in real scenes.

When the cameras are installed in fixed positions the detection of moving people can be achieved by finding the difference between the input frame and a model of the static content of the monitored scene, i.e. the background model. Background modeling is a complex task in "real-world" applications; many difficulties are due to environmental and lighting conditions, micro-movement (e.g. moving tree branches), or illumination changes. The background model must also be constantly updated during the day because of natural intrinsic changes in the scene itself, such as clouds covering the sun, rain and other possible natural artifacts.

The motion detection algorithm adopted in this work is specifically designed to ensure a robust and reliable background estimation even in complex outdoor scenarios. It is a modification of the Statistical And Knowledge-Based Object deTector (SAKBOT) system [3], that increases robustness in outdoor uncontrolled environments [23]. The SAKBOT background model is a temporal median model with a selective knowledge-based update stage. To accommodate the bootstrapping issue [48], the initial background model is initialized by subdividing the input image I into 16×16 size blocks and fast updating them based on single frame differencing: blocks composed of more than 95% of still pixels are forced into the background model. After this bootstrapping stage, the background model is updated using a selective temporal median filter, which retains, for each pixel, the median value of the last k samples, collected in a circular buffer.

The difference between the current image and the background model is computed as the maximum of the differences in R, G and B, and then binarized using two different local and pixel-varying thresholds: a low threshold T_{low} to filter out the noisy pixels extracted due to small intensity variations and a high threshold T_{high} to identify the pixels where a large intensity variation occurs. Moreover, these two thresholds are adapted to the current values in the buffer in order to adapt to illumination changes. More in details, the thresholds are computed as follows:

$$T_{low}(i,j) = \lambda\left(b_{\frac{k+1}{2}+l} - b_{\frac{k+1}{2}-l}\right) \tag{3.1}$$

$$T_{high}(i,j) = \lambda\left(b_{\frac{k+1}{2}+h} - b_{\frac{k+1}{2}-h}\right) \tag{3.2}$$

where $b_p(i,j)$ is the value at position p inside the ordered circular buffer b of pixel (i,j) which contains the last n values of the considered pixel and, consequently, $b_{\frac{k+1}{2}+1}$ the median. λ is a fixed multiplier, while l and h are fixed scalar values. We experimentally set $\lambda = 7, l = 2$ and $h = 4$, for a buffer of $n = 9$ values.

The final binarized motion mask M_t is obtained as a composition of the two binarized motion masks computed respectively using the low and the high thresholds.

Finally, the list MVO_t of moving objects at time t is extracted from M_t by grouping connected pixels. However, an object-level validation step is required to remove all the moving objects generated by small movement in the background, for example by moving tree branches. This validation takes into account the joint contribution of the gradient and color coherences of the objects.

The *gradient coherence* GC_t is evaluated over a $n \times n$ neighborhood as the block-wise minimum of absolute differences between the current gradient values G_t and the background gradient values GBG_t in a block. The gradient is computed with respect to both spatial and temporal coordinates of the image I_t. In the case of stationary points, the past image samples $I_{t-\Delta t}$ can be approximated with the background model BG_t and then the gradient module G_t is computed as the square sum of all the components:

$$\begin{aligned}
\frac{\partial I_t(i,j)}{\partial(x,t)} &= BG_t(i-1,j) - I_t(i+1,j) \\
\frac{\partial I_t(i,j)}{\partial(y,t)} &= BG_t(i,j-1) - I_t(i,j+1) \\
G_t &= \left\{ g(i,j) | g(i,j) = \sqrt{\left\|\frac{\partial I_t(i,j)}{\partial(x,t)}\right\|^2 + \left\|\frac{\partial I_t(i,j)}{\partial(y,t)}\right\|^2} \right\}
\end{aligned} \tag{3.3}$$

This joint spatio-temporal gradient module turns out to be quite robust to small movements in the background, mainly thanks to the use of a temporal partial derivative. Moreover, the joint spatio-temporal derivative makes the gradient computation more informative, since it also detects the non-zero gradient module even in the inner parts of the object and not only on the boundaries, as found in typical techniques.

Unfortunately, when the gradient module (either G_t or GBG_t) is close to zero, the data are not reliable. To overcome this problem, we combine the gradient coherence with the *color coherence* CC_t computed block-wise as the minimum of the Euclidean norm in the RGB space between the current image pixel color $I_t(i,j)$ and the background model values in the block centered at (i,j). The overall validation score is the normalized sum of the per-pixel validation score, obtained by multiplying the two coherence measures. Objects are validated by thresholding the overall coherence and pixels belonging to discarded objects are labeled as part of the background.

After being identified, moving objects should be tracked over time. For this purpose we used an appearance-based tracking algorithm called *Appearance Driven tracking with Occlusion Classification* (*Ad Hoc*) [49]. The tracking problem is formulated as a probabilistic Bayesian model, taking into account both motion and appearance. The probabilistic estimation is redefined at each frame and optimized as a MAP (Maximum A Posteriori) problem so that a single solution for each frame is obtained deterministically. We do not track each object separately but rather the whole object set is considered in the tracking in a two-step process: the first top-down step, provides an estimation of the best positions of all the objects, predicts their positions and optimizes them in a MAP algorithm according to the pixel's appearance and a specifically defined *probability of non-occlusion*. The second step is discriminative and bottom-up, and associates each observation point to the most probable object. Thus, the appearance model of each object is selectively updated at the pixel level in the visible part, thus ensuring high reactivity in shape changes. Further details can be found in [49].

3.2. Multiple camera processing

In large outdoor environments, multi-camera systems are required. In fact, distributed video-surveillance systems may exploit multiple video streams to enhance observation ability. Hence, the problem of tracking is extended from single to multiple cameras: people's shapes and status must be consistent not only in a single view, but also in space (i.e., observed by multiple views). This problem is known as *consistent labeling*, since identification labels must be consistent in time and space.

If the cameras fields of view (FoVs) overlap, consistent labeling can exploit geometry-based computer vision. This can be done with precise system calibration and 3D reconstruction can be used to resolve any ambiguity. However, this is not often feasible, in particular if the cameras are pre-installed and intrinsic and extrinsic parameters are not available. Thus, partial calibration or self-calibration methods can be adopted to extract only some of the geometrical constraints, e.g. to compute the ground-plane homography. We adopted a geometric approach called *HECOL* (Homography and Epipolar-based COnsistent Labeling) [6] that exploits cameras FoV relations and constraints to impose identity consistency.

Specifically, when cameras partially overlap, the shared portion of the scene is analyzed and people's identities are matched geometrically. First of all, an initial unsupervised and automatic training phase is employed to compute the overlapping regions among FoVs, ground-plane homographies and the epipole location for pairwise overlapping cameras. In this phase a single person moves into the scene and pairs of coordinates for the lower support points of the person (which lie on the ground plane $z = 0$) on two overlapped cameras are collected. Using SVD or LSQ methods the homography matrix between the two cameras ground planes can be easily estimated.

The consistent labeling problem is then solved online whenever a new object τ appears in the field of view of a given camera C^1. The multi-camera system must check whether τ corresponds to a completely new object or to one which is already present in the FoV of other cameras. Thus, the $\mathbf{lp}$ of each of the K objects in the overlapped camera C^2 is warped to the image plane of C^1 through the previously computed holography. The likelihood is then computed by testing the fit of each hypothesis against current evidence. The main goal is to distinguish between single hypothesis, group hypotheses and possible segmentation errors by exploiting only geometrical properties in order to avoid uncertainties due to color variation and adopting the vertical axis of the object as an invariant feature.

In addition, the axis of the object τ can be warped correctly only with the homography matrix and the knowledge of epipolar constraints among cameras. To obtain the correct axis inclination the vertical vanishing point (computed by a robust technique as described in [50]) is then used as shown in Fig. 1. The lower support point $\mathbf{lp}$ of τ is projected on camera C^2 by using the homography matrix. The corresponding point on the image plane of camera C^2 is denoted as $\mathbf{a_1} = H\mathbf{lp}$, where H is the homography matrix from C^1 to C^2. The warped axis will lie on a straight line passing through $\mathbf{vp^2}$ and $\mathbf{a_1}$ (Fig. 1d). The ending point of the warped axis is computed by using the upper support point $\mathbf{up}$, which is the middle point of the upper side of the objects bounding box. Since this point does not lie on the ground plane, its projection on the image of camera C^2 does not correspond to the actual upper support point; however, the projected point lies on the epipolar line. Consequently, the axis' ending point $\mathbf{a_2}$ is obtained as the intersection between the epipolar line $\langle \mathbf{e^2}, H\mathbf{up} \rangle$ and line $\langle \mathbf{vp^2}, H\mathbf{lp} \rangle$ passing through the axis.

