Moving pixels in static cameras: detecting dangerous situations due to environment or people

Simone Calderara[1], Rita Cucchiara[1], and Andrea Prati[2]

[1] D.I.I. - University of Modena and Reggio Emilia
[2] Di.S.M.I. - University of Modena and Reggio Emilia

Summary. Dangerous situations arise in everyday life and many efforts have been lavished to exploit technology to increase the level of safety in urban areas. Video analysis is absolutely one of the most important and emerging technology for security purposes. Automatic video surveillance systems commonly analyze the scene searching for moving objects. Well known techniques exist to cope with this problem that is commonly referred as "change detection". Every time a difference against a reference model is sensed, it should be analyzed to allow the system to discriminate among a usual situation or a possible threat. When the sensor is a camera, motion is the key element to detect changes and moving objects must be correctly classified according to their nature. In this context we can distinguish among two different kinds of threat that can lead to dangerous situations in a video-surveilled environment. The first one is due to environmental changes such as rain, fog or smoke present in the scene. This kind of phenomena are sensed by the camera as moving pixels and, subsequently as moving objects in the scene. This kind of threats shares some common characteristics such as texture, shape and color information and can be detected observing the features' evolution in time. The second situation arises when people are directly responsible of the dangerous situation. In this case a "subject" is acting in an unusual way leading to an abnormal situation. From the sensor's point of view, moving pixels are still observed, but specific features and time-dependent statistical models should be adopted to learn and then correctly detect unusual and dangerous behaviors. With these premises, this chapter will present two different case studies. The first one describes the detection of environmental changes in the observed scene and details the problem of reliably detecting smoke in outdoor environments using both motion information and global image features, such as color information and texture energy computed by the means of the Wavelet transform. The second refers to the problem of detecting suspicious or abnormal people behaviors by means of people trajectory analysis in a multiple cameras video-surveillance scenario. Specifically, a technique to infer and learn the concept of normality is proposed jointly with a suitable statistical tool to model and robustly compare people trajectories.

1 Introduction

Dangerous situations arise in everyday life and many efforts have been lavished in new technologies to increase the level of safety in urban areas. Everyday threats are of different nature and a reliable solution to prevent all them is a chimera. Nevertheless, even though a "solution" is not possible, an aid to ease the work of public officers is still technologically feasible. The aim of this aid is to pop up the attention on significant parts of the scene, to stores evidences of relevant events, to enhance the image/sound quality for investigation purposes, and so on.

Many media have been explored and effectively exploited to prevent criminal acts or to react to dangerous situations. Among these, it is worth to mention how sound or images can be exploited to successfully detect dangerous events in everyday life. Many applications exist, ranging from sound analysis to image retrieval, that are currently adopted for forensic purposes. For instance, several approaches to computational audio analysis have been proposed, mainly focused on the computational translation of psychoacoustics results, such as the so-called *computational auditory scene analysis* (CASA) [1], aimed at the separation and classification of sounds. However, video analysis is absolutely one of the most important and emerging technology for security purposes. There are several reasons that concur to make the video information so crucial; obviously a video footage is a rich source of information containing time-dependent visual features thus allowing a deep analysis of objects' motion and behavior. Although CCTV systems are widespread in many urban areas, automatic video surveillance is a relatively new task and recently many effort have been spent in creating "intelligent systems" capable of detect dangerous situations in real scenarios. The reasons for this reside in the complexity of video data which need to be efficiently processed to extract useful information. In particular, a challenging task in video analysis is to limit false positives. Reflexes, shadows, false motion (due, for instance, to camera vibrations), changes in the illumination, are some examples of the causes of false positives and false alarms.

For example, automatic video surveillance systems commonly analyze the scene searching for moving objects. Well known techniques exist to cope with this problem that is commonly referred as "change detection". Every time a difference against a reference model is sensed, it should be analyzed to allow the system to discriminate among a usual situation or a possible threat. When the sensor is a camera, motion is the key element to detect changes and moving objects must be correctly classified according to their nature. In this context we can distinguish among two different kinds of threat that can lead to dangerous situations in a video-surveilled environment. The first one is due to environmental changes such as rain, fog or smoke present in the scene. This kind of phenomena are sensed by the camera as moving pixels and, subsequently as moving objects in the scene. This kind of threats shares

some common characteristics such as texture, shape and color information and can be detected observing the features' evolution in time.

The second situation arises when people are directly responsible of the dangerous situation. In this case a "subject" is acting in an unusual way leading to an abnormal situation. From the sensor's point of view, moving pixels are still observed, but specific features and time-dependent statistical models should be adopted to learn and then correctly detect unusual and dangerous behaviors.

With these premises, this chapter will present two different case studies. The first one describes the detection of environmental changes in the observed scene and details the problem of reliably detecting smoke in outdoor environments using both motion information and global image features, such as color information and texture energy computed by the means of the Wavelet transform. The second refers to the problem of detecting suspicious or abnormal people behaviors by means of people trajectory analysis in a multiple cameras video surveillance scenario. Specifically, a technique to infer and learn the concept of normality is proposed jointly with a suitable statistical tool to model and robustly compare people trajectories.

2 State of the Art

2.1 State of the Art on Video-based Smoke Detection

Smoke detection in video surveillance systems is still an open challenge for computer vision and pattern recognition communities. It concerns the definition of robust approaches to detect, as soon as possible, fast propagation of smoke possibly due to explosions, fires or special environmental conditions. These systems can replace standard smoke and fire sensors, which cannot be applied in large and open spaces. Moreover, detecting smoke by visual cues could allow fast and reactive alarms also in some specific situations, where smoke is growing in unconventional directions, so that the time-to-alarm of normal sensors could become unacceptable.

The video analysis tasks for smoke detection are not trivial due to the variability of shape, motion and texture patterns of smoke, which appearance is dependent on the luminance conditions, the background manifolds and colors of the scene. The smoke identification becomes more challenging in presence of other moving objects and shadows and whenever the background is variable too.

The problem of studying environmental effects on the scene have been deep investigated in literature. In particular several situations arise when weather condition affects the object visibility in the scene leading to particular and dangerous situations. All these cases belong to the macro category of event detection problems that may be studied to trigger alarms or signaling whether a dangerous events happens or not. Among the natural phenomena that may

Event	Technique	Authors
Haze	Statistical model	Oakley et. al. [45]
	Airtight and Attenuation scattering model	Narasimhan and Nayar [46]
	Light polarization	Schechner et. al. [47]
Rain	Setting camera parameters	Garg et. al. [50]
	Intensity constrain, photometric constrain, spatio-temporal correlation	Garg et. al. [53]
Fire	Color information	Phillips et. al. [49]
	FFT and boundary analysis	Fastcom Tech.SA [52]
	FFT and shape analysis	C.B. Liu [51]
Smoke	non-self similarity and motion irregularities	Kopilovic et. al. [34]
	Chromatic analysis, growth-rate and disorder measure	Chen et. al. [36]
	Mean Crossing Rate	Xiong et. al. [38]
	Wavelet transform, energy analysis and shape analysis	Toreyin et. al. [41]

Table 1. Summary of Reference for several natural event detection techniques.

be visually analyzed we can annoverate for examples rain, fog and smoke due to fire or different sources.

Narasimhan and Nayar have extensively studied the visual manifestation of different weather conditions. In [46] they propose an interesting method, based on atmospheric optic, to recover the "clear-day" scene color from two or more images taken under different and unknown weather conditions. They also developed a method for depth-segmentation and extracting three dimensional scene structure. In order to do this they used two scattering model: Attenuation and Airlight (scattering caused by fog or haze), with the constraint that both observer and the object observed must lie at the ground level. Schechner et al. [47] propose a method for haze removal from an image. His method is based on the fact that usually the natural environmental light scattered by the atmosphere is partially polarized. This approach does not rely of previous knowledge about the scattering model or knowledge about the illumination direction, but require only two independent images. Oakley et al. [45], instead use a statistical model for detecting Airlight . Then a linear dependency between the real pixel value and the distorted pixel value is issued for both monochromatic and color images. This approach does not distort images taken in clear weather condition and tested with video sequences, it presents color stability for subsequent frame. Regarding the visual effect due to the rain, Garg et al. [48] studied the appearance of a single raindrop, developing a photometric and geometric model for the raindrop refraction and reflection. He shows that a raindrop behaves like a wide-angle lens and although it is a transparent entity, its brightness is independent from the background brightness, because the drop has a large field of view and the background subtend only a small angle. Subsequently in [50] is proposed how to remove the rain from a video without post-processing and without altering the scene percep-

tion, in fact he derives the relationship between the properties of rain, camera exposure time, depth of field and scene brightness. He shows that the rain visibility increase with the square of the raindrop size, and decrease linearly with the brightness of the background. Finally in [53] authors developed an algorithm capable to detect rain in a video sequence. The detection of the rain is composed by several step: in the first step all the pixel that present a peak in intensity over a set of three subsequent frame are selected; in the second step the false positive are discarded using the photometric constrain; subsequently the spatio-temporal correlation and the direction of the rain fall are computed. Although these method are all of some interest in many security applications the problem of fire and smoke detection is definitely crucial for improving people safety and represents a hard challenge to be solved using cameras sensor.

