Video Surveillance and Multimedia Forensics: an Application to Trajectory Analysis

Simone Calderara
DII - Univ. of Modena-Reggio Em.
Via Vingolese, 905/b
Modena, Italy
simone.calderara@unimore.it

Andrea Prati
DiSMI - Univ. of Modena-Reggio Em.
Via Amendola, 2 - Pad. Morselli
Reggio Emilia, Italy
andrea.prati@unimore.it

Rita Cucchiara
DII - Univ. of Modena-Reggio Em.
Via Vingolese, 905/b
Modena, Italy
rita.cucchiara@unimore.it

ABSTRACT

This paper reports an application of trajectory analysis in which forensics and video surveillance techniques are jointly employed for providing a new tool of multimedia forensics. Advanced video surveillance techniques are used to extract from a multi-camera system the trajectories of the moving people which are then modelled by either their positions (projected on the ground plane) or their directions of movement. Both these two representations can be very suitable for querying large video repositories, by searching for similar trajectories in terms of either sequences of positions or trajectory shape (encoded as sequence of angles, where positions do not care). Preliminary examples of the possible use of this approach are shown.

1. INTRODUCTION

Digital forensics is a well-established discipline conveying ICT (Information and Communication Technology), computer hardware and software technologies for processing sensors-based and multimedia data for forensics. The involved techniques and the implemented tools are mainly based on manual intervention and feedback of humans involved in investigation actions. New researches in the area of human-centered multimedia interaction can be useful in forensics for multimedia data management, indexing, searching and querying. A huge amount of annotated data must be available and retrieved starting from partial and imprecise queries and often visual information is particularly suitable, especially if presented in the actual environment of the crime scene. Up to now the investigators was mainly oriented in the use of practical manual tools capable to extract precise measures from images, e.g. for people authentication, face and fingerprint identification or other biometrics, 3D scene reconstructions for ballistic measures, etcetera, by exploiting more or less sophisticated computer vision techniques. For example, several image processing techniques have been implemented in a tool [6] for video filtering and data extraction for forensic applications. As a consequence of this interest, the name of video forensics is now spreading, especially in commercial world. It is now the time that other fields involved in video analysis, in particular multimedia processing and video surveillance, step forward to join with video forensics to re-use and better finalize paradigms and models of data evaluations.

Multimedia technologies are mature enough for retrieving concepts, data and events in videos. A good survey of event mining in media streams is reported in [15], discussing the six common-sense aspect 5W1H (who?, when?, where?, what?, why?, and how?) to characterize events by journalism principles. At the same time, video surveillance research has done a more-than-ten-year effort in extracting similar knowledge from videos with the additional challenge of the real-time requirements. In surveillance, the corresponding pioneering work of W4 (Who?, When?, Where? and What?) [5] introduced for the first time similar concepts.

Focusing on video analysis for people information extraction, video surveillance and video forensics share models and techniques. People video surveillance aims at detecting and tracking the movement of single person, group of people and crowds in order to recognize in real-time specific situations which could gather the attention for security and safety purposes. Some examples are: the growing of queues in public environments such as stations or metros; the presence of single or group of people stopped in dangerous areas, such as near train platforms; the suspicious paths and activities of people in open areas, such as in parking zones; the interaction between people and objects, such as for abandoning packs or luggage; and so on [12, 8].

In these situations, some tight requirements are the system reactivity, the reliability on a 24/7 basis, and the high detection rate. In synthesis, video surveillance looks for few well-defined models of behaviours which must be detected fully automatically and with high reliability, possibly with limited false alarms. There is a weaker interest in accuracy and people identification, so that tracking techniques focusing on region of interest only are suitable enough to cope with the limited requirement of target localization in cluttered environments.