Based on geometrical constraints, the warped axis $\langle \mathbf{a_1}, \mathbf{a_2} \rangle$ of τ in the image plane of C^2 is univocally identified but its computation is not error free. In order to improve robustness to computation errors, we also account for the dual process that can be performed for each of the K potential matching objects: the axis of the object in C^2 is warped on the segment $\langle \mathbf{a_1}, \mathbf{a_2} \rangle$ on camera C^1.

The measure of axis correspondence is not merely the distance between axes $\langle \mathbf{a_1}, \mathbf{a_2} \rangle$ and $\langle \mathbf{lp}, \mathbf{up} \rangle$; it is defined as the number of matching pixels between the warped axis and the foreground blob of the target object. This makes it easier to define a normalized value to quantify the matching.

In the end, the likelihood is defined as the maximum value between the *forward* and *backward* contributions, where the forward contribution refers to the case in which the measure is computed from the camera where the new object appears (camera C^1) to the image plane of the hypothesis (camera C^2), whereas the

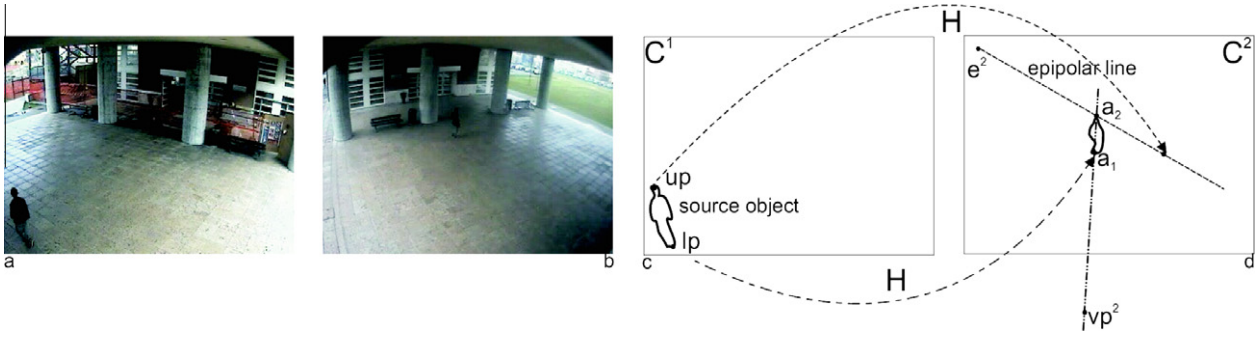


Fig. 1. Example of exploiting vanishing point and epipolar geometry to warp the axis of the object τ to the image plane of camera C^1.

backward contribution is computed similarly from the hypotheses space to the observed object [6]. The use of the maximum value ensures that the contribution where the extraction of support points is generally more accurate and suitable for the matching will be used. The effectiveness of the double backward/forward contribution is evident in the full characterization of groups of people. The forward contribution helps solve situations when a group of objects is already inside the scene while the group's components appear one at a time in another camera. By contrast the backward component is useful when two people appear in a new camera detected as a single blob. Group disambiguation can be solved by exploiting the fact that in the other camera the two objects are detected as separated. The backward contribution is also useful to solve the case of *segmentation errors* in which a person has been erroneously extracted by the object detection system as two separate objects, but a full view of the person exists from the past in an overlapped camera.

The algorithms described in this section have the scope to keep a person tracked when he/she moves in a wide and complex area. In other words, the video containing the moving people can be converted in a different time series for each moving person in it: the time series contains the trajectory T_j for the person j, composed by a sequence of (x,y) coordinates, rectified on the ground plane.

4. From time series to graphs

A possible indicator of potential threats is the fact that a person moves in an anomalous way, i.e. following an abnormal path: for instance, if someone moves in the opposite direction of a crowd, or if he/she follows a trajectory never (or rarely) seen before, etc. On the other hand, a system that raises an alarm for every trajectory which has not seen before will raise many false alarms. Hence, the system needs to recognize normal trajectories even though it has not "seen" them before. For this purpose, the trajectories obtained through the algorithms described in Section 3 need to be analyzed to detect anomalies. This problem is addressed here by embedding the training trajectories in a graph and then measuring the effect of a new trajectory relative to the reference graph. To achieve this, the first step is to transform the time series (trajectories) into a weighted graph, where a node v represents movement from one location in the scene to another, while the weight of edge $e_{i,j}$ is the probability to see the i movement followed by the j movement.

The direct use of the (x,y) samples is unfeasible since it will result in a very high number of nodes (as extreme case, square of the number of pixels), which severely compromises the use of graph-based approaches due to computational cost, as well as the robustness of the representation. Moreover, the (x,y) data are often affected by noise and tracking errors, and thus need to be filtered before use. The simplest solution to both problems is the quantization of image (x,y) plane, which in this context translates into dividing the scene into a fixed number of cells and assigning each data point to its containing cell. The naive scheme is to divide the scene using a fixed-size grid. Unfortunately, the grid size is a crucial parameter in this approach. Let $N \times M$ be the size of the image (bird-eye view of the complete scene taken from multiple cameras), and N_r and N_c be the number of rows and columns of the grid, respectively. The direct use of the coordinates would result in $(N \times M)^2$ nodes, reduced to $(N_r \times N_c)^2$ using a uniform grid: if N_r and N_c are too high (say $N_r = 100$ and $N_c = 100$ for a $N \times M = 1000 \times 1000$ image) the approximation is good but the computational load can still be too high (100,000,000 nodes in the example); if N_r and N_c are reduced (e.g., $(10 \times 10)^2 = 10{,}000$