For this reason we focus our attention on the problem of smoke detection and how to fast and reliably detect dangerous smoke presence in the scene. The problem of smoke detection has been discussed in the past in some works where local features of pixels in the images or measures on the shape temporal variations are exploited. In an early work, Kopilovic et al. [34] took advantage of irregularities in motion due to non-rigidity of smoke. They computed optical flow field using two adjacent images, and then used the entropy of the motion directions distribution as key feature to differentiate smoke motion from non-smoke motion. Similarly, motion was exploited in [35] where local motions from cluster analysis of points in a multidimensional temporal embedding space are extracted. The goal was to track local dynamic envelopes of pixels, and then use features of the velocity distribution histogram to discriminate between smoke and various natural phenomena such as clouds and wind-tossed trees that may cause such envelopes. In this work, the presence of other moving objects, typical of video surveillance scenes, has not taken into account.

Recently, Chen, Yin et al. [36] present a smoke detection approach working on pixel-level classification after motion segmentation based on frame difference. Pixels can be initially classified as a smoke-pixel with a very simple chromaticity-based static decision rule; it is based on two thresholds in the color space assuming that smoke usually displays grayish colors. A Further dynamic decision rule is dependent on the spreading attributes of smoke: the ratio between the sums of circumferences of smoke regions segmented and the number of smoke-pixel extracted can give a measure of disorder in the segmented objects. Similarly other works evaluate the contours of the object that are candidate to be classified as smoke. In [38], smoke detection is based on four steps: background subtraction, flickering extraction, contour initialization, and contour classification using both heuristic and empirical knowledge about smoke. Background subtraction uses the Stauffer and Grimson algorithm [37]. Then a measure of flickering is provided. They state that flickering frequency of turbulent flame has shown experimentally to be around 10Hz and it could be as low as 2 or 3 Hz for slowly-moving smoke. The temporal periodicity can be calculated using Fast Fourier Transform (FFT), Wavelet

Transform or Mean Crossing Rate (MCR). They adopt the Mean Crossing Rate (MCR). Finally as in [36], a measure of the shape complexity given by the ratio between edge length and area is provided to achieve classification. Also in this work, only qualitative measure are provided.

An interesting and robust approach has been defined by Toreyin et. al. [39] and further improved in [40] and [41]. They use the Collins background subtraction method to extract moving objects [42]. Then as in previous work a flickering analysis and a measure of turbulence is provided by evaluating the edge and texture variation using the Wavelet Transform. In each block of the sub-image resulting, after the wavelet decomposition, the variation of energy is computed. The energy is given by the sum of the high-frequency components in the wavelet domain. Finally two thresholds are given to measure an acceptable energy variation. The dynamism of the variation is modeled with a simple three state Random Markov Model (RMM), trained with smoke and non-smoke pixels. Finally, an analysis of smoke shape complexity is provided as in [36] and [38], based on the distance between the contour points and the center of mass of the shape. This approach is quite robust in the given examples, but a precise evaluation of the different features contributions is not provided. Many systems exist for detecting natural event in the scene and could be effectively used for detecting tempestively possible threat. In the following we will present a case study that deeply covers the problem of detecting whether a moving object in the scene is smoke or not using motion clues and texture.

2.2 State of the Art on People Path Analysis

Recent advances in computational resources and algorithms have made distributed video surveillance very appealing to both the academia and the industry. As a consequence, there exist in the literature many works addressing some of or all the steps related to distributed video surveillance: from motion detection and moving object segmentation [2, 3, 4], to object tracking with occlusion handling [5, 6, 7], to fusion among multiple cameras, with either overlapped [8, 3, 9] or disjoint views [10], to higher-level reasoning modules to analyze the behaviors and the interactions, or to detect and classify events [11, 12].

Despite this considerable amount of papers and techniques, the focus of these proposals has been mainly on proposing innovative solutions capable to handle the most complex situation possible, with few (or none) attention to the real-time constraints or to the computational requirements in general. However, even if complexity is a requirement in order to propose significant advances with respect to the state of the art, real-time alarming is often a must in this type of systems, since off-line processing does not guarantee a timely response to relevant events. Obviously, a careful tuning of the trade-off between efficiency and accuracy must be achieved in order to preserve as

much as possible the flexibility and the applicability of the system to different contexts.

Several real-time video surveillance systems have been proposed in the past, but they basically proposed quite complex techniques for fusing single (low-level) algorithms from multiple cameras, such as moving object detection and tracking [13, 14]. Instead, the implementation of higher level tasks (e.g., trajectory/path classification) in real time has not been deeply explored. This section will mainly focus on related works in the field of (real-time) trajectory analysis, which is the main contribution of this chapter.

Trajectory analysis has been studied in depth over the last years, especially for its application in people surveillance. Morris and Trivedi in [15] proposed a recent survey on state-of-art techniques for modeling, comparing and classifying trajectories in video surveillance. The simplest way to define a similarity measure between trajectories is the adoption of Euclidean distance between spatial coordinates as proposed in [16], while the Hausdorff distance was adopted by Junejo *et al.* [17]. However, both these measures only perform point-to-point comparison on trajectories of the same length and, additionally, the Euclidean distance needs that the trajectories have the same length, while Hausdorff distance does not need same length, but cannot distinguish the opposite directions. Chen et al. in [18] presented a method to compare the trajectories after projecting them in a null-space to obtain a representation insensitive to projective transformation of the trajectories themselves.

The distance between correspondent points only can be affected by segmentation errors, noise, temporal shifts, or in general misalignments between trajectories. Thus, many inexact matching techniques have been extensively used both for trajectory analysis [19, 20] and for several different applications ranging from speech [21] to handwriting recognition [22]. Alignment technique like *Longest Common SubSequence* (LCSS) and *Dynamic Time Warping* (DTW) have been efficiently applied to compare trajectory shapes in sign language recognition and surveillance applications [21, 20].

The similarity measures with or without alignment are typically defined in a statistical framework. Mecocci and Panozzo in [23] suitably modified the iterative *Altruistic Vector Quantization* algorithm to robustly cluster trajectories by pure spatial observations obtaining representative prototypes. The anomaly detection is based on fitting a spatial Gaussian on each prototype and statistically checking for fitness of new trajectory samples. In [17], Junejo *et al.* applied graph cuts to cluster trajectories with the Hausdorff distance. In [9] a system for learning statistical motion patterns using a two-stage fuzzy k-means is presented. Porikli [24] proposed the use of a HMM-based similarity measure where each trajectory is modeled with a HMM and compared using the cross likelihood. The results are promising but, in general, a large amount of data is needed to avoid overfitting in the HMM training phase.

Our approach mutuates from the two most common approaches for trajectory comparison and clustering. On the one hand, it adopts an alignment-based distance measure to compare sequences of different lengths and, on the

other hand, it employs a statistical measure to perform point-to-point comparison to deal with inaccuracies of the automatic video surveillance system that extracts the people trajectories. Additionally, a specific on-line distance measure has been developed to obtain a robust and efficient comparison among paths each time a new trajectory point is extracted by the system.

3 Environment Threat Detection: Video-based Smoke Detection

Even though humans can quite easily identifying smoke with a joint use of vision and nose senses, finding smoke in digital video is a challenging problem since smoke shares several features with other objects but also results to be very transparent. This section will present a complete system for detecting smoke in difficult situations by only processing video streams.

3.1 Smoke detection for foreground object classification

The proposed model evaluates the joint contribution coming from the graylevel image energy and color intensity attenuation to classify an object as possible smoke. We assume that when smoke grows and propagates in the scene its image energy is attenuated by the blurring effect of smoke diffusion.
We firstly detect possible candidate objects by means of a motion segmentation algorithm. When a new foreground object is detected we analyze its energy using the Wavelet Transform coefficients and evaluate its temporal evolution. The color properties of the object are analyzed accordingly to a smoke reference color model to detect if color changes in the scene are due to a natural variation or not. The input image is then divided in blocks of fixed sized and each block is evaluated separately. Finally a Bayesian approach detect whether a foreground object is smoke.

Energy analysis using the direct wavelet transform

An efficient way to evaluate the energy variation of an intensity image is the discrete wavelet transform DWT [43].

The DWT is obtained convolving the image signal with several banks of filters obtaining a multiresolution decomposition of the image. Given the input image I_t the decomposition produces four subimages, namely the compressed version of the original image C_t, the horizontal coefficient image H_t, the vertical coefficient image V_t and the diagonal coefficient image D_t. An example decomposition is computed with the algorithm proposed in [43] is shown in Fig. 1

The energy is evaluated blockwise dividing the image in regular blocks of fixed size and summing up the squared contribution coming from each coefficient image:

Fig. 1. Example of discrete wavelet transform. The leftmost image is the original image. The right image is the transformed one. The components are: top left compressed image C_t, top right horizontal coefficient image H_t, bottom left vertical coefficient image V_t and bottom right diagonal coefficient image D_t.

$$E(b_k, I_t) = \sum_{i,j \in b_k} V_t^2(i,j) + H_t^2(i,j) + D_t^2(i,j) \tag{1}$$

where b_k is the k^{th} block in the input image I^t.
The energy value of a specific block varies significantly over time in presence or absence of smoke, Fig. 2.

