Using Circular Statistics for Trajectory Shape Analysis

Andrea Prati[1], Simone Calderara[2], Rita Cucchiara[2]
[1] Di.S.M.I. - Via Amendola, 2 - 42100 Reggio Emilia; [2] D.I.I. - Via Vignolese, 905 - 41100 Modena
University of Modena and Reggio Emilia, Italy

Abstract

The analysis of patterns of movement is a crucial task for several surveillance applications, for instance to classify normal or abnormal people trajectories on the basis of their occurrence. This paper proposes to model the shape of a single trajectory as a sequence of angles described using a Mixture of Von Mises (MoVM) distribution. A complete EM (Expectation Maximization) algorithm is derived for MoVM parameters estimation and an on-line version proposed to meet real time requirement. Maximum-A-Posteriori is used to encode the trajectory as a sequence of symbols corresponding to the MoVM components. Iterative k-medoids clustering groups trajectories in a variable number of similarity classes. The similarity is computed aligning (with dynamic programming) two sequences and considering as symbol-to-symbol distance the Bhattacharyya distance between von Mises distributions. Extensive experiments have been performed on both synthetic and real data.

1. Introduction

Many surveillance-related scenarios require to learn patterns of movement. Among the possible patterns descriptors, *trajectory shape* is one of the most common and discriminant. Indeed, recent advances in object tracking algorithms made it possible to obtain reliable and long trajectories for moving people. The analysis of such trajectories allows to analyze people behavior and identify suspicious or abnormal movements by detecting patterns that deviate from "normal" cases. Without any a-priori knowledge, "normal" patterns can be only modeled by considering their statistical occurrence. For this reason, it could be convenient to learn the probability distribution of "normal" patterns and then, given a new pattern, evaluate its fitness with the posterior probability using a similarity measure. The challenge is to define a correct and stable similarity measure between trajectories. Due to its generality, this low-level classification based on the occurrence only can be the starting point for more general behavior recognition systems and coupled easily with context-based rules.

The people trajectory projected on the ground plane is a very compact representation of patterns of movement, normally characterized by a sequence of 2D data (x-y coordinates). In this paper we propose a 1D representation of trajectories based on a sequence of angles, and a new statistical representation of the patterns by means of a *mixture of von Mises* (MoVM) distributions.

The description by means of a sequence of angles exhibits several advantages: first, it is a more compact representation; second, it is invariant to spatial translation of the starting point of the trajectory; third, by using absolute angles and eliminating spatial references, the shape similarity can be measured locally, independently by spatial shifts of parts of the trajectories. Moreover, supposing a constant sample rate of points in the trajectory, the sequence of angles also considers speed, since when a person moves fast along a given direction he will generate less points (i.e., less angles) wrt the case when he moves slowly.

The main objective of this paper is to demonstrate the usefulness of circular statistics (in particular, von Mises distribution) for statistical pattern recognition in the field of motion analysis. The adoption of MoVM requires the definition of the complete procedure to derive the mixture parameters from samples, to compare the trajectories and cluster them in groups. The main contributions of this paper can be summarized in: (i) the *derivation of EM* (Expectation-Maximization) formulation for modeling a single trajectory with a MoVM; (ii) each trajectory in the training set is compared with the others using an *inexact matching* in order to consider temporal shifts and noise; (iii) *closed form of the Bhattacharyya distance* between von Mises distributions is used to compute the global distance between aligned trajectories; (iv) *iterative k-medoids* is exploited to cluster the observed trajectories in a certain (not fixed) number of clusters, where each cluster has associated a prior probability representing whether it corresponds to "normal" trajectories or not; (v) for new trajectories, *sufficient statistics* for a MoVM are derived and used for on-line updating of the mixture parameters (to meet real-time requirements) and compared with the available cluster prototypes (or medoids) to dynamically update the clusters.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

2. Related Works

Although clustering and classification of trajectories is a relatively new problem, it has been deeply studied in the last years. Many approaches have been proposed using both robust statistics and clustering techniques. In [3] the EM framework is adopted to learn a set of motion models' parameters given the spatial position of the points of each trajectory as observation. Panozzo *et al.* in [17] 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 [13] Junejo *et al.* applied graph cuts to cluster trajectories using the Hausdorff distance measure. In [12] a system for learning statistical motion patterns is presented. Trajectory clustering is performed using a two-stage fuzzy k-means. First, trajectories are clustered in the spatial domain, then each cluster is subclustered in the temporal domain. New trajectories are checked against cluster centers using Bayes rule and anomaly detection is performed by thresholding resulting probability.