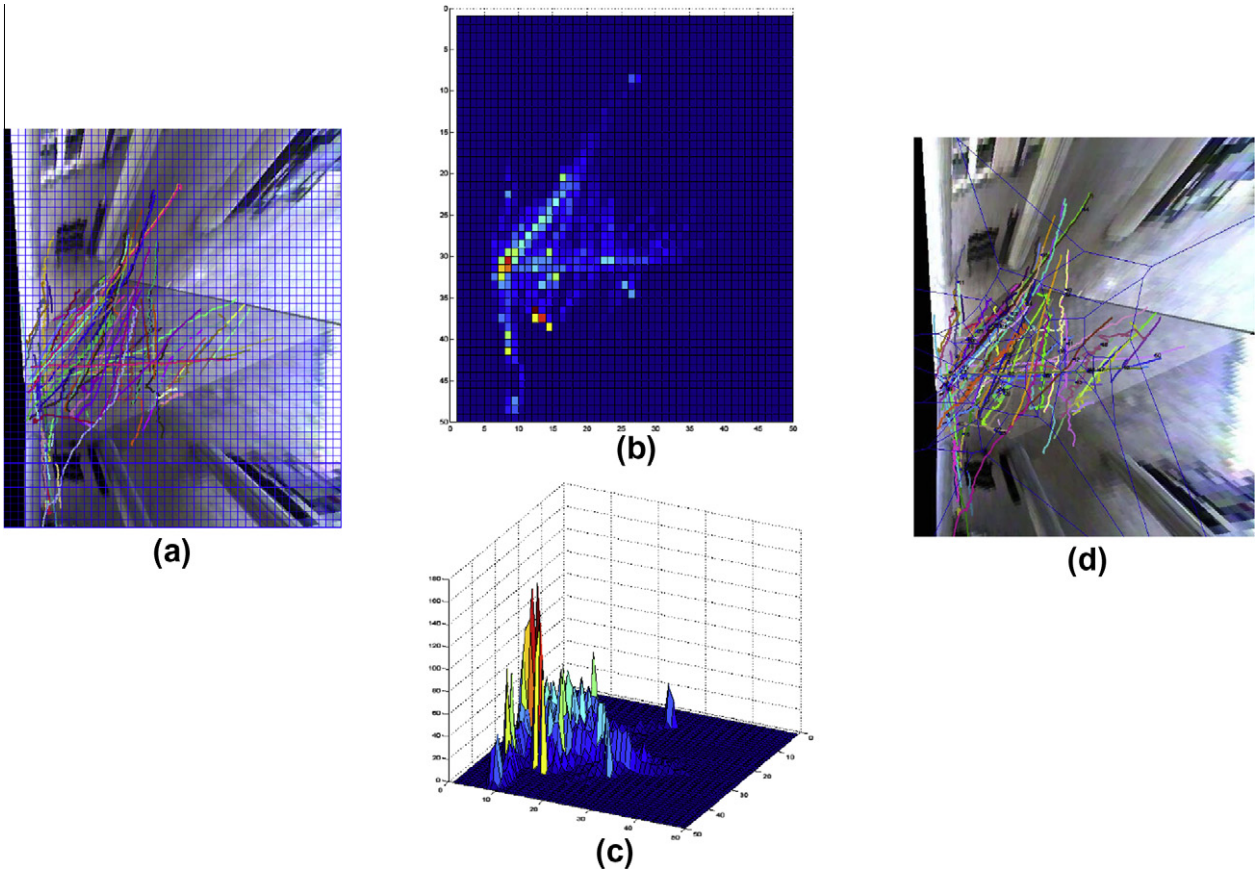


Fig. 2. Irregular partitioning of the image area through Voronoi diagrams: (a) Reports the first regular division of the image ($50 \times 50 = 2500$ cells in this example); (b) shows the top view of the 2D histogram, while (c) shows a side view; and (d) shows the resulting Voronoi diagram with 50 cells.

nodes) the complexity becomes more acceptable (but still not practical) at the price of the risk of having an overly coarse quantization of the data. Another disadvantage of a uniform grid is the uneven statistics of the cells occupation in natural scenes, yielding suboptimal statistical quantization of the trajectories.

These problems are tackled at two levels. First, we do not use the fixed-size regular-geometry grid scheme but rather a density-sensitive variable-geometry one. Second, graph nodes are assigned only to observed transitions. By not forcing any specific geometry of the cells (as in the case of a regular grid), the task of finding an adequate partition to cells is reduced to finding appropriate center-points. Having established the centers, the cells' boundaries are determined by the locus of points that are at the same distance from two centers, hence creating a Voronoi tessellation. In order to select the centers, certain properties can be considered: first, an area that is rarely traversed needs only a rough description; conversely, busy areas require a high resolution partitioning in order to distinguish between normal and abnormal walks. A natural solution is to use as centers points that are randomly sampled on the training trajectories, taking into account small sample size effects. In that way we can use fewer cells (i.e., nodes) but still maintain high resolution in the "most populated" areas.

The procedure is summarized in Fig. 2. Given a training set composed of normal trajectories, the image is first divided in $N_r \times N_c$ rectangular cells of fixed size (Fig. 2a). For this preliminary step, N_r and N_c are less critical parameters and can be high (we used 250×500 in our experiments). Using this division, a 2D histogram $\mathcal{H}$ can be built (Fig. 2b and c), where $\mathcal{H}(i,j)$ represents the number of trajectory points falling in the cell at row i and column j. It is worth noting that during this first step no tracking information is required and the trajectories are treated as a *set* of points instead of a *sequence* of points.

The 2D histogram represents a 2D distribution of the samples in the scene, with peaks (Fig. 2c) in the major areas (cells) of activity in the scene. To obtain the best coverage (related to the training data) and the most suitable partition of the scene with the fewest cells, they need to be distributed according to the discrete distribution described by $\mathcal{H}$. Thus, given N_{center} as the number of cells/nodes used, we draw N_{center} samples from the distribution approximated by $\mathcal{H}$, which increases the likelihood of sampling from peaks of $\mathcal{H}$, avoiding sampling from areas where no points are present in the training set. These N_{center} samples represent the seeds of a Voronoi tessellation of the scene (Fig. 2d). The adjacency map and the transition matrix are then computed on these cells, making the graph treatable in computational terms.

Having established the centers, each point is replaced by the center closest to it, so that the trajectories are transformed into a sequence of centers. A node is assigned to each observed transition from cell a to cell b. Let node i represent the transition from node a to b, and in the same manner j represents the transition from cell b to cell c. Hence, the edge $e_{i,j}$ will represent the occurrence of moving from cell a to cell b and then to cell c, while the weight of edge $e_{i,j}$ will be the probability of such a movement. Note that we only assign nodes to an observed transition, in order to minimize the graph dimensions (see above).

When transforming trajectories into a graph, the global description of the trajectory is lost. In order to overcome to this deficiency, we first create a second-order graph, thus creating nodes not for each cell but for each movement from cell to cell. Moreover, we sample the trajectory at several time scale to better catch the anomaly. In our tests each trajectory is first treated in every sample, then every 10 samples, every 20 samples, and so on until reaching a third of the trajectory length. This multi-level sampling allows a good generalization, not sensitive to normal changes, carrying the advantage of having a broad point of view on trajectories achieved by looking at different time scales.

This procedure transforms a collection of trajectories into a graph. When a new trajectory is encountered, the cell centers are determined according to the above scheme, but using the new trajectory together with all the normal ones. To guarantee that the new trajectory is represented in the graph, a fixed number of centers (in our trials 3) is sampled from the new trajectory, whereas the remainder (in our trials 47) are sampled from the normal trajectories. By using these centers a graph is constructed using the new trajectory, while another is built using only the normal trajectories. Once these two graphs have been constructed, anomalies are detected by searching for substantial differences between them, as described in the following section.

5. From graphs to anomalies

Anomalies detection entails two competing requirements: the first is to be able to detect anomalous instances and the second is to ignore normal instances. In contrast to most supervised learning problems, where different classes are explicitly considered, in this case we are only able to characterize one class (the normal class), and then define the abnormal class by exclusion. Therefore, a normal instance is defined as one that is similar to the previous normal instances. By defining the problem in this way, the crucial issue is to find a proper similarity function without assuming too much about the specific nature of the anomalies.

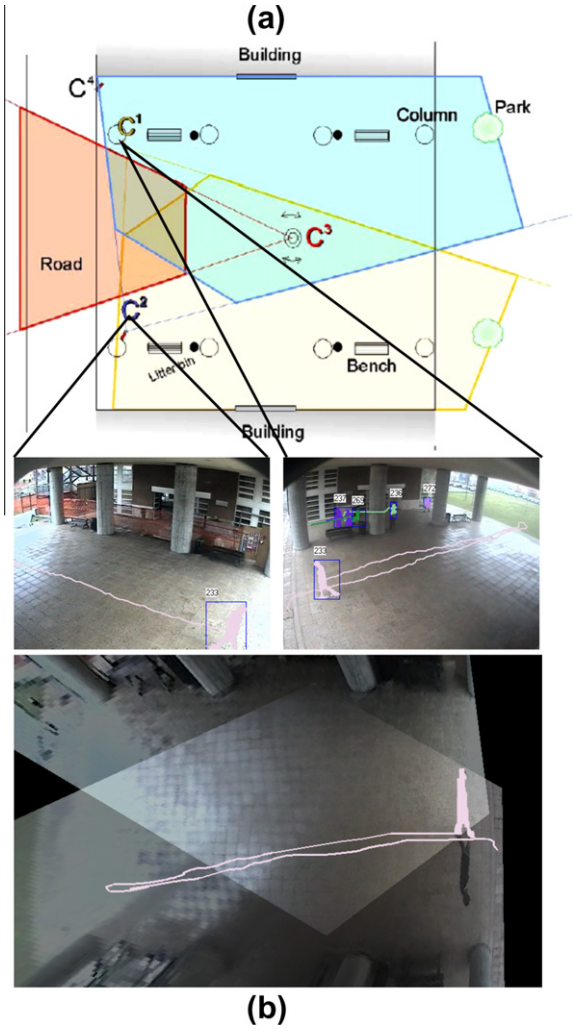


Fig. 3. (a) Map of the scenario used for testing. (b) Snapshots from the camera used for tracking and logging people trajectories and their projection on the ground plane map.

Our specific approach is motivated by the algebraic properties of similarity matrices proved highly successful in spectral clustering [13]. The procedure is explicitly described below.