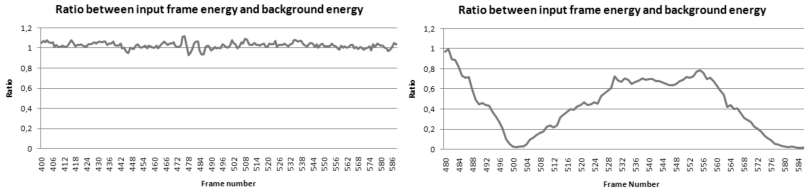

Fig. 2. Left figure: the energy ratio trend of a non smoke block. Right figure: the energy ratio trend of a smoke block. In presence of smoke the energy ratio is subjected to gradual drops in its value.

When the smoke covers part of the scene the edges are smoothed and the energy consequently lowered. This energy drop can be further emphasized computing the ratio $r(B_k)$ between the image energy of the current input frame and the one of the background model. The energy ratio has the advantage of normalizing the energy values and allowing a fair comparison between different scenes where the block energy itself can vary significantly. The ratio of the block b_k is given by:

$$r(b_k, I_t, Bg_t) = \frac{E(b_k, Bg_t)}{E(b_k, I_t)} \tag{2}$$

where Bg_t is the background model up to time t and I_t is the input frame.

The analysis of the energy ratio is performed in two different context to account for both global and local energy drops. Firstly the image energy variation is computed frame by frame to bias the detection using global information. Several clips containing a smoke events have been analyzed and the global energy ratio of the scene computed by sum the block energy. The *Parzen window* technique is adopted to build a non parametric distribution from global energy ratio values computed on several clips. The parzen window method is a kernel density estimator that computes a non parametric distribution from a set of iid samples $X = \{x_i \,|\, i = 1 \dots N\}$ of a random variable x. Adopting a specific kernel distribution the approximated pdf is computed summing the kernel for all the sampled values:

$$\widehat{f} = \frac{1}{N\,h} \sum_{i=1}^{N} K(x - x_i) \tag{3}$$

using a standard Gaussian kernel function $K = \frac{1}{2\pi} e^{-\frac{1}{2}\,x^2}$.

Secondly each block is then locally evaluated to capture the temporal evolution of the energy ratio. When an energy drop is observed for a significant period of time an edge smoothing process occurs. The edge smoothing process can be affected by noise due to light variation in the scene. A Mixture of Gaussian model is adopted to improve the analysis robustness.

The MoG has the great advantage to correctly catch variations for multimodal distributions. To compute the probability for each frame the on-line expectation maximization algorithm proposed in [44] is used. In detail, for all blocks b_k of the image I_t at time t the value $r(b_k, I_t, Bg_t)$ is computed and the MoG of block b_k updated using a selective update method.

This process has a main advantage. The mixture component reweighting process is able to catch slow and gradual variations of energy ratio. Values that do not occur frequently are filter out and assigned to the least probable Gaussian of the mixture. This property is helpful for evaluating the gradient intensity lowering process of smoking regions that has the peculiarity of being slow and continuous in time, Fig. 3.

To capture the time variation of the energy ratio the Gaussian Mixture Model was preferred to a Hidden Markov model (HMM). Although HMMs are widely adopted to classify and model temporal stochastic processes, the data values sequence is crucial to obtain a good classification. Instead, as previously stated, the block energy ratio is subject to strong fluctuations of energy values due to noise and natural scene lighting. This reason makes the lowering sequence unpredictably variable in different setups; thus the specific energy drop trajectory can produce misleading results. On the contrary is interesting to analyze the global trend.

Color analysis to detect blended smoke regions

When a smoke event occurs, scene regions covered by smoke change their color properties. The smoke can either be completely opaque or partially transpar-

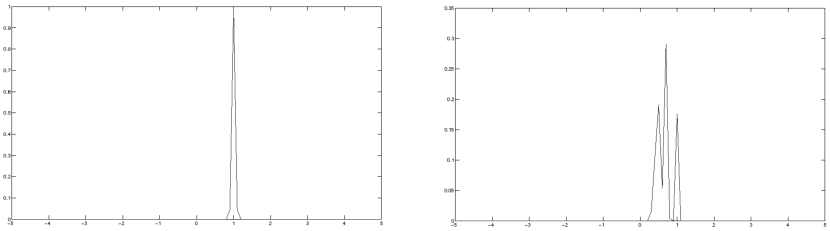

Fig. 3. Gaussian mixtures obtained observing energy ratio values at a single block. The left plot shows the mixture when there is no smoke in the block. The right plot shows how the mixture changes when smoke is in the scene. It is worth noting that when a block is covered by smoke the mixture components mean values move gradually towards 0.

ent. In the former case the covered region changes completely its color while in the latter case the color of the covered region appears to be blended with the smoke color.

This simple observation remains valid in all the observed cases and intuitively suggests a hint to characterize the color of a smoke region. The proposed model simply adopts an evaluation based on a blending function mutuated from computer graphics. A reference color model is chosen in the RGB color space to represent the color of the smoke in the scene. The model is selected by analyzing the different color tones produced combusting different materials. For explanatory purposes is possible to concentrate the analysis to the case of a light gray color model as the smoke in the leftmost image of Fig. 1. Each pixel $I_t(i,j)$ of the input frame at time t is then checked against the smoke model and the background model Bg_t to evaluate the reference color presence computing the blending parameter bl using equation 4. The evaluation takes into account the case where the scene color and the smoke color are mixed together.

$$bl(i,j,I_t,Bg_t,S) = \frac{I_t(i,j) - Bg_t(i,j)}{S - Bg_t(i,j)} \tag{4}$$

where Bg_t is the current background model at time t and S is the smoke reference color model.
To filter out the errors and possible measurements inaccuracy the blending value is computed for each image block as the average of bl values in the block:

$$\beta_{b_k}(I_t,Bg_t,S) = \frac{1}{N^2} \sum_{i,j \in \ ,b_k} \frac{I_t(i,j) - Bg_t(i,j)}{S - Bg_t(i,j)} \tag{5}$$

where block size is $N \times N$
In conclusion the β measure quantifies how much each block globally shares chromatic properties with the reference color model.

A Bayesian approach for classification

Previously the blockwise energy ratio measure r and the color blending measure β have been presented as possible discriminant features to identify a smoke region in the scene. A Bayesian formulation has been chosen to identify whether a block b_k is likely to belong to a smoke region. For each block the posterior probability of smoke presence, the event $f = 1$, considering the block b_k is defined:

$$P(f = 1|b_k) \propto P(b_k|f = 1)P(f = 1) \tag{6}$$

The likelihood value is obtained by combining both the contributions coming from energy ratio and color information. These terms are considered probabilistically independent to simplify the treatment.

$$P(b_k|f = 1) = P(r_{b_k}, \beta_{b_k}|f = 1) = P_r(b_k|f = 1) \cdot P_\beta(b_k|f = 1) \tag{7}$$

The likelihood contribution due to energy ratio decay is obtained by summing the weighted Gaussians of the MOG having mean value below a considered threshold th_1 computed empirically observing the mean energy ratio value in smoke regions.

$$P_r(b_k|f = 1) = \sum_{i=1}^{K} w_i N(r(b_k, I_t, Bg_t)|\mu_i\, \sigma_i) \tag{8}$$

when the i^{th} Gaussian mean value $\mu_i < th_1$.

The color contribution to the likelihood value is directly computed as the block color blending measure β_{b_k} according to equation 5.

$$P_\beta(b_k|f = 1) = B_k(I_t, Bg_t, S) \tag{9}$$

The classification is biased making use of prior knowledge acquired observing several clips containing smoke. The prior probability of a smoke event in the current frame is directly related to the mean energy ratio value of the scene and computed using the non parametric distribution obtained by equation 3.

$$P(f = 1) = \widehat{f}(\frac{1}{M} \sum_{\forall b_k \in I_t} r(b_k, I_t, Bg_t)) \tag{10}$$

where I_t is composed by M blocks.

The posterior probability value is thresholded to identify a candidate smoke block. The test for smoke presence is performed after foreground object segmentation. For any segmented object in the scene the number of candidate blocks intersecting the object's blob is computed. Finally an object is classified as smoke when the 70% of its area overlays candidated smoke blocks.

3.2 Experimental Validation of the Model

The proposed smoke detection system can be used in conjunction with a whichever video surveillance system providing moving object segmentation using a background model. The background model should be updated regularly but smoke regions should not be included in the background. This can be achieved choosing a slow background update rate and avoiding updating the background model areas where a smoke object is detected. The tests were performed using both the Stauffer and Grimson background model with selective update [37] and the SAKBOT median background model with knowledge based update proposed in [32]. Although the results did not vary significantly changing the background model and object detection technique, the second method has been preferred since discriminates the presence of possible shadows objects too.