People video forensics, instead, aims at searching, identifying and measuring people, actions, interactions and related objects which could be of interest for investigations. A large effort is devoted to image processing tools for improving visual quality to human analysis. Another specific area of interest is biometry for face, fingerprint and other visual feature recognition, if the frame resolution and quality are satisfactory enough. A system where conventional automatic video surveillance, object detection and tracking, and biometric inspection, fingerprint recognition, synergistically cooperate for detecting unauthorized access is presented in [9].

Finally, an emerging area addresses the techniques for searching

MiFor'09, October 23, 2009, Beijing, China.

and mining in large video repositories in order to find situations of interest: here, efficient and fast techniques for video analysis are required, where the time constraints are not bounded by the frame rate but by the large amount of data.

One of the most emerging needs in forensic analysis is the presence of tools for automatic understanding of people behavior that simplify significantly the investigation reducing the time spent in searching for video segment containing sequences of interest. Among the many different behaviors, people trajectories are one of the most informative and can be reliably extracted automatically with conventional video surveillance systems. The amount of this kind of data could increase rapidly in crowded scenarios, becoming impossible to be analyzed without the aim of automatic tools.

In this paper we propose a method for comparing trajectories analyzing different characteristics: trajectories shape and trajectories positions in a given scene. The shape analysis is important when infrequent 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.

2. TRAJECTORY MODEL FOR POSITIONAL ANALYSIS

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. Both these representations can be useful in forensic applications, depending on which type of analysis is required: spatial locations can be used to infer the passage on forbidden areas, the typical entry-exit points in the scene and for querying by similarity given a certain path; directional representation, instead, describes the shape of the trajectory and can be used to search for similar paths.

2.1 Spatial Model

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 forensic analysis. 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 $T_k = \{\mathbf{t_{1,k}} \ldots \mathbf{t_{n_k,k}}\}$, 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 $\mathbf{\Sigma}$ can be defined as:

$$\mathcal{N}_{i,k} = \mathcal{N}(x, y \mid \boldsymbol{\mu_{i,k}}, \mathbf{\Sigma}) \tag{1}$$

An example of the fitting of Gaussians onto the trajectory points is shown in Fig. 1, 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 (1).

2.2 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 $[0, 2\pi)$:

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

In order to analyze its shape, *circular* or *directional statistics* [10] 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 [4, 1]. 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:

$$\mathcal{V}\left(\theta|\theta_0, m\right) = \frac{1}{2\pi I_0\left(m\right)} e^{m\cos(\theta-\theta_0)} \tag{3}$$

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

$$I_0\left(m\right) = \frac{1}{2\pi} \int_0^{2\pi} e^{m\cos\theta} d\theta \tag{4}$$

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:

$$p\left(\theta\right) = \sum_{k=1}^{K} \pi_k \mathcal{V}\left(\theta|\theta_{0,k}, m_k\right) \tag{5}$$

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 [13].

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:

$$S_{i,j} = \underset{r=1,\ldots,K}{\arg\max}\, p\left(\theta_{0,r}, m_r|\theta_{i,j}\right) = \underset{r=1,\ldots,K}{\arg\max}\, p\left(\theta_{i,j}|\theta_{0,r}, m_r\right) \tag{6}$$

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 $\overline{T}_j = \left\{S_{1,j}, S_{2,j}, ..., S_{n_j,j}\right\}$.