Many methods ranging from simple Euclidean distance to complex *Dynamic Time Warping* have been applied for comparing trajectories. Euclidean distance was proposed in [10] as the simplest example of trajectory similarity measure. Although the poor computational cost, this approach is suitable for comparing trajectories that have the same length that results unfeasible in many real applications. To strengthen the comparison, Principal Component Analysis was performed and then Euclidean measure applied [2]. Hausdorff distance was proposed by both Lou *et al.* [15] and Junejo *et al.* [13]. This measure only performs pairwise comparison and is unable to distinguish trajectories in the same path but opposite direction. Alignment technique like *Longest Common SubSequence* (LCSS) and *Dynamic Time Warping* have been efficiently applied to compare trajectories shape in sign language recognition and surveillance applications [14, 5, 22]. Both these methods are able to compare trajectories having different lengths by finding the best alignment but perform poorly in dealing with noisy data because of the pairwise comparison adopted. Statistics can partially mitigate the problem of noise in acquired data. Porikli [18] proposes to use 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. HMMs have been also used in [1] for classifying the activities.

In this paper we propose a novel method to compare trajectories shapes that share the virtues of alignment-based techniques (such as [14, 5, 22]) and robustly deals with noise using circular statistics to match data.

3. Learning the Trajectory Model

The complete procedure for learning the trajectory model is sketched in Fig. 1, where all the phases are shown with reference to a given example.

3.1. Phase (a): Trajectory modeling

Using a constant frame rate, we model the single trajectory T_j as a sequence of n_j directions θ, defined in $[0, 2\pi)$:

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

Circular or *directional statistics* [16] 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 [9]. The von Mises-Fisher distribution is a probability distribution on the $(p-1)$-dimensional sphere in $\mathbb{R}^p$ defined as:

$$f_p\left(\overline{\boldsymbol{\theta}}|\theta_0, m\right) = C_p\left(m\right) e^{m\theta_0^T \overline{\boldsymbol{\theta}}} \tag{2}$$

where $\overline{\boldsymbol{\theta}}$ is the multi-variate (p-dimensional) random unit vector, $m \geq 0$ denotes the *concentration* parameter (analogous to the precision or inverse of the variance) and θ_0 the *mean*. The normalization constant $C_p\left(m\right)$ is equal to:

$$C_p\left(m\right) = \frac{m^{p/2-1}}{\left(2\pi\right)^{p/2} I_{p/2-1}\left(m\right)} \tag{3}$$

where I_j represents the modified Bessel function of the first kind and order j. In general, parameter m denotes how much concentrate is the distribution around the mean θ_0 and for $m = 0$ the distribution collapses in a uniform distribution on the p-dimensional sphere. For $p = 2$ the distribution reduces to the *von Mises distribution* on the circle.

The von Mises distribution [9, 4] is also known as the *circular normal* or the *circular Gaussian*, and it is particularly useful for statistical inference of angular data. In fact, it yields many of the key properties for statistical inference that the normal distribution has for linear data.

When the variable is univariate, the probability density function (pdf) results to be:

$$\mathcal{V}\left(\theta|\theta_0, m\right) = \frac{1}{2\pi I_0\left(m\right)} e^{m\cos(\theta-\theta_0)} \tag{4}$$

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

$$I_0\left(m\right) = \frac{1}{2\pi} \int_0^{2\pi} e^{m\cos\theta} d\theta \tag{5}$$

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. Please note that for large m this distribution shape is, for 2π interval, approximately a Gaussian centred in θ_0.