			Temporal Analysis		**Color Analysis**		**Global analysis**	
Clip	**Frame No**	**Type**	**TtD**	**FP**	**TtD**	**FP**	**TtD**	**FP**
Clip1	165	Outdoor	22	-	1	-	1	-
Clip2	210	Indoor	18	-	1	-	1	-
Clip3	2200	Outdoor	28	-	34	-	20	-
Clip4	3005	Indoor	212	-	273	-	285	-
Clip5	1835	Indoor	87	-	100	3	52	-
Clip6	2345	Outdoor	129	-	161	-	116	-
Clip7	2024	Indoor	57	3	99	-	35	-
Clip8	2151	Outdoor	88	2	88	-	42	-
Clip9	1880	Outdoor	59	-	56	-	45	-
Clip10	2953	Outdoor	457	-	498	-	300	-
Clip11	1485	Indoor	62	-	x	5	62	-
Clip12	499	Outdoor	43	-	8	-	16	-
Clip13	195	Indoor	53	-	23	-	27	-
Clip14	1226	Outdoor	77	-	370	-	69	-
Clip15	109	Outdoor	29	-	x	1	3	-

Table 2. System results on 15 reference clips. The detection rate was evaluated using energy and color component respectively and their joint contribution. Time to detect(TtD) and False positives rate(FP) are reported for each considered contribution.

In all the tests carried out the learning rate α, [37], of the MoGs used to model the energy ratio decay was set to 0.1. Although changing this parameter does not have major effects on the system performance, a fast learning rate is preferable to detect energy ratio variations rapidly. The system was tested on 50 clips of varying length in both indoor and outdoor setups where moving objects such as people or vehicles were present in the scene during the smoke event. Each clip contained a smoke event. Part of the dataset is

publicly available at website *http://imagelab.ing.unimore.it/visor*. Each likelihood term was evaluated separately to measure the impact on the system performance.

The table Tab. 2 summarizes the results obtained on 15 reference clips. The first column of the table reports the video type and its frame-length. The average clips framerate is 25fps. The remaining columns report the results obtained using each likelihood term separately and finally the results of the whole system. The detection time after the smoke event occurs is reported for all the test clips. The table clearly shows that the likelihood term due to temporal analysis (eq.8) is effective in most of the observed cases. The main problem is the long detection time. This is caused by the time based statistics used to capture the energy ratio decay. Although the likelihood contribution due to color blending has the advantage of speed up the detection process it tends to detect much false positives if used alone. See seventh column of Tab. 2. Observing the last two columns of Tab. 2 we can state that the complete approach is fast and reliable enough even in situations where each likelihood contribution fails. The overall system results on the 50 clips used for testing purposes report a detection rate of 77% 3 seconds later the smoke event occurs, 98.5% 6 seconds later and finally 100% 10 seconds later with an average true positive rate of 4%. Fig. 4 shows some snapshots of the system working on different conditions.

In conclusion the proposed case study underlines another important application of pattern recognition in video surveillance contexts for a deep understanding of the monitored scene and event detection.

4 People Threat Detection: Abnormal Path Detection

Recent advances in computational resources and algorithms have made distributed video surveillance very appealing to both the academia and the industry. As a consequence, there exist in the literature many works addressing some of or all the steps related to distributed video surveillance: from motion detection and moving object segmentation [2, 3, 4], to object tracking with occlusion handling [5, 6, 7], to fusion among multiple cameras, with either overlapped [8, 3, 9] or disjoint views [10], to higher-level reasoning modules to analyze the behaviors and the interactions, or to detect and classify events [11, 12].

This section will describe a complete and full-working system for real-time detection of abnormal paths in real multi-camera scenarios. In fact, in most of the surveillance scenarios “abnormal” is often synonymous of “dangerous”. The first step consists in the extraction of the points composing the trajectory/path and it requires to segment moving objects in all the cameras and to track them both in each single camera and across adjacent cameras.

Moving object detection and tracking from a single static camera is a well-known and almost-solved problem. Our system makes use of the approach

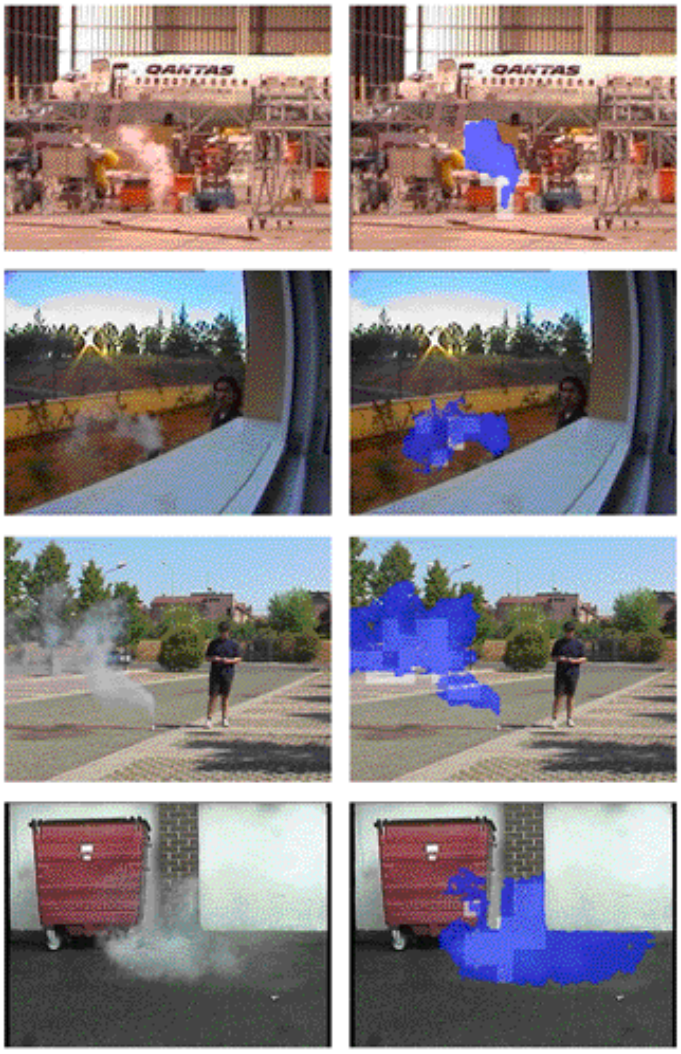

Fig. 4. Snapshots of the proposed system working on several clips in different conditions. The blue area in the images is detected as smoke.

proposed by us in [3]. The approach is based on background suppression using a suitable modification of the median background model that improves both the initialization and the update phases using a knowledge based reasoning scheme. Proper techniques for background bootstrapping, ghost suppression, and object validation have also been introduced to improve the accuracy of the segmentation in cluttered outdoor setup. The objects detected as moving are then tracked in each single view by means of an appearance-based algorithm

proposed in [5]. The algorithm uses a classical predict-update approach, but it takes into account not only the status vector containing position and speed, but also the memory appearance model and the probabilistic mask of the shape.

Once each camera has processed the video stream and obtained the object tracks, there is the need to render the track labels/ids consistent among the different cameras: this step is crucial to keep track of the object when it moves across the fields of view of the different cameras and thus to obtain longer and more stable trajectories. This problem is also known as *consistent labeling* and has been fully studied in the literature [3]. We borrowed the approach, proposed in [3], valid for cameras with partially-overlapped fields of view (FoVs), which adopts a geometric approach that exploits cameras' FoV relationships and constraints to impose identities consistency. In detail, when cameras partially overlaps the shared portion of the scene is analyzed and people identities are matched geometrically, by exploiting ground-plane homographies and pairwise epipolar geometry.

Once that the people trajectories (on the ground plane) are obtained, we can develop a method for comparing trajectories analyzing different characteristics: trajectories shape and trajectories positions in a given scene. The shape analysis is important when unfrequent or particular behaviors must be extracted without the knowledge of where and when the event of interest occurs. Conversely, positional analysis is useful when a specified portion of the scene should be analyzed and scene properties, such as entry or exit zones, can be deduced directly from people activities.

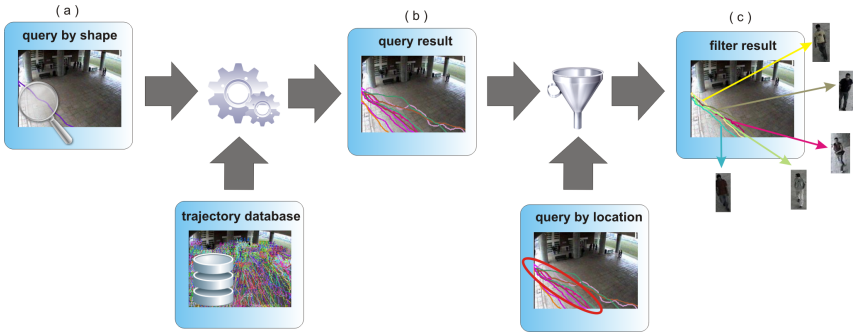

Fig. 5. Example of a possible application scenario of the trajectory analysis framework.

A typical reference application is shown in Fig. 5. Here it is depicted how the system could be useful for instance in forensic application. First, a query is performed on the trajectory shape, Fig. 5.a; second, several exemplars having the desired shape are shown to the user. It is then possible to choose a specific example, according to its position in the scene, and the system will retrieve all

the trajectories similar to the desired one (Fig. 5.c). Finally, it is possible to retrieve people snapshots and trajectories information (such as creation time) that could be of interest during the investigation process.

4.1 Trajectory Model

As stated in the previous Section, people trajectories can be modelled by means of either the sequence of spatial locations or the sequence of directions.