5.1. The divergence measures

Let us first introduce some notations. Let $W \in \Re^{+^{N\times N}}$ be a symmetric matrix that represents the edge weight so $W_{i,j}$ is the weight of edge $e_{i,j}$. Let $A \in \Re^{N\times N}$ be a matrix, and denote the matrix eigenvalues as $\{\lambda_1,\ldots,\lambda_N\}$, ordered in increasing order. The corresponding eigenvector will be denoted as $\{\phi_1,\ldots,\phi_N\}$. We will represent the graphs as normalized Laplacian $L = I - D^{-\frac{1}{2}}WD^{-\frac{1}{2}}$, where D is a diagonal matrix such that $d_{i,i} = \sum_{j=1,\ldots,N} W_{i,j}$ and I is identity matrix. This can be justified by the result [11] that the eigenvectors of the Laplacian converge to the Laplace–Beltrami eigenfunction of the manifold. More intuitively, note that the Laplacian enforces the diffusion roll, i.e. the amount of input entering is equal to the amount leaving in all vertices.

Let us now define the variables k and δ.

Definition 1. Let $\lambda_1,\ldots,\lambda_N$ be the eigenvalues of L, then k and δ are defined as:

$$\delta = \max_i(\lambda_{i+1} - \lambda_i) \tag{5.1}$$

$$k = \arg\max_i(\lambda_{i+1} - \lambda_i) \qquad \square \tag{5.2}$$

i.e., k was defined as to maximize the spectral gap δ.

From the Laplacian, we consider only the first k eigenvectors $\Phi_{[1,\ldots,k]}$ corresponding to the smallest k eigenvalues. By doing so the non-significant (high-frequency) effects are removed and only the fundamental structure of the graphs is compared.

Given two graphs G and $\widetilde{G}$ the eigenvectors of one graph are projected on the other by means of the definition of the matrix M.

Definition 2. Let G and $\widetilde{G}$ be two graphs. Let Φ and $\widetilde{\Phi}$ be their corresponding eigenvector matrices, and k and $\tilde{k}$ be computed as reported in Definition 1. Then the matrix M is defined as:

$$M = \begin{cases} \Phi^T_{[1\cdots k]}\widetilde{\Phi}_{[1\cdots\tilde{k}]} & k \geqslant \tilde{k} \\ \widetilde{\Phi}^T_{[1\cdots\tilde{k}]}\Phi_{[1\cdots k]} & \text{else} \end{cases} \qquad \square \tag{5.3}$$

The singular values of Matrix M can be considered as the $\sin^2$ of angles between the two subspaces. These angles are called the canonical angels and are the standard invariant (under coordinate

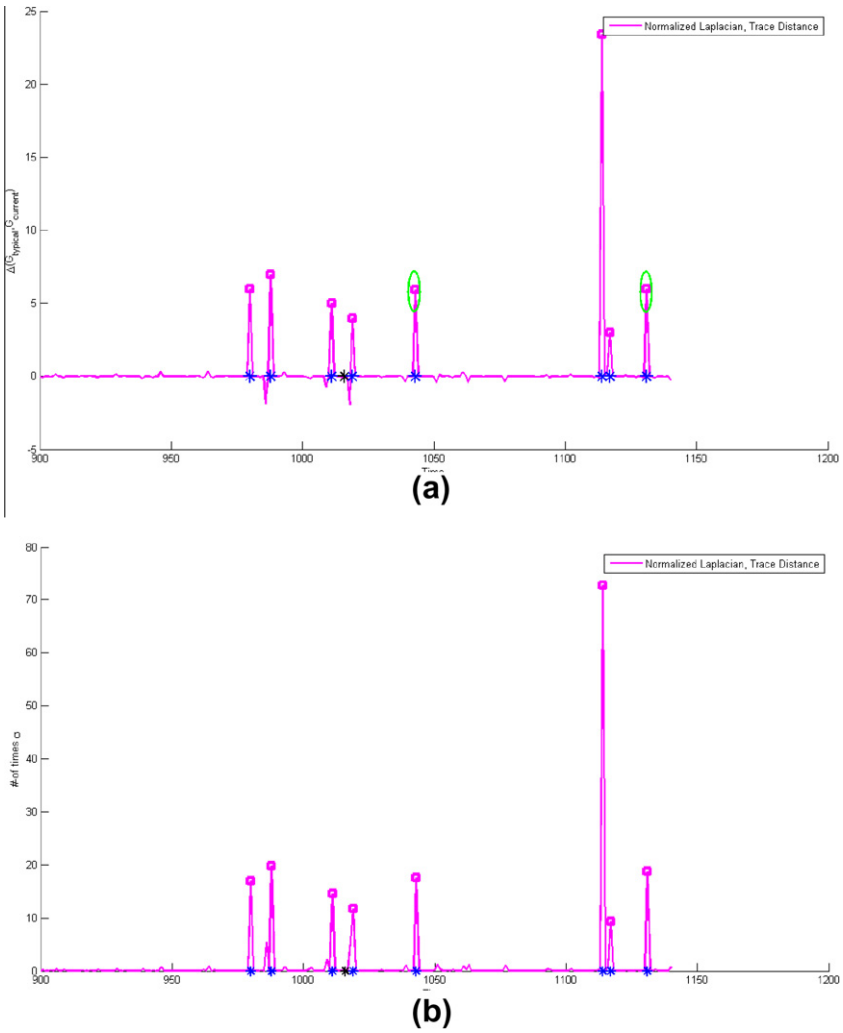


Fig. 4. (a) Plot of the temporal graph distance used to detect anomalies. The square indicates the time the system raised an alarm. The blue asterisk are the true positive ground truthed alarms while the black one represents a true negative. The green circles refer to the anomalous situations of Fig. 5. (b) Plot of the number of standard divergence error from the learned mean $-\frac{|A-E[A]|}{\sqrt{Var[A]}}$.

changes) way to measure scale invariant distances between linear subspaces. We suggest two ways to apply the singular values in our context. The first is summing their squared values (the trace measure below) and the second taking the product of their squared value (the determinant measure below). Both are invariant measures that reflect the geometrical overlap of the two subspaces.

Definition 3. Given two graphs G and $\widetilde{G}$ and the corresponding matrix M as defined in Definition 2, the *determinant divergence* is defined as:

$$\Delta(G,\widetilde{G}) = 1 - \sqrt{det(M^T M)} \qquad \square \tag{5.4}$$

This measure incorporates a geometric intuition. Namely, the eigenvalues of M may be regarded as $\cos\theta$, where θ denotes the angles between the subspaces. The natural way to proceed is to find the volume of the parallelotope created by the cosine of the angles. Finding the volume of a k-dimensional parallelotope is achieved by calculating the Gramian matrix of M, i.e. the inner product between all the vectors of a matrix. The volume is calculated by taking the square-root of the Gramian matrix's determinant, which results in $\sqrt{det(M^T M)}$.

Definition 4. Given two graphs G and $\widetilde{G}$, the corresponding matrix M as defined in Definition 2, and defining $\hat{k} = \min(k, \tilde{k})$, the *trace divergence* is defined as:

$$\Delta(G,\widetilde{G}) = \hat{k} - Trace(M^T M) \qquad \square \tag{5.5}$$

To understand this measure, it is worth recalling the well-known Frobenius norm of a matrix A : $\|A\|_{fro} = \sqrt{Trace(A^T A)} = \sqrt{\sum_{i=1}^{N} \lambda_i^2}$, where λ_i is the ith singular value. Since the singular value of M represents the angles between the subspaces, it is evident that in the case of equality between the graphs, all angles will be zero, which means that the cosines will be 1 and their sum will equal k, i.e. the minimum between the two dimensions of the subspace.