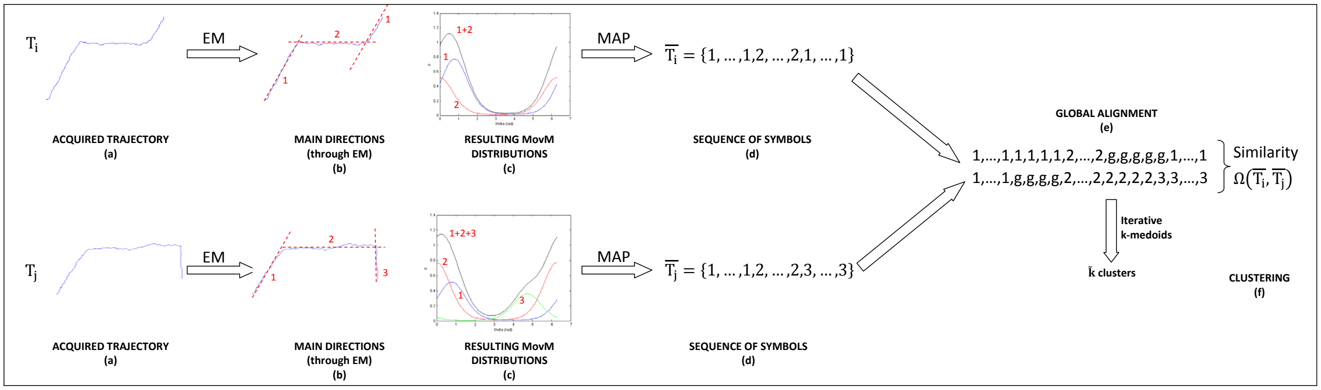

T_i
EM
1
2
1
1+2
1
2
MAP
T̄_i = {1, … ,1,2, … ,2,1, … ,1}
ACQUIRED TRAJECTORY (a)
MAIN DIRECTIONS (through EM) (b)
RESULTING MovM DISTRIBUTIONS (c)
SEQUENCE OF SYMBOLS (d)
GLOBAL ALIGNMENT (e)
1,…,1,1,1,1,1,2,…,2,g,g,g,g,g,1,…,1
1,…,1,g,g,g,g,2,…,2,2,2,2,2,3,3,…,3
Similarity Ω(T̄_i, T̄_j)
Iterative k-medoids
k̃ clusters
CLUSTERING (f)
T_j
EM
2
1
3
1+2+3
2
1
3
MAP
T̄_j = {1, … ,1,2, … ,2,3, … ,3}
ACQUIRED TRAJECTORY (a)
MAIN DIRECTIONS (through EM) (b)
RESULTING MovM DISTRIBUTIONS (c)
SEQUENCE OF SYMBOLS (d)

Figure 1. Description of the procedure for learning trajectory model.

Von Mises distribution has been often adopted in directional statistics because it results much more suitable to treat periodic variables. In fact, Gaussians are often used since they are well known and suitable statistical treatment has been deeply analyzed. However, in the case of angular variables, Gaussians can be affected by errors due to the discontinuity in the origin. Bishop in [4] well described this limitation which does not affect von Mises distribution because it is independent on the origin, being “circularly defined”.

3.2. Phases (b)-(c): Inferring MoVM parameters through EM

Von Mises distribution is thus an ideal pdf to describe a trajectory T_j. 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 multimodal pdf, and thus we propose the use of a mixture of von Mises distributions. 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 “responsabilities” of a given sample with respect to a given component of the mixture).

The MoVM is defined as:

$$p\left(\theta\right)=\sum_{k=1}^{K}\pi_k\mathcal{V}\left(\theta|\theta_{0,k},m_k\right) \tag{6}$$

Given a set $\boldsymbol{\theta}$ of i.i.d. observations (the directions of T_j) and the set of corresponding latent points $\boldsymbol{Z}$ (introduced with a 1-of-K variable z_i), we can write the likelihood of the *complete data set* $\{\boldsymbol{\theta},\boldsymbol{Z}\}$ as follows:

$$p\left(\boldsymbol{\theta},\boldsymbol{Z}|\boldsymbol{\pi},\boldsymbol{\theta_0},\boldsymbol{m}\right)=\prod_{i=1}^{n_j}\prod_{k=1}^{K}\pi_k^{z_{ik}}\mathcal{V}\left(\theta_i|\theta_{0,k},m_k\right)^{z_{ik}} \tag{7}$$

where z_{ik} denotes the k^{th} component of z_i and $\boldsymbol{\pi}$, $\boldsymbol{\theta_0}$ and $\boldsymbol{m}$ represent the mixture parameters. Thus, the expected value $\mathbb{E}_{\boldsymbol{Z}}\equiv\mathbb{E}_{\boldsymbol{Z}}\left[\ln p\left(\boldsymbol{\theta},\boldsymbol{Z}|\boldsymbol{\pi},\boldsymbol{\theta_0},\boldsymbol{m}\right)\right]$ of the log likelihood for the complete data set results to be:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{Z}}&=\sum_{i=1}^{n_j}\sum_{k=1}^{K}\gamma_{ik}\left\{\ln\pi_k+\ln\mathcal{V}\left(\theta_i|\theta_{0,k},m_k\right)\right\}\\&=\sum_{i=1}^{n_j}\sum_{k=1}^{K}\gamma_{ik}\left\{\ln\pi_k-\ln 2\pi I_0\left(m_k\right)+m_k\cos\left(\theta_i-\theta_{0,k}\right)\right\}\end{aligned} \tag{8}$$

where $\gamma_{ik}\equiv\mathbb{E}\left[z_{ik}\right]$ is the responsability of component k for data point θ_i, and can be estimated during the E step using the parameter values of the previous iteration (randomly initialized for the first iteration) as follows:

$$\gamma_{ik}=\frac{\pi_k\mathcal{V}\left(\theta_i|\theta_{0,k},m_k\right)}{\sum\limits_{s=1}^{K}\pi_s\mathcal{V}\left(\theta_i|\theta_{0,s},m_s\right)} \tag{9}$$