Spatial Model for Positional Analysis

The people trajectory projected on the ground plane is a very compact representation based on a sequence of 2D data ($\{(x_1, y_1), \cdots, (x_n, y_n)\}$ coordinates), often associated with the motion status, e.g. the punctual velocity or acceleration.

When large data are acquired in a real system they should be properly modeled to account for tracking errors, noise in the support point extraction and inaccuracies due to the multi-camera data fusion module. Positional trajectories must then be correctly extracted by the tracking system and analyzed in order to discriminate or aggregate different kinds of people behaviors.

When observing a video surveillance scenario some paths are considerably more common than others, and this can be very meaningful in suurveillance applications. Different path frequencies are mainly due to two factors. First, the structure of the environment may condition significantly the way people move. Second, according to the scenario, people tend to reproduce frequent behaviors.

Given the k^{th} rectified trajectory projected on the ground plane $T_k = \{\mathbf{t_{1,k}} \dots \mathbf{t_{n_k,k}}\}$, where $\mathbf{t_{i,k}} = (x_{i,k}, y_{i,k})$ with n_k the number of points of trajectory T_k, a bi-variate Gaussian centered on each data point $\mathbf{t_{i,k}}$ (i.e., having the mean equal to the point coordinates $\boldsymbol{\mu_{i,k}} = (x_{i,k}, y_{i,k})$) and with fixed covariance matrix $\boldsymbol{\Sigma}$ can be defined as:

$$\mathcal{N}_{i,k} = \mathcal{N}(x, y \mid \boldsymbol{\mu_{i,k}}, \boldsymbol{\Sigma}) \tag{11}$$

An example of the fitting of Gaussians onto the trajectory points is shown in Fig. 6, where (a) shows an exemplar trajectory, (b) the 3D plot of the superimposed Gaussians and the x-y projection.

The main motivation for this modeling choice relies in the fact that when comparing two points belonging to different trajectories small spatial shifts may occur and trajectories never exactly overlap point-to-point. Using a sequence of Gaussians, one for each point, allows to build an envelope around the trajectory itself, obtaining a slight invariance against spatial shifts.

After assigning a Gaussian to each trajectory point, the trajectory can be modeled as a sequence of symbols corresponding to Gaussian distributions $\overline{T}_j = \left\{S_{1,j}, S_{2,j}, ..., S_{n_j,j}\right\}$, where each symbol $S_{i,j}$ is modeled as in equation 11.

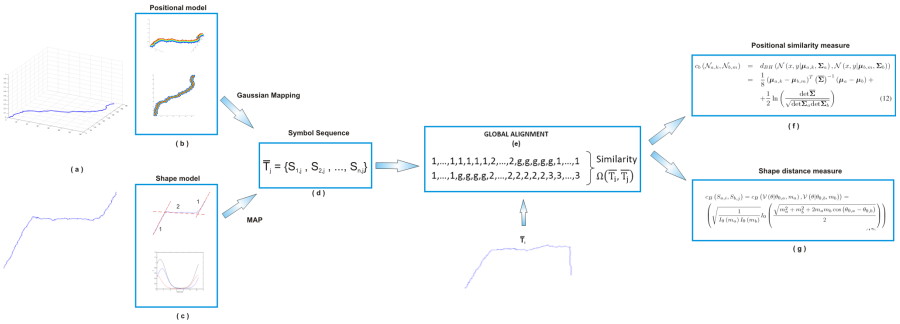

Fig. 6. Example of the trajectory model.

Angular Model for Shape Analysis

Using a constant frame rate, the sequence of (x, y) coordinates can be easily converted in directions/angles, in order to model the single trajectory T_j as a sequence of n_j directions θ, defined in $[0, 2\pi)$:

$$T_j = \left\{\theta_{1,j}, \theta_{2,j}, \ldots, \theta_{n_j,j}\right\} \tag{12}$$

In order to analyze its shape, *circular* or *directional statistics* [25] is a useful framework for the analysis. We propose to adopt the von Mises distribution, that is a special case of the von Mises-Fisher distribution [26, 27]. The von Mises distribution is also known as the *circular normal* or the *circular Gaussian*, and it is particularly useful for statistical inference of angular data. When the variable is univariate, the probability density function (pdf) results to be:

$$\mathcal{V}\left(\theta|\theta_0, m\right) = \frac{1}{2\pi I_0\left(m\right)} e^{m\cos(\theta-\theta_0)} \tag{13}$$

where I_0 is the modified zero-order Bessel function of the first kind, defined as:

$$I_0\left(m\right) = \frac{1}{2\pi}\int_0^{2\pi} e^{m\cos\theta} d\theta \tag{14}$$

and represents the normalization factor. The distribution is periodic so that $p\left(\theta + M2\pi\right) = p\left(\theta\right)$ for all θ and any integer M.

Von Mises distribution is thus an ideal pdf to describe a trajectory T_j by means of its angles. However, in the general case a trajectory is not composed only of a single main direction; having several main directions, it should be represented by a multi-modal pdf, and thus we propose the use of a mixture of von Mises (MovM) distributions:

$$p\left(\theta\right) = \sum_{k=1}^{K} \pi_k \mathcal{V}\left(\theta|\theta_{0,k}, m_k\right) \tag{15}$$

As it is well known, EM algorithm is a very powerful tool for finding maximum likelihood estimates of the mixture parameters, since the mixture model depends on unobserved latent variables (defining the "responsibilities" of a given sample with respect to a given component of the mixture). The EM algorithm allows the computation of the parameters for the K components of the MovM. A full derivation of this process can be found in [28].

Each direction $\theta_{i,j}$ is encoded with a symbol $S_{i,j}$ with a MAP approach, that, assuming uniform priors, can be written as:

$$S_{i,j} = \underset{r=1,...,K}{\arg\max}\, p\left(\theta_{0,r}, m_r|\theta_{i,j}\right) = \underset{r=1,...,K}{\arg\max}\, p\left(\theta_{i,j}|\theta_{0,r}, m_r\right) \tag{16}$$

where $\theta_{0,r}$ and m_r are the parameters of the r^{th} components of the MovM. With this MAP approach each trajectory T_j in the training set is encoded with a sequence of symbols $\overline{T}_j = \left\{S_{1,j}, S_{2,j}, ..., S_{n_j,j}\right\}$.

4.2 Sequence similarity measure

In order to cluster or classify similar trajectories, a similarity measure $\Omega\left(\overline{T}_i, \overline{T}_j\right)$ is needed. Due to acquisition noise, uncertainty and spatial/temporal shifts, exact matching between trajectories is unsuitable for computing similarity. Thus, two sequences of symbols can be compared by using an inexact matching technique. The main motivation resides in the fact that trajectories are never equal both in number and position of points. Small changes can occur between two similar sequences: for example, there may be some time stretches that result in sequences having different lengths; additionally, sequences may be piecewise-similar, sharing some common parts, but they can be different in other parts. In choosing the similarity measure it is desirable to gain control on the amount of common points that two sequences must share in order to be considered "similar".

For these motivations, the best way to compare two sequences is to identify the best alignment of the sequence data, based on a given point-to-point distance metrics. Point-to-point comparison can be made either directly on the data or by selecting a data representation which assigns a symbol (with a given "meaning") to each data and performing a symbol-to-symbol comparison. However, the trivial model that simply performs a point-wise comparison in the rectified Euclidean plane will result extremely imprecise. We decided to adopt a model that employs statistics to model data points sequences, being consequently robust against measurement errors and data uncertainties, but imposing some constraint and limitation to achieve real-time performance. As stated in the introduction, this permits to achieve a good trade-off between efficiency and accuracy.

Once a sequence of data/symbols is achieved, we can borrow from bioinformatics the method for comparing DNA sequences in order to find the best inexact matching between them, also accounting for gaps. Then, we propose to adopt the *global alignment*, specifically the well-known Needleman-Wunsch

algorithm [29] for comparing sequences of probability distributions. A global alignment (over the entire sequence) is preferable over a local one, because preserves both global and local shape characteristics. Global alignment of two sequences $\overline{T}_i$ and $\overline{T}_j$ is obtained by first inserting spaces, either into or at the ends of the sequences so that the length of the sequences will be the same; by doing this, every symbol (or space) in one of the sequences is matched to a unique symbol (or space) in the other.

The algorithm is based on the concept of "modification" to the sequence (analogous to the mutation in a DNA sequence). The modifications to a sequence can be due to *indel* operations (insertion or deletion of a symbol) or to *substitutions*. By assigning different weights/costs to these operations it is possible to measure the degree of similarity of the two sequences. Unfortunately, this algorithm can be very onerous in terms of computational complexity if the sequences are long. For this reason, *dynamic programming* is used to reduce computational time to $O(n_i \cdot n_j)$, where n_i and n_j are the lengths of the two sequences. Dynamic programming overcomes the problem of the recursive solution to global alignment by not comparing the same subsequences for more than one time, and by exploiting tabular representation to efficiently compute the final similarity score. Each element $V(a,b)$ of the table contains the alignment score of the symbol $S_{a,i}$ of sequence $\overline{T}_i$ with the symbol $S_{b,j}$ of sequence $\overline{T}_j$. This inexact matching is very useful for symbolic string recognition and theoretically could be used on whichever data have been organized in a sequence. However, we do not adopt it directly on the data since they can be affected by measurement noise, but on the pdf corresponding to trajectory data. Thus, the one-to-one score between symbols can be measured statistically as a function of the distance between the corresponding distributions. If the two distributions result sufficiently similar, the score should be high and positive, while if they differ significantly, the score (penalty) should be negative.