With respect to the existing approaches, our method assumes the mapping between graph nodes to be known and exploits this

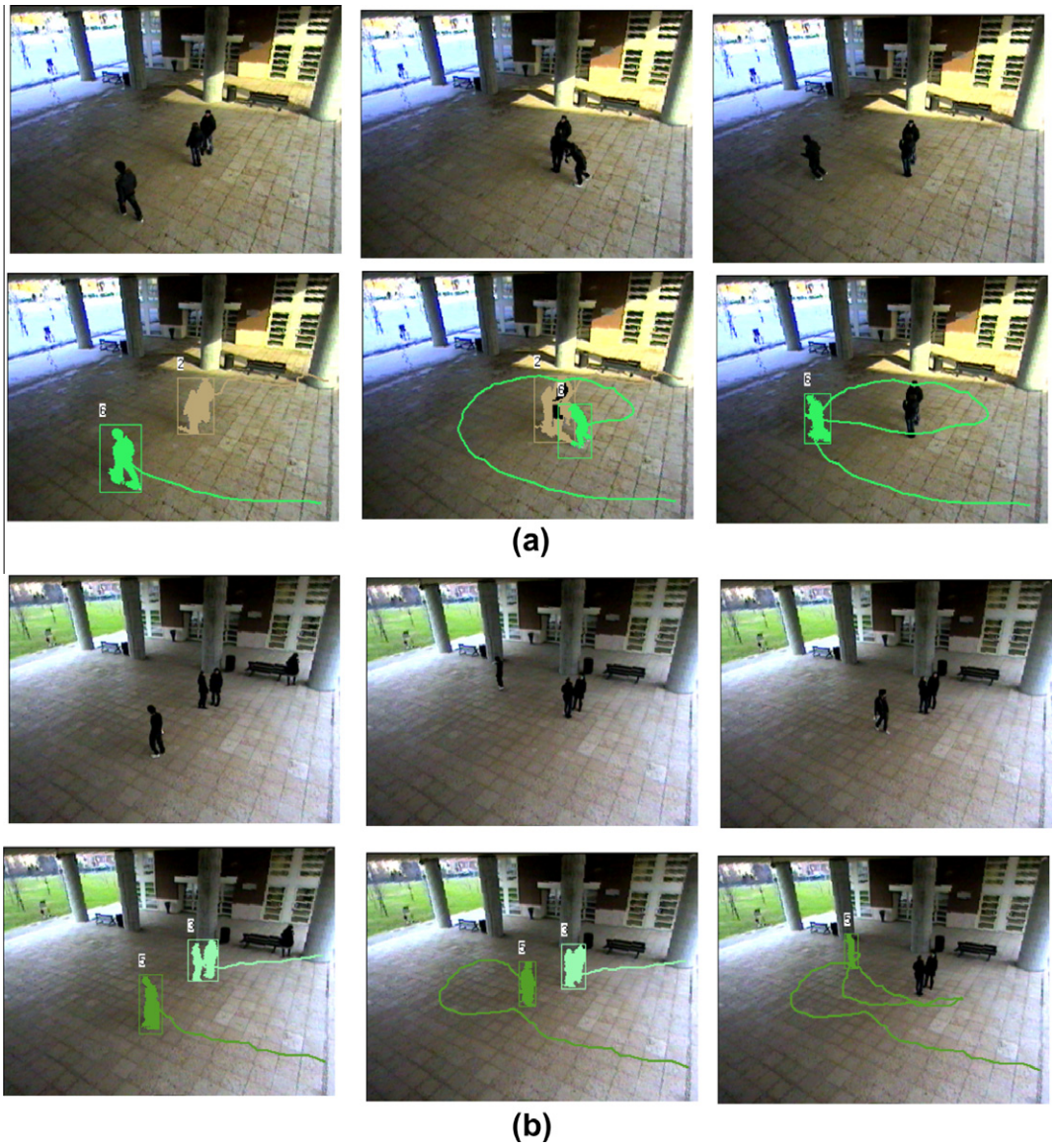
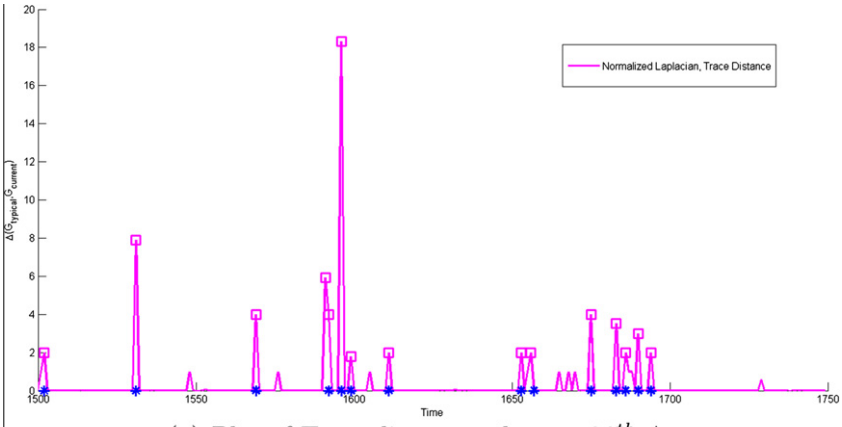
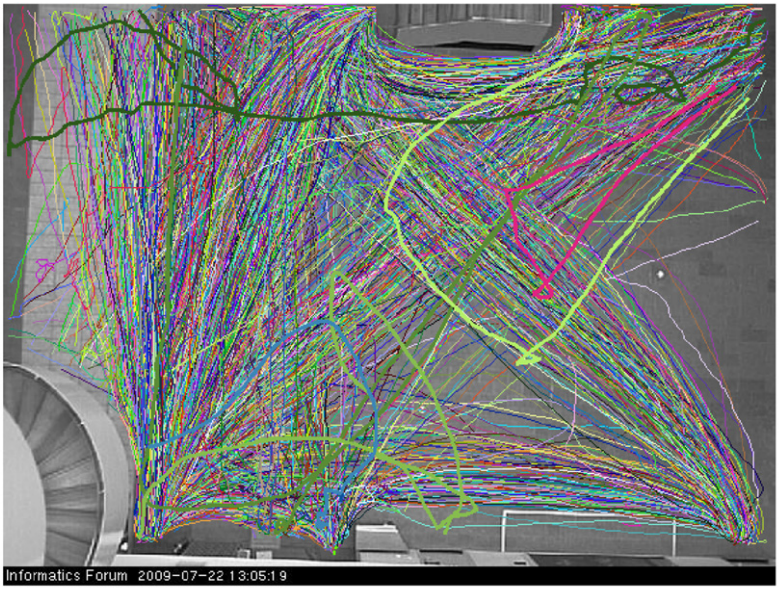


Fig. 5. Two different examples of typical anomalous situations that may occur in urban scenarios. In (a) a theft is stealing a bag and running away. In (b) a person is moving suspiciously around people walking.



(a) Plot of Trace distance - dataset 26^{th} Aug



(b) Map of trajectories - dataset 26^{th} Aug

Fig. 6. Results on Edinburgh Informatics Forum Pedestrian Database – dataset 26th August.

information to employ a distance measure more suitable to the problem. This measure eliminates the existing drawbacks of other methods (e.g., graph edit distance), such as their sensitivity to scaling on the edges' values and that they cannot distinguish between probable and improbable changes. Regarding the first issue, our approach results to be invariant to scaling, since it ignores the exact eigenvalues of the adjacency matrix, while focusing on the angles between subspaces induced by the Laplacian eigenvectors.

Speaking of the second issue, since we assume that pedestrians follow the same paths, our graph will create several clusters, one for each of such path. On the one hand, given our case scenario, it is highly likely that people do not follow exactly (step-by-step) the same path, also given the uncertainty in the extraction of the trajectories. This type of deviation from normal paths must be properly addressed and ignored. On the other hand, if a pedestrian crosses between two normal paths (generating a never-seen-before path), our system must detect this as an anomalous path. Our method will deal correctly with both cases: in the first case, there will be no change in the path cluster hence no change in the first eigenvectors; instead, the second case will cause the connection between two path clusters (i.e. the change of the first eigenvectors).