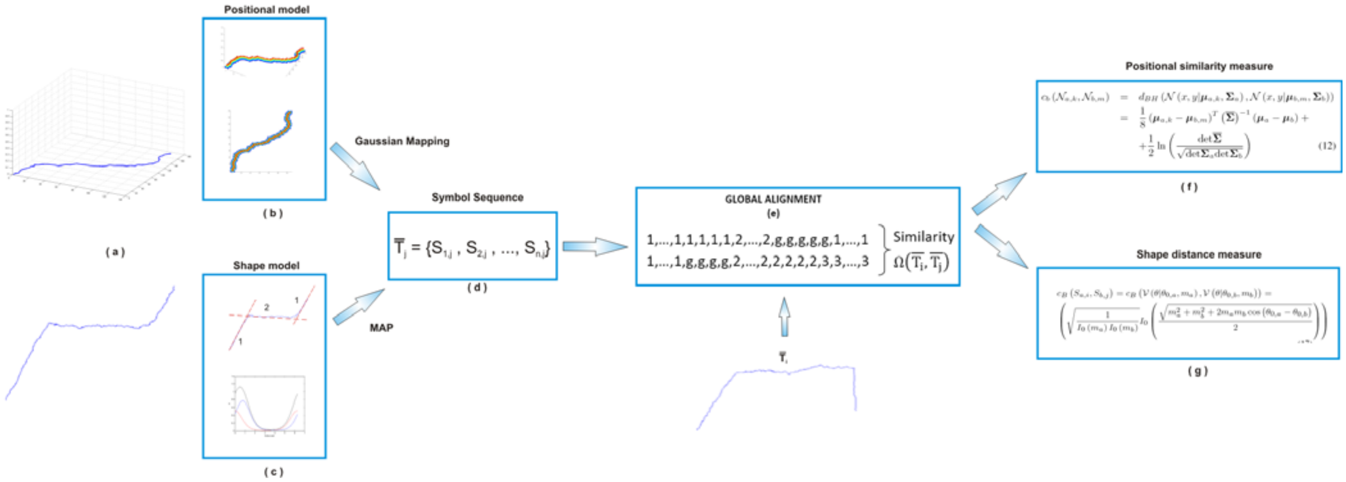
Positional model
(b)
(a)
Gaussian Mapping
Shape model
2
1
1
(c)
MAP
Symbol Sequence
$\overline{T}_j = \{S_{1,j}, S_{2,j}, ..., S_{n,j}\}$
(d)
GLOBAL ALIGNMENT
(e)
1,...,1,1,1,1,1,2,...,2,g,g,g,g,g,1,...,1
1,...,1,g,g,g,g,2,...,2,2,2,2,2,3,3,...,3
Similarity
$\Omega(\overline{T}_i, \overline{T}_j)$
$\overline{T}_i$
Positional similarity measure
$c_b(\mathcal{N}_{a,k}, \mathcal{N}_{b,m}) = d_{BH}(\mathcal{N}(x,y|\mu_{a,k},\Sigma_a), \mathcal{N}(x,y|\mu_{b,m},\Sigma_b)) = \frac{1}{8}(\mu_{a,k}-\mu_{b,m})^T(\Sigma)^{-1}(\mu_a-\mu_b) + +\frac{1}{2}\ln\left(\frac{\det\Sigma}{\sqrt{\det\Sigma_a\det\Sigma_b}}\right)$ (12)
(f)
Shape distance measure
$c_B(S_{a,i}, S_{b,j}) = c_B(\mathcal{V}(\theta|\theta_{0,a}, m_a), \mathcal{V}(\theta|\theta_{0,b}, m_b)) = \left(\sqrt{\frac{1}{I_0(m_a)I_0(m_b)}} I_0\left(\frac{\sqrt{m_a^2+m_b^2+2m_am_b\cos(\theta_{0,a}-\theta_{0,b})}}{2}\right)\right)$
(g)

Figure 1: Example of the trajectory model.

3. 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 [11] 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 $O\left(n_i \cdot n_j\right)$, 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:

$$
\begin{aligned}
V(a, 0) &= \Omega\left(S_{a,i}, -\right) \\
V(0, b) &= \Omega\left(-, S_{b,j}\right)
\end{aligned}
\tag{7}
$$

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:

$$V(a,b) = \max \begin{cases} V(a-1,b-1) + \Omega\left(S_{a,i}, S_{b,j}\right) \\ V(a-1,b) + \Omega\left(S_{a,i}, -\right) \\ V(a,b-1) + \Omega\left(-, S_{b,j}\right) \end{cases} \tag{8}$$