The alignment is simply achieved by arranging the two sequences in a table, the first sequence row-wise and the second column-wise, starting from the base conditions:

$$\begin{aligned} V(a,0) &= \Omega(S_{a,i}, -) \\ V(0,b) &= \Omega(-, S_{b,j}) \end{aligned} \tag{17}$$

where Ω represents a suitable similarity measures and $-$ indicates a zero-element or gap.

This is due to the fact that the only way to align the first k elements of the sequence $\overline{T}_i$ with zero elements of the sequence $\overline{T}_j$ (or viceversa) is to align each of the elements with a space in the sequence $\overline{T}_i$.

Starting from these base conditions, the alignment is performed exploiting the recurrent equation of global alignment that computes the best alignment score for each subsequence of symbols:

$$V(a,b) = \max \begin{cases} V(a-1,b-1) + \Omega\left(S_{a,i}, S_{b,j}\right) \\ V(a-1,b) + \Omega\left(S_{a,i}, -\right) \\ V(a,b-1) + \Omega\left(-, S_{b,j}\right) \end{cases} \tag{18}$$

with $1 \leq a \leq n$ and $1 \leq b \leq m$ and where $V(a,b)$ is the score of the alignment between the subsequence of $\overline{T}_i$ up to the a^{th} symbol and the subsequence of $\overline{T}_j$ up to the b^{th} symbol.

Assuming that two distributions are sufficiently similar if the coefficient is above 0.5 and that the score for perfect match is +2, whereas the score (penalty) for the perfect mismatch is -1 (that are the typical values used in DNA sequence alignments), we can write the general score of alignment between two symbols/distributions as follows:

$$\Omega\left(S_i, T_j\right) = \begin{cases} 2 \cdot (c_B) & \text{if } c_B \geq 0.5 \\ 2 \cdot (c_B - 0.5) & \text{if } c_B < 0.5 \\ 0 & \text{if } S_i \text{ or } T_j \text{ are gaps} \end{cases} \tag{19}$$

where c_B represents the cost of aligning two symbols. The following Section will report the proposed way for computing this cost in the two cases of spatial and angular data.

4.3 Statistics Symbol-to-Symbol Distance Metrics

Distance in the case of Spatial Model

In the case of symbol sequences that represent spatial-Gaussian probability distributions, a proper symbol-to-symbol similarity measure must be defined in order to perform the global alignment. Among the possible metrics to compare probability distributions we chose to employ the Bhattacharyya coefficient as in the case of shape model, to measure the distance between the two normal distributions $\mathcal{N}_{a,k}$ and $\mathcal{N}_{b,m}$ corresponding to a^{th} and b^{th} symbols of sequences $\overline{T}_k$ and $\overline{T}_m$, respectively:

$$\begin{aligned} c_b\left(\mathcal{N}_{a,k}, \mathcal{N}_{b,m}\right) &= d_{BH}\left(\mathcal{N}\left(x, y | \boldsymbol{\mu}_{a,k}, \boldsymbol{\Sigma}_a\right), \mathcal{N}\left(x, y | \boldsymbol{\mu}_{b,m}, \boldsymbol{\Sigma}_b\right)\right) \\ &= \frac{1}{8}\left(\boldsymbol{\mu}_{a,k} - \boldsymbol{\mu}_{b,m}\right)^T \left(\overline{\boldsymbol{\Sigma}}\right)^{-1} \left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right) + \\ &\quad + \frac{1}{2} \ln\left(\frac{\det \overline{\boldsymbol{\Sigma}}}{\sqrt{\det \boldsymbol{\Sigma}_a \det \boldsymbol{\Sigma}_b}}\right) \end{aligned} \tag{20}$$

where $2 \cdot \overline{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_a + \boldsymbol{\Sigma}_b$. Since in our case $\boldsymbol{\Sigma}_a = \boldsymbol{\Sigma}_b = \boldsymbol{\Sigma}$, we can rewrite the distance as:

$$c_b\left(\mathcal{N}_a^k, \mathcal{N}_b^m\right) = \frac{1}{8}\left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right)^T \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right) \tag{21}$$

As previously performed for the angular model this coefficient can be injected in equation 19 to obtain the symbol to symbol similarity measure used in the alignment process.

Distance in the case of Angular Model

When the data sequences is modeled using the Mixture of Von Mises Model, Section 4.1, one possible symbol-to-symbol distance between the univariate pdf associated to each symbol, following the scheme of Fig. 6, is the Bhattacharyya coefficient between Von Mises distribution,[30]. We can derive the *Omega* score for the Mixture of Von Mises Model; specifically, we measured the distance between distributions p and q using the Bhattacharyya coefficient:

$$c_B(p,q) = \int_{-\infty}^{+\infty} \sqrt{p(\theta)\, q(\theta)}\, d\theta \tag{22}$$

following the derivation in [28] for two univariate Von Mises distribution the analytic form of the coefficient results:

$$\begin{aligned} &c_B(S_i, T_j) = c_B\left(\mathcal{V}(\theta|\theta_{0,i}, m_i), \mathcal{V}(\theta|\theta_{0,j}, m_j)\right) = \\ &\left(\sqrt{\frac{1}{I_0(m_a)\, I_0(m_b)}}\, I_0\left(\frac{\sqrt{m_i^2 + m_j^2 + 2m_i m_j \cos(\theta_{0,i} - \theta_{0,j})}}{2}\right)\right) \end{aligned} \tag{23}$$

where it holds that $0 \leq c_B(S_i, T_j) \leq 1$.

4.4 Experimental results and discussion

Once a proper similarity measure is available, sequences can be compared according to either their position (Section 4.3) or their shape (Section 4.3). In particular, it could be of interest to retrieve all the sequences similar to a given exemplar (*query problem*) or the most or least frequent sequence sharing shape or position characteristics (*clustering problem*). In forensic and video surveillance applications, this could be of undoubtful utility; sequences can be retrieved according to their shape and then filtered according to their position or vice-versa. Most common paths can also be extracted to synthesize a clear picture of normal and frequent (abnormal and unfrequent) behaviors in a specific scenario. To group together paths sharing some common characteristics we choose to adopt the k-medoids [31] clustering algorithm using the similarity measures introduced in section 4.2. Hereinafter we use interchangeably the trajectory and its symbolic representation in the Ω measure to keep the notation light, but the similarity measure is obviously computed, as previously stated, on the symbolic representation of the trajectory and consequently on the chosen probability density funtction.

The adopted clustering algorithms, K-medoids, is a suitable modification of the well-known k-means algorithm which has the appreciable characteristic to compute, as prototype of the cluster, the element that minimizes the sum of intra-class distances. In other words, let us suppose to have a training set $TS =$