5.2. The algorithm

This section summarizes the algorithm used to detect anomalies in people trajectories.

```
Detecting(a, learn){
counter = 0;
T = getTrajectory();
Tr_normal{counter} = T;
  while(1){
   alarm = 0;
    T = getTrajectory();
    [El,E2] = fromTrajectoryToGraph(Tr_normal,T);
    dis = getDistance(El,E2);
    if(dis-m > a*(sqrt(v)) && counter > learn){
      alarm = abnormal
    }else{
    counter++;
    Tr_normalcounter = T
    d = dis − m;
    m = m + d/counter;
```

(*continued on next page*)

```
      v = (v*(counter - 1) + d*(dis - m))/counter;
      }
        if(alarm == abnormal){
          raise an alarm;
        }
    }
  }
```

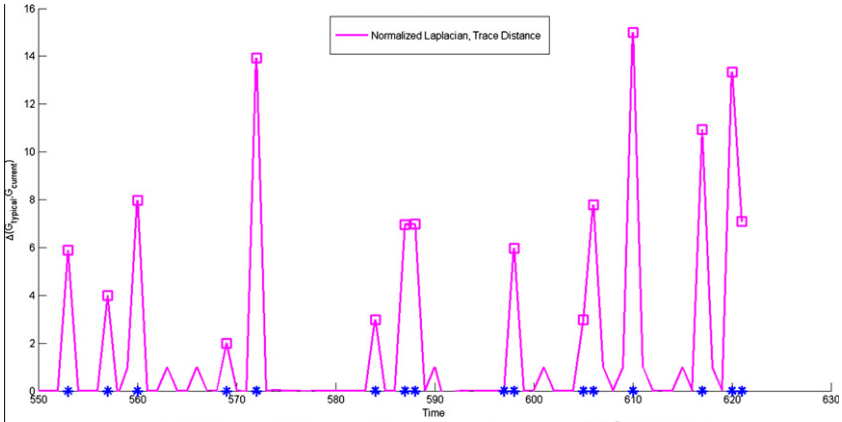
This algorithm is constructed in three main steps. The first step consists of processing the video data and transforming them into a time series: this step is performed by the function `getTrajectory()`, which corresponds to the algorithms described in Section 3. This function can either return one complete trajectory, or a set of trajectories (not necessarily complete). The second step consists of the transformation of the time-series data into a graph representation (method `fromTrajectoryToGraph(Tr_normal,T)` described in Section 4): in particular, one graph represents only the normal trajectories (the reference graph), while the other contains the new (test) trajectory (or trajectories). Finally, the third step implements one of the divergence functions which we have defined in Section 5.1.

The algorithm also has two parameters, `a` and `learn`. The latter defines the period of time necessary for the system to "learn normality", while the former controls the tradeoff between *sensitivity* and *specificity*, i.e. if the distance between the two graphs is larger than the mean distance by `a` times the standard error (which means that we assume a Gaussian distribution) we can decide whether this sample is anomalous or not.

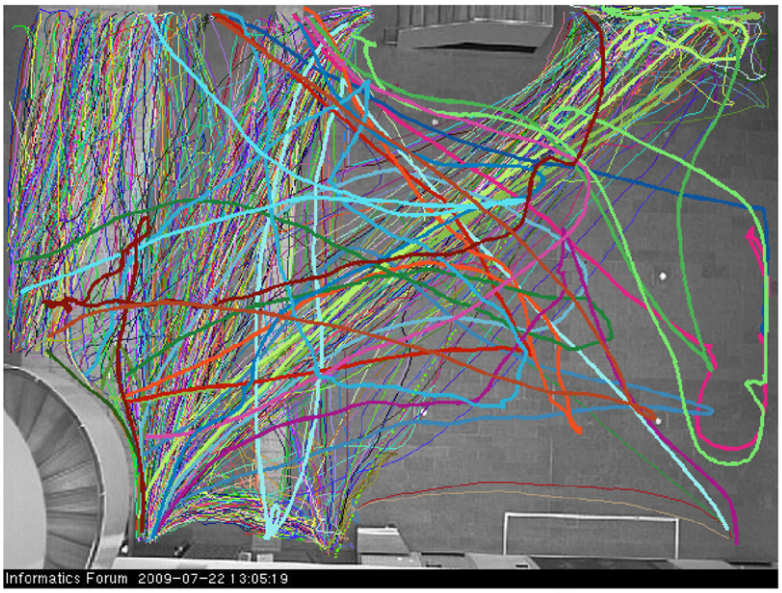
When a sample is found normal, we keep updated the perception of normal behavior by consequently updating the mean (`m`) and variance (`v`) of the distance measure.

As mention above our distance measures are motivated by the observation that trajectories (as other data) can be divided into several groups. In the considered scenario we found this assumption valid observing that anomalous trajectories created new groups. This can be intuitively seen when looking at the scale on *y*-axis of Fig. 4.

The choice of the trace distance measure instead of the determinant one can be motivated recalling the geometric meaning of the measure. In fact, the intuition for selecting the determinant measure, as mentioned above, is to evaluate the volume of the parallelotope. When a new cluster is formed, a new eigenvector is added to the selected eigenvectors (thus, *k* will grow). As a consequence,



(a) Plot of Trace distance - dataset 14^{th} Jul P1



(b) Map of trajectories - dataset 14^{th} Jul P1

Fig. 7. Results on Edinburgh Informatics Forum Pedestrian Database – dataset 14th July P1.

since the projection of this new eigenvector on the former set will be zero, thus the volume of the parallelotope will be zero too. If more than one cluster is formed, the volume will still be zero, resulted in the inability to discern between the number of new clusters. Consequently, the trace distance is preferred given its capability to discern between the number of new clusters.

Moreover in this scenario a simpler distance measure, such as the frobenius norm, may be sufficient but we strongly believe our proposal is more general and can detect more challenging anomalies with which simpler norms will fail.

6. Experimental results

Detecting anomalies in video surveillance data depends crucially on the reliability of the feature extracted and the anomaly detection techniques used.

We tested the system on a corpus of trajectories acquired using the automatic video-surveillance system composed respectively of the single camera processing model, described in Section 3.1, and the data fusion model of Section 3.2. The system is capable of reliably detecting and tracking people from multiple overlapping cameras under severe occlusion and different environmental conditions, and can acquire and log both target trajectories and appearances, and eventually project trajectories onto the ground plane to correct distortions due to perspective.

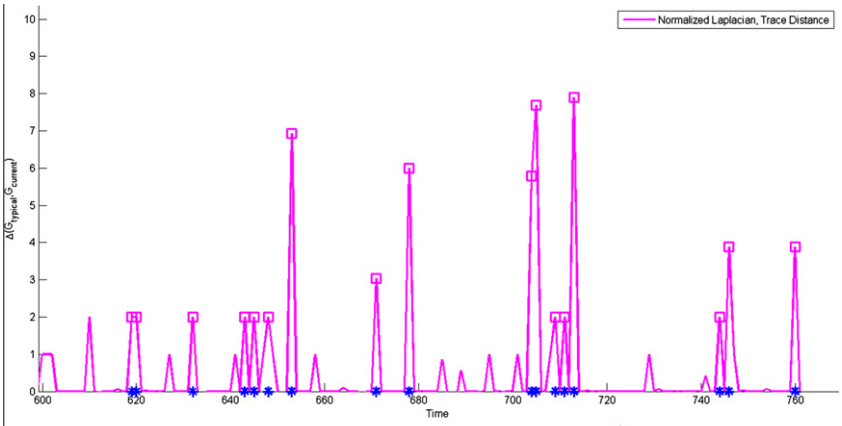
The experiments were carried out on a real scenario as a test bed and people's trajectories were collected for a month. We focused on two-camera setup at the University of Modena and Reggio Emilia, see Fig. 3; in this scenario, on ordinary working days, many people typically walk in the scene following recurrent paths.

Modeling the behaviors of pedestrians in real world scenes is likely to fail, because every scene has its peculiar normal paths and associated behavioral patterns, moreover the same scene can have various normal behaviors at different time of the day.

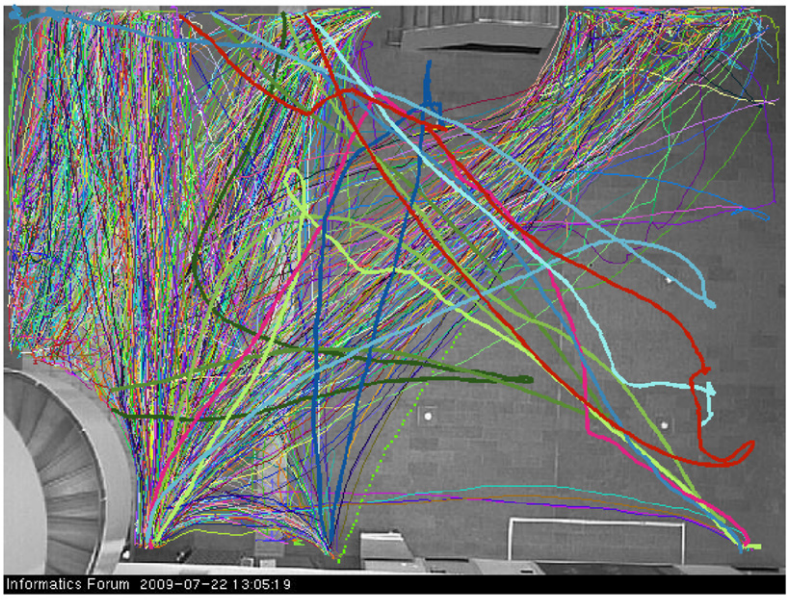
Hence, as mentioned above, we decided to focus on a scene-centric approach that models the scene as a whole by a graph of movements.