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:

$$\Omega\left(S_i, T_j\right) = \begin{cases} 2 \cdot (c_B) & \text{if} \quad c_B \geq 0.5 \\ 2 \cdot (c_B - 0.5) & \text{if} \quad c_B < 0.5 \\ 0 & \text{if} \quad S_i \text{ or } T_j \text{ are gaps} \end{cases} \tag{9}$$

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. STATISTICS SYMBOL-TO-SYMBOL DISTANCE METRICS

4.1 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:

$$\begin{aligned} c_b\left(\mathcal{N}_{a,k}, \mathcal{N}_{b,m}\right) &= d_{BH}\left(\mathcal{N}\left(x,y|\boldsymbol{\mu}_{a,k}, \boldsymbol{\Sigma}_a\right), \mathcal{N}\left(x,y|\boldsymbol{\mu}_{b,m}, \boldsymbol{\Sigma}_b\right)\right) \\ &= \frac{1}{8}\left(\boldsymbol{\mu}_{a,k} - \boldsymbol{\mu}_{b,m}\right)^T \left(\overline{\boldsymbol{\Sigma}}\right)^{-1}\left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right) + \\ &\quad + \frac{1}{2}\ln\left(\frac{\det\overline{\boldsymbol{\Sigma}}}{\sqrt{\det\boldsymbol{\Sigma}_a \det\boldsymbol{\Sigma}_b}}\right) \end{aligned} \tag{10}$$

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:

$$c_b\left(\mathcal{N}_a^k, \mathcal{N}_b^m\right) = \frac{1}{8}\left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right)^T \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\mu}_a - \boldsymbol{\mu}_b\right) \tag{11}$$

As previously performed for the angular model this coefficient can be injected in equation (9) to obtain the symbol to symbol similarity measure used in the alignment process.

4.2 Distance in the case of Angular Model

When the data sequences is modeled using the Mixture of Von Mises Model, Section 2.2, one possible symbol-to-symbol distance between the univariate pdf associated to each symbol, following the scheme of Fig. 1, is the Bhattacharyya coefficient between Von Mises distribution,[7]. 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:

$$c_B\left(p,q\right) = \int_{-\infty}^{+\infty} \sqrt{p\left(\theta\right) q\left(\theta\right)} d\theta \tag{12}$$

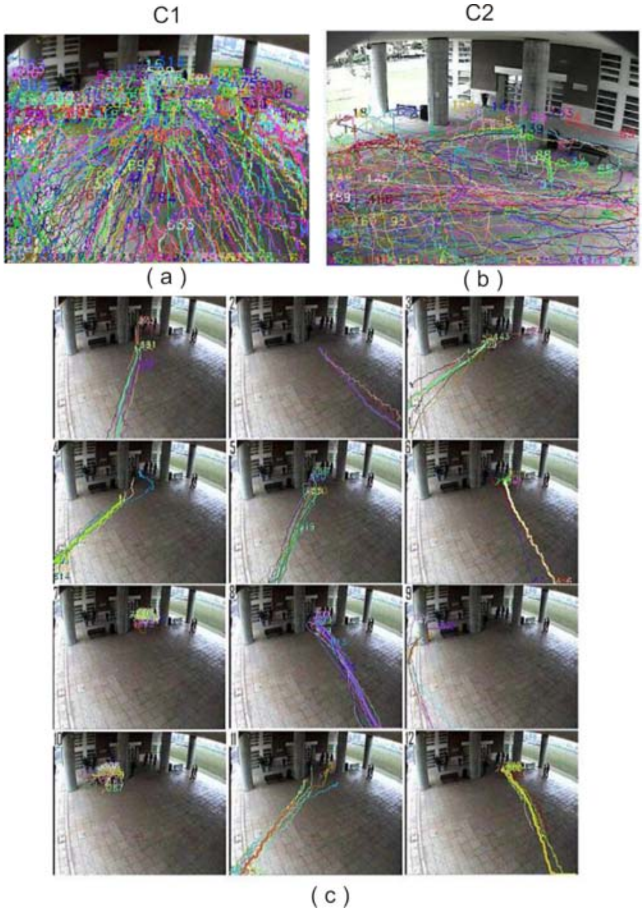

Figure 2: In (a) and (b) is shown the training set used during the learning stage. (c) shows the obtained most frequent behaviors projected on the $C1$ view.