The M step maximizes the log likelihood of eq. 8 with respect to each of the parameters $\left(\boldsymbol{\pi},\boldsymbol{\theta_0},\boldsymbol{m}\right)$. Computing the derivative of eq. 8 with respect to π_k by making use of a Lagrange multiplier, we can obtain:

$$\pi_k=\frac{1}{n_j}\sum_{i=1}^{n_j}\gamma_{ik} \tag{10}$$

Deriving eq. 8 on $\theta_{0,k}$ and setting the result equal to zero, leads to the following equation:

$$\sum_{i=1}^{n_j}\gamma_{ik}m_k\sin\left(\theta_i-\theta_{0,k}\right)=0$$

$$\sum_{i=1}^{n_j}\gamma_{ik}m_k\left(\sin\theta_i\cos\theta_{0,k}-\cos\theta_i\sin\theta_{0,k}\right)=0$$

$$\theta_{0,k}=\arctan\left(\frac{\sum\limits_{i=1}^{n_j}\gamma_{ik}\sin\theta_i}{\sum\limits_{i=1}^{n_j}\gamma_{ik}\cos\theta_i}\right) \tag{11}$$

Maximization with respect to m_k is more complicated due to the presence of the Bessel function:

$$\frac{\partial \mathbb{E}_{\boldsymbol{Z}}}{\partial m_k} = \frac{I_1\left(m_k\right)}{I_0\left(m_k\right)} \sum_{i=1}^{n_j} \gamma_{ik} + \sum_{i=1}^{n_j} \gamma_{ik} \cos\left(\theta_i - \theta_{0,k}\right) \quad (12)$$

where $I_1\left(m\right)$ is the modified Bessel function of the first kind of order 1 and is the first derivative of $I_0\left(m\right)$. Equation 12 can be solved in m_k by introducing a function $A\left(m_k\right) = \frac{I_1(m_k)}{I_0(m_k)}$ [4]:

$$A\left(m_k\right) = \frac{\sum\limits_{i=1}^{n_j} \gamma_{ik} \cos\left(\theta_i - \theta_{0,k}\right)}{\sum\limits_{i=1}^{n_j} \gamma_{i,k}} \quad (13)$$

By finding the zeros of this function and then inverting the function numerically, we can obtain the value of m_k.

3.3. Phase (d): Trajectory encoding through MAP

Equations 10, 11 and 13 allow the computation of the parameters for the K components. If the trajectory T_j contains less than K main directions, some components will have similar parameters. Each direction $\theta_{i,j}$ is encoded with a symbol $S_{i,j}$ with a Maximum-A-Posteriori (MAP) approach, that, assuming uniform priors, can be written as:

$$S_{i,j} = \arg\max_{r=1,\dots,K} p\left(\theta_{0,r}, m_r | \theta_{i,j}\right) = \arg\max_{r=1,\dots,K} p\left(\theta_{i,j} | \theta_{0,r}, m_r\right) \quad (14)$$

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

3.4. Phase (e): Global alignment of symbol sequences

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. We can borrow from bioinformatics a method for comparing sequences in order to find the best inexact matching between them, also accounting for gaps. Among the many different techniques, we used the *global alignment* [11]. Using global alignment instead of local one is preferable because the former preserves both global and local shape characteristics. Global alignment of two sequences S and T is obtained by first inserting spaces, either into or at the ends of S and T so that the length of the sequences will be the same, and then placing the two resulting sequences one above the other so that every symbol or space in one of the sequences is matched to a unique symbol in the other (see Fig. 1). 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 results 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. Without going too much into details, dynamic programming overcomes to the problem of the recursive solution to global alignment by not comparing the same subsequences for more than one time, and exploiting tabular representation to efficiently compute the final similarity score (for further details refer to [11]).

Each element (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 symboling string recognition but it has not been used for trajectory data before since it can be affected by measurement noise. Our proposal overcomes this problem because each symbol corresponds to a von Mises distribution. Thus, the 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 much the score (penalty) should be negative. Assigning zero to the gap penalty, the best alignment can be found by searching for the alignment that maximizes the global score.

Specifically, we measured the distance between distributions p and q using the Bhattacharyya coefficient:

$$c_B\left(p, q\right) = \int_{-\infty}^{+\infty} \sqrt{p\left(\theta\right) q\left(\theta\right)} d\theta \quad (15)$$

It has been demonstrated [6] that if p and q are two von Mises distributions, the $c_B\left(p, q\right)$ can be computed in closed form as follows:

$$c_B\left(S_{a,i}, S_{b,j}\right) = c_B\left(\mathcal{V}\left(\theta|\theta_{0,a}, m_a\right), \mathcal{V}\left(\theta|\theta_{0,b}, m_b\right)\right) = \left(\sqrt{\frac{1}{I_0\left(m_a\right) I_0\left(m_b\right)}} I_0\left(\frac{\sqrt{m_a^2 + m_b^2 + 2 m_a m_b \cos\left(\theta_{0,a} - \theta_{0,b}\right)}}{2}\right)\right) \quad (16)$$

where it holds that $0 \leq c_B\left(S_{a,i}, S_{b,j}\right) \leq 1$.