For assessing the capabilities of our detection algorithm, a corpus of 1150 trajectories was collected by the surveillance system over a period of several hours every day for more than a month; among them 1131 were retained based on minimum length criterion.

This corpus presents some challenging characteristics: the presence of fragmented trajectories, a huge variety of different paths,



(a) Plot of Trace distance - dataset 14^{th} Jul P2



(b) Map of trajectories - dataset 14^{th} Jul P2

Fig. 8. Results on Edinburgh Informatics Forum Pedestrian Database – dataset 14th Jul P2.

the presence of two different people going on the same path in opposite direction and the unpredictability of paths neither guided or preformed by actors.

The anomaly detection is preceded by a learning stage where the normal and most frequents behaviors of the scene are estimated in conjunction with the mean and variance of the adopted similarity measure. Hence we divided our normal trajectories in two sets: the first 900 trajectories were used in the training phase, while the other 231 normal trajectories were used in the test phase.

To verify the proposed anomaly detection framework we manually selected nine abnormal trajectories that were randomly scattered among the normal ones constituting the test set. In the considered abnormal behaviors, six uncommon path types, that consist of completely new or poorly followed trajectories, are present. Additionally, two sequences are of particular interest because represent the typical behaviors that must be signaled and spotted in an automatic video-surveillance systems: in the first one a theft is stealing a bag and running away (Fig. 5a), while in the second one a person is moving suspiciously around people walking (Fig. 5b).

Fig. 4 plots the Trace distance measure (see Section 5.1) for the test set. The nine ground-truthed anomalous events are marked with an asterisk, the blue ones underline correctly detected events while the black one missed detection. Additionally the green circles refer to the anomalous situations described in Fig. 5.

The system correctly detected 8 out of the 9 abnormal trajectories in total absence of false positive alarms. (times in which the system declares as abnormal are marked by square on the distance line).

These results suggest that this new proposed anomaly detection framework is able to detect anomalies in a complex scenarios, such as the one considered when the data present structure complexity and noise due to the automatic tracking techniques. It is nevertheless important to emphasize that good performances depend strongly on stable and long trajectories that can be extracted using the surveillance system previously described.

To extensively evaluate the capability of the system in detecting anomalies we tested the proposed anomaly detection framework on publicly-available surveillance data from Edinburgh Informatics Forum Pedestrian Database (http://homepages.inf.ed.ac.uk/rbf/FORUMTRACKING/). This dataset contains several days of people trajectories taken from a bird-eye view camera.

We chose to perform three different sets of tests on the dataset 26th of August and 14th of July, and the latter was split in two parts, P1 and P2 to test the capability of the system to work with both a large amount of training data (26th August) and a smaller amount (14th July, P1 and P2). A subset of the trajectories of the three datasets has been manually ground truthed and classified by experts as normal or abnormal.

The tests have been performed evaluating the Laplacian Trace distance measure of equation (5.5) and results are shown in Figs. 6–8a, where ground-truthed anomalous trajectories are marked with an asterisk and detected ones with a square. The system was able to detect 46 anomalies out of 47 in a corpus of 3131 trajectories in the three datasets. Qualitative examples of normal and abnormal people paths in this scenario are given in Figs. 6–8b.

Our successful application of the spectral graph-theoretical framework is yet another demonstration to the power of statistical pattern analysis on graphs and the robustness of the graph Laplacian based algorithms. The main reason for the success in this case relies in the simple representation of the trajectories in the graph nodes and weights. The relative ease with which these algorithms detected the anomalies in our tests call for further studies with more crowded scenes and more ambiguous tracking, such as in airports or train stations.

7. Conclusions

Automatic detection of suspicious behavior in surveillance video cameras is clearly becoming a major technological challenge for the world's security systems. In this work we addressed one important component of this problem – detecting anomalous people's trajectories by stationary overlapping cameras. We combined two state-of-the art complementary methodologies. The first one is a combination of tracking algorithms developed at University of Modena and Reggio Emilia, and the second is a novel application of algebraic graph theoretical methods in unsupervised statistical machine learning, developed at Hebrew University of Jerusalem.

We combine the two methods by a sequence transformations and statistical techniques. First, we discretize the tracked trajectories, using a Voronoi tessellation of randomly sampled points on the training trajectories. This gives us a robust and statistically efficient representation, without making any assumption on the geometry of the scene or the nature of the trajectories. In the second step, we represent transitions between adjacent Voronoi cells as nodes in the graph, and a sequence of two consecutive transitions as weights on the graph edges. This graphical representation is sensitive to both the locations and dynamics of people's movements, again without any specific modeling assumption. In the third step we filter the data by projecting this weighted symmetric adjacency matrix on the low-frequency eigenvectors of its graph Laplacian. This idea is motivated by the success of spectral clustering algorithms. Notice, however, that we do not perform any clustering here, but rather filter-out the high frequency noisy components of the trajectories in a clear analogy with lowpass filtering by Fourier components in signal processing. This Laplacian filtering further increase the robustness of the representation, with the sole assumption that the larger connected components of the trajectory graph are better characteristic of normal behavior. Another important idea is to perform this analysis on multiple time scales, or different powers of the adjacency matrix. This allows detecting anomalies in all scales of a new trajectory. Namely, anomaly can be in one unusual step, or in an unusual combination of long normal segments. Both types of anomalous trajectories can be detected using the same principles, but on different time scales of the same process. The final algorithmic novelty is in our choice of the similarity measure. Here again we rely on the application of geometrical invariants, in this case the canonical angles between subspaces, as represented by the reference Laplacian eigenvectors. We showed that several invariants measures that are based on the angles are useful for detecting anomalous trajectories on the basis of our tracking algorithms.

Our successful proof-of-concept calls for extensions and further work in two related and highly needed directions: less reliable tracking and more crowded scenes. We believe that the success of these methods as demonstrated in this work will follow by further progress in addressing these important extensions.

Acknowledgment

The authors would like to thank the support of the NATO Science for Peace Grant (Number 982480).

References

[1] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man, and Cybernetics – Part C 34 (3) (2004) 334–352.
[2] C. Stauffer, W. Grimson, Learning pattern of activity using real-time tracking, IEEE Transactions on PAMI 22 (8) (2000) 747–757.
[3] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, Detecting moving objects, ghosts and shadows in video streams, IEEE Transactions on PAMI 25 (10) (2003) 1337–1342.

[4] F. Jiang, Y. Wu, A. Katsaggelos, Detecting contextual anomalies of crowd motion in surveillance video, in: Proc. of IEEE Int'l Conference on Image Processing, 2009, pp. 1117–1120.

[5] S. Khan, M. Shah, Consistent labeling of tracked objects in multiple cameras with overlapping fields of view, IEEE Transactions on PAMI 25 (10) (2003) 1355–1360.

[6] S. Calderara, R. Cucchiara, A. Prati, Bayesian-competitive consistent labeling for people surveillance, IEEE Transactions on PAMI 30 (2) (2008) 354–360.

[7] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, S. Maybank, A system for learning statistical motion patterns, IEEE Transactions on PAMI 28 (9) (2006) 1450–1464.

[8] J. Li, C.-S. Chua, Y.-K. Ho, Color based multiple people tracking, in: Seventh International Conference on Control, Automation, Robotics and Vision, ICARCV 2002, Singapore, 2–5 December 2002, Proceedings, 2002, pp. 309–314.

[9] C. Madden, E.D. Cheng, M. Piccardi, Tracking people across disjoint camera views by an illumination-tolerant appearance representation, Machine Vision and Applications 18 (3–4) (2007) 233–247.