following the derivation in [13] for two univariate Von Mises distribution the analytic form of the coefficient results:

$$c_B\left(S_i, T_j\right) = c_B\left(\mathcal{V}\left(\theta|\theta_{0,i}, m_i\right), \mathcal{V}\left(\theta|\theta_{0,j}, m_j\right)\right) = \left(\sqrt{\frac{1}{I_0\left(m_a\right) I_0\left(m_b\right)}} I_0\left(\frac{\sqrt{m_i^2 + m_j^2 + 2m_i m_j \cos\left(\theta_{0,i} - \theta_{0,j}\right)}}{2}\right)\right) \tag{13}$$

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

5. EXPERIMENTS FOR VIDEO FORENSICS

Once a proper similarity measure is available, sequences can be compared according to either their position (Section 4.1) or their shape (Section 4.2). 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 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 [14] clustering algorithm using the similarity measures introduced in section 3.

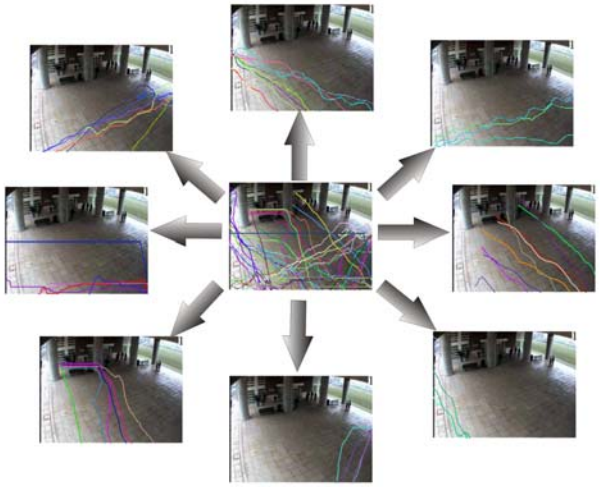

Figure 3: The center figure shows the training set used for trajectories shape clustering. The remaining figure shows the most frequent behaviors according to their shape.

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 $TS = \{T_1, \cdots, T_N\}$ 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. 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:

$$\widetilde{j} = \underset{j=1,\ldots,\widetilde{k}}{\arg\max}\ \Omega\left(C_j, T_{new}\right) \tag{14}$$

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:

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

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:

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

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.

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 [3, 2]. 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. 2, 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. 3 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.

A typical reference application is shown in Fig. 4. Here it is depicted how the system could be useful in forensic application. First, a query is performed on the trajectory shape, Fig. 4.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