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 as follows:

$$\sigma\left(S_{a,i},S_{b,j}\right)=\begin{cases}2\cdot\left(c_B\right) & \text{if} \quad c_B\geq 0.5\\ 2\cdot\left(c_B-0.5\right) & \text{if} \quad c_B<0.5\\ 0 & \text{if} \quad S_{a,i} \text{ or } S_{b,j} \text{ are gaps}\end{cases} \tag{17}$$

3.5. Phase (f): Clustering trajectories with iterative k-medoids algorithm

Once the score of the best global alignment is computed (as sum of the scores in the best alignment path), it can be converted in a proper similarity measure $\Omega\left(\overline{T}_i,\overline{T}_j\right)$. This measure is used to cluster the trajectories in the training set by using *k-medoids* algorithm [19]. This 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, at each iteration the prototype of each cluster is given by the member of the cluster at the minimum average distance from all the other members.

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. Let us set $i=0$ and $k(0)=N_t$, where N_t is the cardinality of the training set. As initialization, each trajectory is chosen as prototype (medoid) of the corresponding cluster. 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.

4. Trajectory Shape Classification

The described approach obtains robust unsupervised classification of trajectories. They are clustered in a variable number of similarity clusters. Clusters with fewer trajectories represent the case of abnormal or (better) "infrequent" trajectory shapes. New trajectories can be classified in normal/abnormal depending on the cardinality of the most similar cluster. In this case of application, we cannot employ a classical *learn-then-predict* paradigm in which the "knowledge" learned in the training phase is never updated. However, at the beginning an infrequent class of trajectories can be considered abnormal; if that type of trajectory is detected often, it should be considered normal, since in our scenario, instead, model of normality is neither a-priori known nor fixed. For this reason, we employ a *learn-and-predict* paradigm in which the knowledge (i.e., the trajectory clusters) are continuously updated.

Therefore, whenever a new trajectory T_{new} is collected, its statistical model is computed and compared to the cluster medoids. Based on this comparison, the trajectory can be classified as either belonging to an existing cluster or representing a new cluster (a class of trajectories never seen before).

In order to learn the trajectory model we could use the same EM algorithm described in Section 3.2. However, this is a very time consuming task which is unsuitable for real-time trajectory classification, even though it is acceptable for off-line learning. For this reason, similarly to what has been previously proposed for a mixture of Gaussians [21], we have derived an *on-line EM* algorithm for MoVMs.

On-line EM updating is based on the concept of *sufficient statistics*. A statistic $T(\theta)$ is sufficient for underlying parameter η if the conditional probability distribution of the data θ, given the statistic $T(\eta)$, is independent of the parameter η. Thanks to the Fisher-Neyman's factorization theorem [8], the likelihood function $L_\eta(\theta)$ of θ can be factorized in two components, one independent by the parameters η and the other dependent by them only through the sufficient statistics $T(\theta)$: $L_\eta(\theta)=h(\theta)g_\eta\left(T(\theta)\right)$. It has been shown [4] that in the case of distributions of the exponential family (such as Gaussian and von Mises) the factorization theorem can be written as:

$$p\left(\theta|\eta\right)=h\left(\theta\right)g\left(\eta\right)\exp\left\{\eta^T T(\theta)\right\} \tag{18}$$

Considering a von Mises distribution and a set $\boldsymbol{\theta}$ of i.i.d. angles (composing the trajectory T_j) we can decompose the expression of the distribution $p\left(\boldsymbol{\theta}|\theta_0,m\right)$ as follows:

$$\begin{aligned}&\prod_{i=1}^{n_j}\frac{1}{2\pi I_0(m)}\exp\left\{m\cos\left(\theta_i-\theta_0\right)\right\}=\\ &\frac{1}{\left(2\pi I_0(m)\right)^{n_j}}\exp\left\{m\sum_{i=1}^{n_j}\cos\left(\theta_i-\theta_0\right)\right\}=\\ &\frac{1}{\left(2\pi I_0(m)\right)^{n_j}}\exp\left\{m\cos\theta_0\sum_{i=1}^{n_j}\cos\theta_i+m\sin\theta_0\sum_{i=1}^{n_j}\sin\theta_i\right\}\\ &=\frac{1}{\left(2\pi I_0(m)\right)^{n_j}}\exp\left\{\begin{bmatrix}m\cos\theta_0\\ m\sin\theta_0\end{bmatrix}^T\cdot\begin{bmatrix}\sum_{i=1}^{n_j}\cos\theta_i\\ \sum_{i=1}^{n_j}\sin\theta_i\end{bmatrix}\right\}\end{aligned} \tag{19}$$

Thus, the sufficient statistics for a single von Mises distribution are $T\left(\boldsymbol{\theta}\right) = \begin{bmatrix} \sum\limits_{i=1}^{n_j} \cos\theta_i \\ \sum\limits_{i=1}^{n_j} \sin\theta_i \end{bmatrix}$. In the case of a mixture of distributions belonging to the exponential family, the online updating of the mixture parameters can be obtained by simply updating the sufficient statistics (s.s.) of the mixture, computed as $T_M\left(\boldsymbol{\theta}\right) = \sum\limits_{k=1}^{K} \gamma_k T_k\left(\boldsymbol{\theta}\right)$, where $T_k\left(\boldsymbol{\theta}\right)$ are the s.s. for the k^{th} single distribution. The updating process (having observed up to the sample $(i-1)$-th) can be obtained as:

$$T_k^i\left(\boldsymbol{\theta}\right) = \alpha(i)\gamma_k T_k\left(\theta_i\right) + \left(1-\alpha(i)\right) T_k^{i-1}\left(\boldsymbol{\theta}\right) \qquad (20)$$

where $T_k\left(\theta_i\right) = \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix}$. In [20] a comprehensive discussion on the value of the updating parameter $\alpha(i)$ is given.