[10] R. Vezzani, D. Baltieri, R. Cucchiara, Pathnodes integration of standalone particle filters for people tracking on distributed surveillance systems, in: Proceedings of 25th International Conference on Image Analysis and Processing (ICIAP2009), Vietri sul Mare, Salerno, Italy, 2009.

[11] S. Umeyama, An eigendecomposition approach to weighted graph matching problems, IEEE Transactions on PAMI 10 (5) (1988) 695–703.

[12] I. Gutman, W. Xiao, Generalized inverse of the Laplacian matrix and some applications, Bulletin: Classe des sciences mathe'matiques et natturalles – Sciences mathe'matiques 129 (2004) 15–23.

[13] U. von Luxburg, A tutorial on spectral clustering, Statistics and Computing 17 (4) (2007) 395–416.

[14] H. Buxton, Learning and understanding dynamic scene activity: a review, Image and Vision Computing 21 (1) (2003) 125–136.

[15] A. Adam, E. Rivlin, I. Shimshoni, D. Reinitz, Robust real-time unusual event detection using multiple fixed-location monitors, IEEE Transactions on PAMI 30 (2007) 555–560.

[16] P. Cui, L. Sun, Z.-Q. Liu, S. Yang, A sequential monte carlo approach to anomaly detection in tracking visual events, in: Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition, 2007.

[17] H. Zhong, J. Shi, M. Visontai, Detecting unusual activity in video, in: Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition, vol. 2, 2004, pp. 819–826.

[18] M.T. Chan, A. Hoogs, J. Schmiederer, M. Petersen, Detecting rare events in video using semantic primitives with HMM, in: Proc. of Int'l Conference on Pattern Recognition, 2004, pp. 150–154.

[19] J.M. Gryn, R.P. Wildes, J.K. Tsotsos, Detecting motion patterns via direction maps with application to surveillance, in: WACV-MOTION '05: Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05), vol. 1, 2005, pp. 202–209.

[20] T. Xiang, S. Gong, Video behavior profiling for anomaly detection, IEEE Transactions on PAMI 30 (2007) 893–908.

[21] A. Basharat, A. Gritai, M. Shah, Learning object motion patterns for anomaly detection and improved object detection, in: Proc. of IEEE Int'l Conference on Computer Vision and Pattern Recognition, vol. 0, 2008, pp. 1–8.

[22] I. Saleemi, K. Shafique, M. Shah, Probabilistic modeling of scene dynamics for applications in visual surveillance, IEEE Transactions on PAMI 31 (2008) 1472–1485.

[23] S. Calderara, A. Prati, R. Cucchiara, Hecol: homography and epipolar-based consistent labeling for outdoor park surveillance, Computer Vision and Image Understanding 111 (1) (2008) 21–42.

[24] A. Prati, S. Calderara, R. Cucchiara, Using circular statistics for trajectory analysis, in: Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR 2008), 2008.

[25] Y. Zhou, S. Yan, T. Huang, Detecting anomaly in videos from trajectory similarity analysis, in: IEEE International Conference on Multimedia and Expo 2007, 2007, pp. 1087–1090.

[26] A. Mecocci, M. Pannozzo, A completely autonomous system that learns anomalous movements in advanced videosurveillance applications, in: Proc. of ICIP, Vol. 2, 2005, pp. 586–589.

[27] I. Junejo, O. Javed, M. Shah, Multi feature path modeling for video surveillance, in: Proc. of ICPR, vol. 2, 2004, pp. 716–719.

[28] R. Sillito, R. Fisher, Semi-supervised learning for anomalous trajectory detection, in: Proc. of British Machine Vision Conference BMVC, 2008.

[29] C. Piciarelli, C. Micheloni, G. Foresti, Trajectory-based anomalous event detection, IEEE Transactions on Circuits and Systems for Video Technology 18 (11) (2008) 1544–1554.

[30] A. Lakhina, M. Crovella, C. Diot, Diagnosing network-wide traffic anomalies, in: Proceedings of the 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, ACM, New York, NY, USA, 2004, pp. 219–230.

[31] A. Lakhina, M. Crovella, C. Diot, Characterization of network-wide anomalies in traffic flows, in: Proceedings of the 4th ACM SIGCOMM conference on Internet measurement, ACM, New York, NY, USA, 2004, pp. 201–206.

[32] L. Huang, X. Nguyen, M. Garofalakis, M. Jordan, A. Joseph, N. Taft, In-network PCA and anomaly detection, Advances in Neural Information Processing Systems 19 (2007) 617.

[33] T. Ide, H. Kashima, Eigenspace-based anomaly detection in computer systems, in: Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining, 2004, pp. 440–449.

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

[35] F. Bach, M. Jordan, Learning spectral clustering, Advances in Neural Information Processing Systems (NIPS), vol. 16, MIT Press, 2004, pp. 305–312.

[36] O. Duchenne, J.-Y. Audibert, R. Keriven, J. Ponce, F. Segonne, Segmentation by transduction, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2008), 2008, pp. 1–8. doi:10.1109/CVPR.2008.4587419.

[37] N. Thome, D. Merad, S. Miguet, Learning articulated appearance models for tracking humans: a spectral graph matching approach, Signal Processing: Image Communication 23 (10) (2008) 769–787, doi:10.1016/j.image.2008.09.003. <http://www.sciencedirect.com/science/article/B6V08-4TN0KTR-1/2/ec8925f5f4d9cccf885bc71e31d126ff>.

[38] D.K. Hammond, P. Vandergheynst, R. Gribonval, Wavelets on graphs via spectral graph theory, Applied and Computational Harmonic Analysis, in press (Corrected Proof). doi:10.1016/j.acha.2010.04.005. <http://www.sciencedirect.com/science/article/B6WB3-4YYGH6T-1/2/a938fa59304fa0016915cf1a623448f7>.

[39] Y. Weiss, A. Torralba, R. Fergus, Spectral Hashing, in: NIPS, 2008.

[40] M.A. Eshera, K.-S. Fu, An image understanding system using attributed symbolic representation and inexact graph-matching, Pattern Analysis and Machine Intelligence, IEEE Transactions on PAMI-8 (5) (1986) 604–618, doi:10.1109/TPAMI.1986.4767835.

[41] D. Knossow, A. Sharma, D. Mateus, R. Horaud, Inexact matching of large and sparse graphs using Laplacian eigenvectors, in: A. Torsello, F. Escolano, L. Brun (Eds.), Graph-Based Representations in Pattern Recognition, Lecture Notes in Computer Science, vol. 5534, Springer, Berlin/Heidelberg, 2009, pp. 144–153.

[42] R.C. Wilson, E.R. Hancock, B. Luo, Pattern vectors from algebraic graph theory, IEEE Transactions on PAMI 27 (2005) 1112–1124.

[43] B. Luo, E. Hancock, Structural graph matching using the em algorithm and singular value decomposition, IEEE Transactions on PAMI 23 (2001) 1120–1136.

[44] B. Xiao, E.R. Hancock, R.C. Wilson, Graph characteristics from the heat kernel trace, Pattern Recognition 42 (11) (2009) 2589–2606.

[45] A.Y. Ng, M.I. Jordan, Y. Weiss, On spectral clustering: analysis and an algorithm, Advances in Neural Information Processing Systems, vol. 14, MIT Press, 2001, pp. 849–856.

[46] R.I. Kondor, J.D. Lafferty, Diffusion kernels on graphs and other discrete input spaces, in: ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002, pp. 315–322.

[47] R.C. Wilson, P. Zhu, A study of graph spectra for comparing graphs and trees, Pattern Recognition 41 (9) (2008) 2833–2841, doi:10.1016/j.patcog.2008.03.011. <http://www.sciencedirect.com/science/article/B6V14-4S39MK9-1/2/9e8d14adb7e3b11ea7e22a3486ef900c>.

[48] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: principles and practice of background maintenance, in: Seventh International Conference on Computer Vision (ICCV 1999), 1999, pp. 255–261.

[49] R. Vezzani, R. Cucchiara, Ad-hoc: appearance driven human tracking with occlusion handling, in: First International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS'2008), in Conjunction with BMVC 2008, 2008.

[50] C. Brauer-Burchardt, K. Voss, Robust vanishing point determination in noisy images, in: Proc. of Int'l Conference on Pattern Recognition, vol. 1, 2000, pp. 559–562.