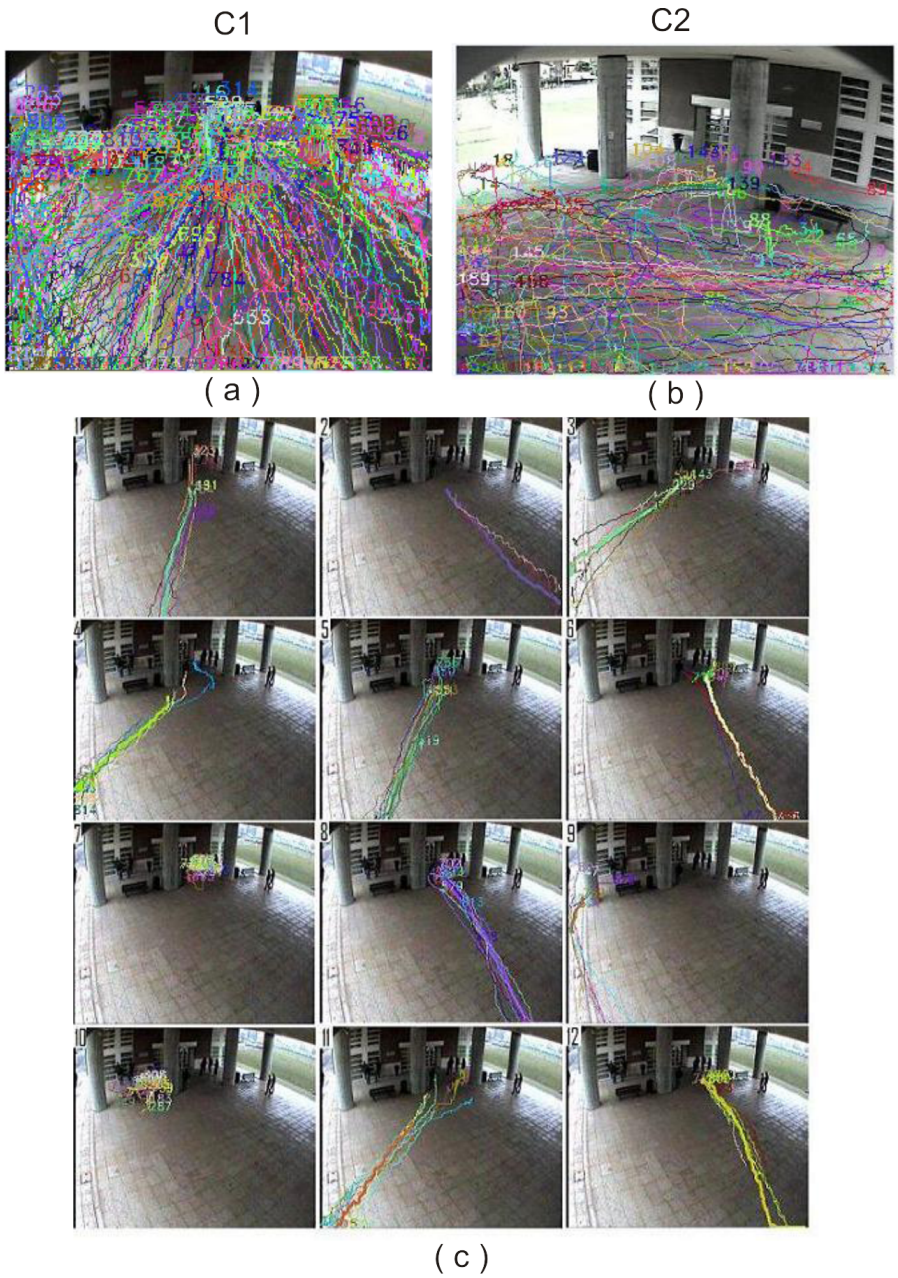

Fig. 7. In (a) and (b) is shown the training set used during the learning stage. (c) shows the obtained most frequent behaviors projected on the $C1$ view.

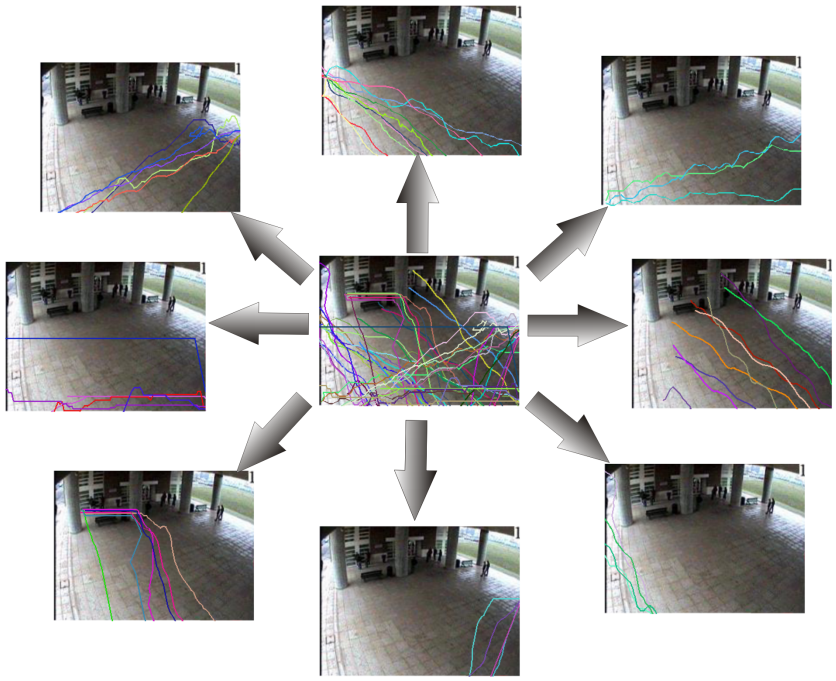

Fig. 8. The center figure shows the training set used for trajectories shape clustering. The remaining figure shows the most frequent behaviors according to their shape.

$\{T_1, \cdots, T_N\}$ composed of N trajectories and set $i = 0$ and $k(0) = N$. As initialization, each trajectory is chosen as prototype (medoid) of the corresponding cluster. The k-medoids algorithm iteratively assigns each trajectory T_j to the cluster $C_{\widetilde{m}}$ at the minimum distance d, i.e. given $k(i)$ clusters $C_1, \cdots, C_{k(i)}$ and the corresponding medoids $M_1, \cdots, M_{k(i)}$, $\widetilde{m} = \underset{m=1,\cdots,k(i)}{\arg\min}\, d\left(T_j, T_{M_m}\right)$, where T_{M_m} is the trajectory corresponding to the medoid M_m. Once all the trajectories have been assigned to the correct cluster, the new medoid M_s for each cluster C_s is computed as that one which minimizes the intra-cluster distances, i.e. $T_{M_s} \equiv T_{\widetilde{p}} = \underset{\forall T_p \in C_s}{\arg\min} \sum_{\forall T_r \in C_s} d\left(T_p, T_r\right) = \underset{\forall T_p \in C_s}{\arg\max} \sum_{\forall T_r \in C_s} \Omega\left(T_p, T_r\right)$. However, one of the limitations of k-medoids (as well as k-means) clustering is the choice of k. For this reason, we propose to use an *iterative k-medoids* algorithm. Then, the following steps are performed:

- Step 1: Run k-medoids algorithm with $k(i)$ clusters
- Step 2: If there are two medoids with a similarity greater than a threshold Th, merge them and set $k(i+1) = k(i) - 1$. Increment i and go back to

step 1. If all the medoids have a two-by-two similarity lower than Th, stop the algorithm

In other words, the algorithm iteratively merges similar clusters until convergence. In this way, the "optimal" number $\widetilde{k}$ of medoids is obtained.

Performing the clustering on a given corpus of trajectories leads to two main advantages. First, after the clustering, clusters cardinality naturally represents by definition how often a specific path occurs, thus allowing to classify the paths in abnormal (unfrequent) and normal (frequent). Second, when the dataset grows dramatically in number of exemplars, the one-to-many approach that consists of comparing a query trajectory with all the trajectories previously stored, can be extremely onerous in term of computational time. Adversely, the adoption of clustering allows the classes to be represented by their prototype, reducing the number of comparisons in the case of query.

To keep this approach consistent when new data are presented to the system, the clusters must be updated every time a new sequence is classified. More operatively, we can define the maximum similarity between the new trajectory T_{new} and the set of clusters $\mathbf{C}$ as $\Omega_{max} = \Omega\left(C_{\widetilde{j}}, T_{new}\right)$, where:

$$\widetilde{j} = \underset{j=1,\ldots,\widetilde{k}}{\arg\max}\ \Omega\left(C_j, T_{new}\right) \tag{24}$$

If this value is below a given threshold Th_{sim} a new cluster $C_{\widetilde{k}+1}$ should be created with T_{new}. The cardinality $\mathcal{C}$ of each class (which represents the prior for a classification normal/abnormal) is updated to take into account the increased number of samples assigned to the cluster:

$$\begin{aligned}
&C_{\widetilde{k}+1} = T_{new}\ ;\ \mathcal{C}\left(C_{\widetilde{k}+1}\right) = \frac{1}{N+1} \\
&\forall i = 1, \ldots, \widetilde{k} \Rightarrow \mathcal{C}_{new}\left(C_i\right) = \mathcal{C}_{old}\left(C_i\right)\frac{N}{N+1} \\
&k = k+1\ ;\ N = N+1
\end{aligned}$$

where N is the current number of observed trajectories.

Conversely, if the new trajectory is similar enough to one of the current medoids, the trajectory is assigned to the corresponding cluster C_j:

$$\begin{aligned}
&T_{new} \in C_j\ ;\ \mathcal{C}_{new}\left(C_{\widetilde{k}}\right) = \frac{\mathcal{C}_{old}\left(C_{\widetilde{k}}\right)\cdot N + 1}{N+1} \\
&\forall i = 1, \ldots, \widetilde{k}, i \neq j \Rightarrow \mathcal{C}_{new}\left(C_i\right) = \mathcal{C}_{old}\left(C_i\right)\frac{N}{N+1} \\
&N = N+1
\end{aligned}$$

Moreover, if the average similarity of the new trajectory with respect to other medoids is smaller than the average similarity of the current medoid C_j, T_{new} is a better medoid than C_j since it increases the separability with

other clusters. Consequently, T_{new} becomes the new medoid of the cluster. Finally, to avoid old and rare trajectories afffecting our model, clusters witth small cardinality and with no new trajectories assigned for a fixed-length time window are dropped.

We tested our system in a two cameras setup at our campus. People are extracted and tracked across camera streams using the multi-camera tracking system described in [32, 33]. Once the trajectories are reliably obtained, we first performed the clustering described above on a dataset of 900 trajectories acquired during an ordinary working day. In this way the most frequent behaviors in the chosen scenario, as shown in Fig. 7, can be extracted according to their position. In this case trajectories sharing similar shape and location are clustered together and it is possible to easily detect the most frequent activity zones of the scene, for example benches where people use to stop. In Fig. 8 trajectories are clustered according to their shape only. In this case it is possible to extract similar trajectories, and most frequent ones as shown in the figure, that share common directions and shape properties independently on where they occur in the scene.

5 Conclusions

This chapter tackles the problem of moving pixel detection from an original perspective: two quite different applications, namely vision-based smoke detection and abnormal path detection, are treated as methods for threat detection (due to the environment and the people behavior, respectively) based on similar concepts. In fact, both these applications start from the detection and analysis of moving pixels and require to effectively distinguish from the two cases. Smoke, in fact, shares some common characteristics such as texture, shape and color information with moving pixels due to objects.