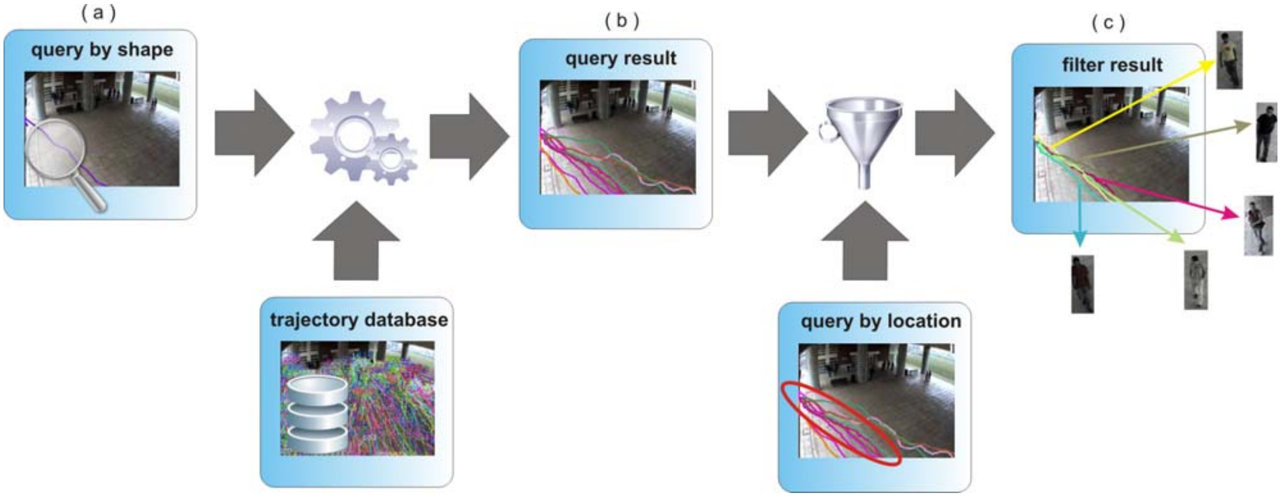

Figure 4: Example of a possible Forensic Application of the trajectory analysis framework.

to the desired one (Fig. 4.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.

6. CONCLUSIONS

The proposed system allows the detection of people walking and standing in surveilled areas and the extraction of their trajectories. This approach can be directly applied in video forensics for extracting valuable information about the paths and for comparing them according to their shape or position. This system can be part of a multimedia forensic tool that could improve and speed up the investigation process. Results are promising and several additional information could be incorporated in the presented model to extend the range of possible queries that could be presented to the system.

7. REFERENCES

[1] C. Bishop. *Pattern Recognition and Machine Learning*. Springer–Verlag, 2006.

[2] S. Calderara, R. Cucchiara, and A. Prati. Bayesian-competitive Consistent Labeling for People Surveillance. *IEEE Trans. on PAMI*, 30(2):354–360, Feb. 2008.

[3] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati. Detecting moving objects, ghosts and shadows in video streams. *IEEE Trans. on PAMI*, 25(10):1337–1342, Oct. 2003.

[4] R. Fisher. Dispersion on a sphere. *Proc. Roy. Soc. London Ser. A.*, 217:295–305, 1953.

[5] I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *IEEE Trans. on PAMI*, 22(8):809–830, Aug. 2000.

[6] M. Jerian, S. Paolino, F. Cervelli, S. Carrato, A. Mattei, and L. Garofano. A forensic image processing environment for investigation of surveillance video. In *Proc. of 4th European Academy of Forensic Science Conference, Helsinki, EAFS2006,*, 2006.

[7] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Transactions on Communication Technology*, COM-15(1):52–60, 1967.

[8] G. Lavee, L. Khan, and B. M. Thuraisingham. A framework for a video analysis tool for suspicious event detection. *Multimedia Tools Appl.*, 35(1):109–123, 2007.

[9] G. L. Marcialis, F. Roli, P. Andronico, P. Multineddu, P. Coli, and G. Delogu. Video biometric surveillance and forensic image analysis. In *First Workshop on Video Surveillance projects in Italy (VISIT 2008)*, Modena (Italy), 22/05/2008 2008.

[10] K. Mardia and P. Jupp. *Directional Statistics*. Wiley, 2000.

[11] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453, 1970.

[12] V. Parameswaran and R. Chellappa. View invariance for human action recognition. *International Journal of Computer Vision*, 66(1):83–101, 2006.

[13] A. Prati, S. Calderara, and R. Cucchiara. Using circular statistics for trajectory shape analysis. In *CVPR*, 2008.

[14] A. Reynolds, G. Richards, and V. Rayward-Smith. *The Application of K-Medoids and PAM to the Clustering of Rules*, volume 3177/2004, pages 173–178. Springer Berlin / Heidelberg.

[15] L. Xie, H. Sundaram, and M. Campbell. Event mining in multimedia streams. *Proceedings of the IEEE*, 96(4):623–647, April 2008.