Once the mixture parameters have been computed, the same MAP approach of Section 3.3 gives the symbol sequence $\overline{T}_{new}$. Given the set $M = \left\{M^1, ..., M^{\widetilde{k}}\right\}$ of current medoids, $\overline{T}_{new}$ is compared with each medoid to find the most similar, using the similarity measure Ω described in Section 3.4:

$$\widetilde{j} = \operatorname*{arg\,max}_{j=1,...,\widetilde{k}} \Omega\left(\overline{T}_{M^j}, \overline{T}_{new}\right) \qquad (21)$$

Defining the maximum similarity as $\Omega_{max} = \Omega\left(\overline{T}_{M^{\widetilde{j}}}, \overline{T}_{new}\right)$, if this value is below a given threshold Th_{sim} a new cluster should be created with T_{new} and the priors (proportional to the number of trajectories assigned to the cluster) updated:

$$\begin{aligned} & M^{\widetilde{k}+1} = T_{new}\,;\, p\left(M^{\widetilde{k}+1}\right) = \frac{1}{N+1} \\ & \forall i = 1, ..., \widetilde{k} \Rightarrow p^{new}\left(M^i\right) = p^{old}\left(M^i\right)\frac{N}{N+1} \\ & \widetilde{k} = \widetilde{k}+1\,;\, N = N+1 \end{aligned}$$

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 $\widetilde{j}$:

$$\begin{aligned} & T_{new} \in \text{cluster}\,\widetilde{j}\,;\, p^{new}\left(M^{\widetilde{k}}\right) = \frac{p^{old}\left(M^{\widetilde{k}}\right)\cdot N+1}{N+1} \\ & \forall i = 1, ..., \widetilde{k}, i \neq \widetilde{j} \Rightarrow p^{new}\left(M^i\right) = p^{old}\left(M^i\right)\frac{N}{N+1} \\ & N = N+1 \end{aligned}$$

Moreover, if the average similarity of the new trajectory wrt other medoids is smaller than the average similarity of the current medoid $M^{\widetilde{j}}$, T_{new} is a better medoid than $M^{\widetilde{j}}$ since it increases the separability with other clusters. Consequently, T_{new} becomes the new medoid of the cluster.

Finally, to avoid that old and rare trajectories could affect our model, clusters with small prior and with no new trajectories assigned for a fixed-length time window are dropped.