The techniques reported in this chapter demonstrated to be very robust in both cases and can be easily included in advance video surveillance systems to provide both these functionalities.

References

1. Bregman A (1990), Auditory Scene Analysis: The Perceptual Organization of Sound. MIT Press, London
2. Haritaoglu I, Harwood D, Davis L (2000) W4: real-time surveillance of people and their activities. IEEE Trans Pattern Anal Mach Intell 22(8):809–830
3. Calderara S, Prati A, Cucchiara R (2008) Hecol: Homography and epipolar-based consistent labeling for outdoor park surveillance. Comput Vis Image Underst 111(1):21–42
4. Hu W, Tan T, Wang L, Maybank S (2004) A survey on visual surveillance of object motion and behaviors. IEEE Trans Syst Man Cybern C 34(3):334–352

5. Vezzani R, Cucchiara R (2008) Ad-hoc: Appearance driven human tracking with occlusion handling. In: Proc of First Int Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS'2008), in conjunction with BMVC 2008
6. Yilmaz A, Javed O, Shah M (2006) Object tracking: A survey. ACM Comput Surv 38(4):13
7. Zhang Z, Piccardi M (2007) A review of tracking methods under occlusions. In: Proc of Int Conf on Mach Vis Appl 146–149
8. Khan S, Shah M (2003) Consistent labeling of tracked objects in multiple cameras with overlapping fields of view. IEEE Trans Pattern Anal Mach Intell 25(10):1355–1360
9. Hu W, Xiao X, Fu Z, Xie D, Tan T, Maybank S (2006) A system for learning statistical motion patterns. IEEE Trans Pattern Anal Mach Intell 28(9):1450–1464
10. Madden C, Cheng ED, Piccardi M (2007) Tracking people across disjoint camera views by an illumination-tolerant appearance representation. Mach Vis Appl 18(3-4):233–247
11. Parameswaran V, Chellappa R (2006) View invariance for human action recognition. Int J Comp Vis 66(1):83–101
12. Lavee G, Khan L, Thuraisingham BM (2007) A framework for a video analysis tool for suspicious event detection. Multimedia Tools Appl 35(1):109–123
13. Javed O, Rasheed Z, Alatas O, Shah M (2003) Knight: a real time surveillance system for multiple and non-overlapping cameras. In: Proc of Int Conf Multimedia and Expo 1:649–652
14. Zhao T, Aggarwal M, Kumar R, Sawhney H (2005) Real-time wide area multi-camera stereo tracking. In: Proc of Int Conf Comp Vis Pattern Recognit 1:976–983
15. Morris B, Trivedi M (2008) A survey of vision-based trajectory learning and analysis for surveillance. IEEE Trans Circuits Syst Video Technol 18(8):1114–1127
16. Ding H, Trajcevski G, Scheuermann P, Wang X, Keogh EJ (2008) Querying and mining of time series data: experimental comparison of representations and distance measures. Proc VLDB Endow 1(2):1542–1552
17. Junejo I, Javed O, Shah M (2004) Multi feature path modeling for video surveillance. In: Proc. of Int. Conf. Pattern Recognit 2:716–719
18. Chen X, Schonfeld D, Khokhar A (2008) Robust null space representation and sampling for view invariant motion trajectory analysis. In: Proc of IEEE Int Conf Comp Vis Pattern Recognit
19. Piciarelli C, Foresti G (2006) On-line trajectory clustering for anomalous events detection. Pattern Recognit Lett 27(15):1835–1842
20. Buzan D, Sclaroff S, Kollios G (2004) Extraction and clustering of motion trajectories in video. In: Proc of Int Conf Pattern Recognit 2
21. Keogh EJ, Pazzani MJ (2000) Scaling up dynamic time warping for datamining application. In: Proc of ACM SIGKDD Int Conf Knowl Discov Data Min 285–289
22. Qiao Y, Yasuhara M (2006) Affine invariant dynamic time warping and its application to online rotated handwriting recognition. In: Proc of Int Conf Pattern Recognit 2:905–908

23. Mecocci A, Pannozzo M (2005) A completely autonomous system that learns anomalous movements in advanced videosurveillance applications. In: Proc of IEEE Int Conf Image Process 2:586–589
24. Porikli F, Haga T (2004) Event detection by eigenvector decomposition using object and frame features. In: Proc of Comp Vis Pattern Recognit Workshop 7:114–121
25. Mardia K, Jupp P (2000) Directional Statistics Wiley.
26. Fisher R (1953) Dispersion on a sphere. Proc Roy Soc London Ser A 217:295–305
27. Bishop C (2006) Pattern Recognit Mach Learn Springer–Verlag
28. Prati A, Calderara S, Cucchiara R (2008) Using circular statistics for trajectory shape analysis. In: Proc of Comp Vis Pattern Recognit
29. Needleman S, Wunsch C (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol 48(3):443–453
30. Kailath T (1967) The divergence and Bhattacharyya distance measures in signal selection. IEEE Trans Commun Tech COM-15(1):52–60
31. Reynolds A, Richards G, Rayward-Smith V (2004) The Application of K-Medoids and PAM to the Clustering of Rules. Springer Berlin / Heidelberg, 3177:173–178
32. Cucchiara R, Grana C, Piccardi M, Prati A (2003) Detecting moving objects, ghosts and shadows in video streams. IEEE Trans Pattern Anal Mach Intell 25(10):1337–1342
33. Calderara S, Cucchiara R, Prati A (2008) Bayesian-competitive Consistent Labeling for People Surveillance. IEEE Trans Pattern Anal Mach Intell 30(2):354–360
34. Kopilovic I, Vagvolgyi B, Sziranyi T (2000) Application of panoramic annular lens for motion analysis tasks: surveillance and smoke detection. In: Proc of 15th Int Conf Pattern Recognit 4:714–717
35. Vicente J, Guillemant P (2002) An image processing technique for automatically detecting forest fire. Int J Therm Sci 41(12):1113–1120
36. Chen T-H, Yin Y-H, Huang S-F, Ye Y-T (2006) The Smoke Detection for Early Fire-Alarming System Base on Video Processing. In: Proc of Int Conf Intell Inf Hiding and Multimedia 427–430
37. Stauffer C, Grimson WEL (1999) Adaptive Background Mixture Models for Real-Time Tracking. In: Proc of IEEE Conf Comp Vis Pattern Recognit 246–252
38. Xiong Z, Caballero R, Wang H, Finn A, Lelic MA, Peng P (2007) Video-based Smoke Detection: Possibilities, Techniques, and Challenges Suppression and Detection Research and Applications. In: A Techn Working Conf (SUPDET)
39. Toreyin BU, Dedeoglu Y, Cetin AE (2005) Flame detection in video using hidden Markov models. In: Proc of IEEE Int Conf Image Proc
40. Toreyin BU, Dedeoglu Y, Cetin AE (2005) Wavelet based real-time smoke detection in video. In: EUSIPCO
41. Toreyin BU, Dedeoglu Y, Cetin AE, Fazekas D, Chetverikov, Amiaz T, Kiryati N (2007) Dynamic texture detection, segmentation and analysis. In: Proc of ACM Conf Image Video Retr 131–134
42. Collins RT, Lipton AJ, Kanade T (1999) A system for video surveillance and monitoring. In: Proc of 8th Int Top Meet on Robot and Remote Syst
43. Mallat SG (1989) A theory for multiresolution signal decomposition: The wavelet representation. IEEE Trans Pattern Anal Mach Intell 11(7):674–693

44. Sato M. Fast learning of on-line EM algorithm. In: Technical Report TR-H-281, ATR Human Information Processing Research Laboratories.
45. Oakley JP, Bu H (2007) Correction of Simple Contrast Loss in Color Images. IEEE Trans Image Proc 16(2):511-522
46. Narasimhan SG, Nayar SK (2002) Vision and the atmosphere. Int J Comput Vis 48(3):233-254
47. Schechner YY, Narasimhan SG, Nayar SK (2003) Polarization-based vision through haze. Appl Opt 42(3)
48. Garg K, Nayar SK (2004) Photometric Model of a Rain Drop. In: Technical Report, Department of Computer Science, Columbia University
49. Wilfred P, Shah M, Lobo NV (2002) FlameRecognition in Video. Pattern Recognit Lett 23(1-3):319–327
50. Garg K, Nayar SK (2005) When Does a Camera See Rain? In: Proc of IEEE Int Conf Comput Vis 2:1067–1074
51. Liu CB, Ahuja N (2004) Vision Based Fire Detection. In: Proc of Int Conf Pattern Recognit 4
52. Fastcom Tech.SA, Blvd. de Grancy 19A, CH-1006 Lausanne, Switzerland: Method and Device for Detecting Fires Based on Image Analysis. In: Patent Coop. Treaty(PCT) Appl.No: PCT/CH02/00118, PCT Pubn.No: WO02/069292.
53. Garg K, Nayar SK (2004) Detection and Removal of Rain from Videos. In: Proc of Comput Vis Pattern Recognit 1:528–535
54. Casella G, Berger R (2002) Statistical Inference, 2nd edition. Duxbury Press