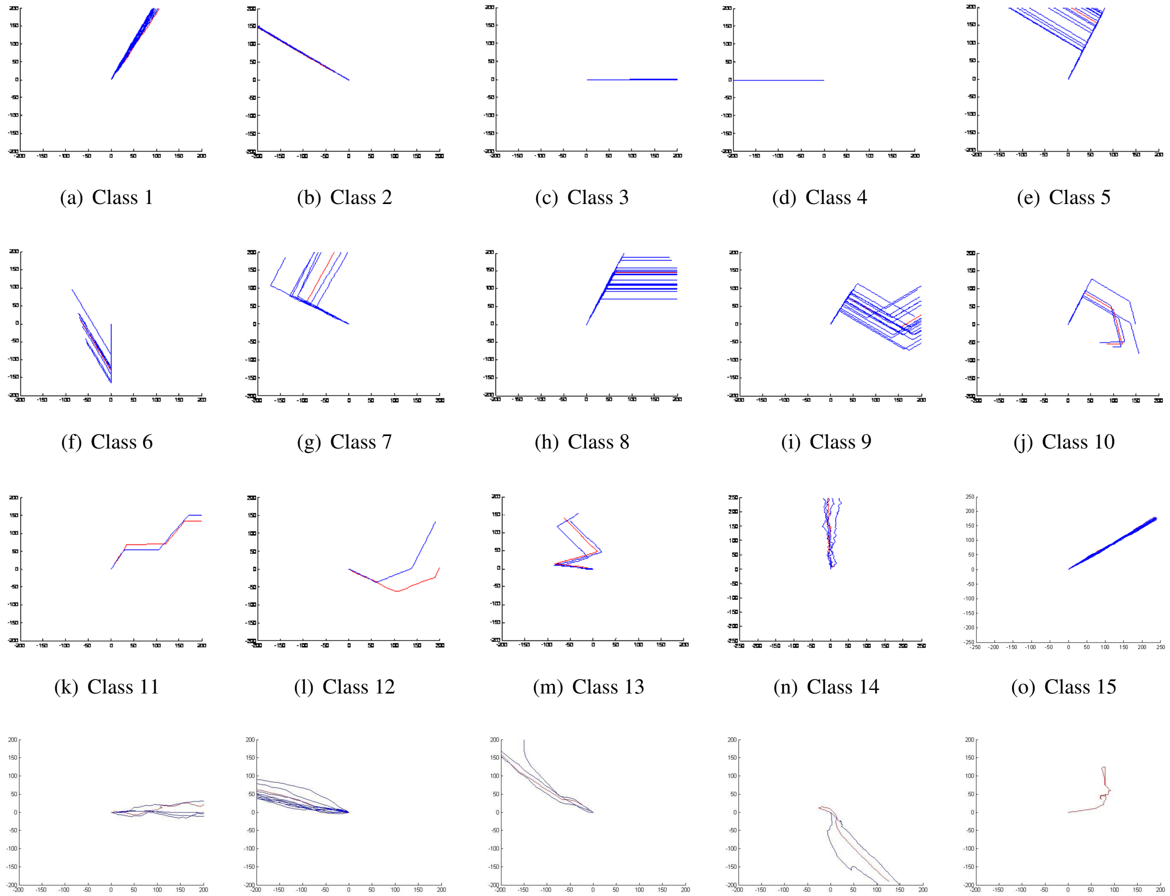
5. Experimental Results

The performance evaluation of the proposed model has been conducted on both synthetic and real data. The use of synthetic data is particularly useful since it allows to have a great amount of data and, in some way, their automatic ground truths. In order to analyze complex trajectories we specifically developed a trajectory generator in Matlab that produces trajectories similar to the ones collected with a real video-surveillance system. The generator allows the user to select graphically the sequence of angles composing the class of trajectories and then it creates a set of trajectories with added noise to both the single angles and their occurrences. Fig. 2 shows some examples of the classes of trajectories used in our tests. In the case of real data (see classes R1 ...R5) the trajectory's points are extracted from the scene using a multi-camera system called *HECOL* (Homography and Epipolar-based COnsistent Labeling) [7]. This system is capable of detecting and tracking multiple moving people by using multiple cameras with partially-overlapped fields of view that allow to exploit ground-plane homography and epipolar geometry for assigning a consistent label to different instances of the same person seen from different cameras/views. Directions are computed as the tangent between consecutive points with an average sampling rate of ten points per second. These trajectories reproduce typical motion patterns performed by people at our campus during several ordinary days.

Table 1 summarizes the performed tests. For each test, used classes of trajectories (with reference to Fig. 2) are depicted, where the asterisk represents an abnormal (infrequent) class. We evaluate for testing purposes both the overall classification accuracy (ability to assign the new trajectory to the correct cluster) and the normal/abnormal accuracy (ability to correctly classify the trajectory as normal or abnormal depending on the cardinality of the specific cluster).

Our system demonstrates optimal results both in classification test and normal-abnormal detection; results are remarkable also in the case of real data. From the proposed results in Table 1 some unique peculiarities of our approach clearly emerge:

1. thanks to the use of von Mises pdf, *periodicity* (test 1) of angles is taken into account; the test contains class 3 (Fig. 2(c)) with angles close to 0°; the trajectories in this class contain a medium-high level of noise that makes often the directions go from negative to positive



(a) Class 1 (b) Class 2 (c) Class 3 (d) Class 4 (e) Class 5

(f) Class 6 (g) Class 7 (h) Class 8 (i) Class 9 (j) Class 10

(k) Class 11 (l) Class 12 (m) Class 13 (n) Class 14 (o) Class 15

(p) Class R1 (q) Class R2 (r) Class R3 (s) Class R4 (t) Class R5

Figure 2. Examples of the classes of trajectories used.

		Test 1	**Test 2**	**Test 3**	**Test 4**	**Test 5**	**Test 6**	**Test 7**	**Test 8**
Type of data		Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Synthetic	Real
Type of test		Periodicity	Noise	Mono-modal	Multi-modal	Sequence	Learn.Norm.	Mixed	Mixed
Total n. traj.	Tr	250	400	450	680	650	150	2530	520
	Te	150	250	400	430	400	200	1700	430
Average points/traj.		65	95	66	74	85	95	105	66
Training set		C_3, C_4^*	C_1, C_{15}	C_1, C_2, C_3	$C_1, C_2,$ C_5, C_6^*	$C_1, C_8,$ C_9, C_{10}^*	C_1, C_7^*	all training	$C_{R1}, C_{R2},$ C_{R3}, C_{R4}^*
Testing set		C_3, C_4^*	C_{14}^*, C_{15}	$C_1, C_2,$ C_3^*, C_4^*	$C_1, C_5,$ C_6^*, C_7^*	$C_8, C_2^*, C_9^*,$ $C_{11}^*, C_{12}^*, C_{13}^*$	C_{13}^*, C_7	all testing	$C_{R1}, C_{R2},$ C_{R4}^*, C_{R5}^*
Class. accuracy		**100%**	**100%**	**100%**	**97.67%**	**94.59%**	**95%**	**99.60%**	**96.77%**
Norm/Abn. accuracy		**100%**	**100%**	**100%**	**100%**	**97.30%**	**100%**	**100%**	**96.77%**
Class. accuracy HMM		75%	83%	94.1%	92.67%	89.1%	94.01%	86.40%	75.19%
Norm/Abn. accuracy HMM		84%	94.1%	93.02%	100%	85.1%	100%	82.4%	66%

Table 1. Summary of the experimental results.

around zero: this variance around the origin makes the model with a Gaussian unsuitable;

2. thanks to the use of a statistical model for the trajectories (Section 3.1), our approach is less sensitive to *noise* (test 2); in fact, the considered trajectories (class 1, 14 and 15) have a high level of noise (shown, for instance, in Fig. 2(n)) and have a similar main direction that can make classification more challenging;

3. thanks to the use of mixture of von Mises distributions as pdf (Section 3.2), our approach is capable to model also trajectories composed by more main directions (*multimodal trajectories*) (test 4);

4. thanks to the use of our MAP-based encoding (Section 3.3), our approach is sensitive to the *sequence* (test 5) with which the directions appear in the trajectory; please note for instance that trajectories of class 11 (abnormal) contain the same angles of those of class 8 (normal) but with different sequence; the same holds for class 12 (abnormal) and class 9 (normal);

5. thanks to the use of our dynamic updating of the clusters (Section 4), our approach can "*learn normality*" (test 6), in the sense that an initially-abnormal class of trajectory (class 7) may become normal if it is seen many times.

Finally, we compared our approach with a HMM-based classification system using the similarity measure proposed in [18] (last two rows of Table 1). The lower classification performance of HMMs is mainly due to the overfitting problem. When little data is available HMM training stage fails to correctly estimate all the parameters. The emission distribution and the optimal number of hidden state are two crucial elements that must be chosen accurately when using an HMM. This can be done in an unsupervised way but this procedure needs a large amount of data that are not always available in real scenarios. Conversely, our approach is not strongly affected by the number of data available because the number of parameters to estimate is significantly lower thus can be applied in many different situations profitably .

In conclusion, this paper proposes the use of circular statistics to analyze trajectory shapes for detecting abnormal patterns of movement. The proposed approach results very accurate also in complicated cases and outperforms standard approaches based on Gaussian pdfs and HMMs. It is remarkable that the use of von Mises distribution and the complete statistical treatment here proposed is also suitable to many other computer vision applications where angular data is involved, such as the clustering of motion vectors for moving camera analysis.

References

[1] F. Bashir, A. Khokhar, and D. Schonfeld. Object trajectory-based activity classification and recognition using hidden markov models. *IEEE Trans. on Image Processing*, 16(7):1912–1919, July 2007.

[2] F. I. Bashir, A. A. Khokhar, and D. Schonfeld. Segmented trajectory based indexing and retrieval of video data. In *Proc. of ICIP*, pages 623–626, 2003.

[3] M. Bennewitz, W. Burgard, and S. Thrun. Using EM to learn motion behaviors of persons with mobile robots. In *IEEE/RSJ International Conference on Intelligent Robots and System*, volume 1, pages 502– 507, 2002.

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

[5] D. Buzan, S. Sclaroff, and K. G. Extraction and clustering of motion trajectories in video. In *ICPR 2004*, volume 2, 2004.

[6] S. Calderara, R. Cucchiara, and A. Prati. Detection of abnormal behaviors using a mixture of von mises distributions. In *Proc. of IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS 2007)*, pages 36–41, 2007.

[7] S. Calderara, A. Prati, and R. Cucchiara. Hecol: Homography and epipolar-based consistent labeling for outdoor park surveillance". *in press on Computer Vision and Image Understanding*.

[8] G. Casella and R. Berger. *Sattistical Inference, 2nd edition*. Duxbury Press, 2002.

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

[10] Z. Fu, W. Hu, and T. Tan. Similarity based vehicle trajectory clustering and anomaly detection. In *Proc. of ICIP*, pages 602–605, 2005.

[11] D. Gusfield. *Algorithms on Strings, Trees, and Sequences*. Cambridge University Press, 1997.

[12] W. Hu, X. Xiao, Z. Fu, D. Xie, T. Tan, and S. Maybank. A system for learning statistical motion patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1450–1464, September 2006.

[13] I. Junejo, O. Javed, and M. Shah. Multi feature path modeling for video surveillance. In *Proc. of ICPR*, volume 2, pages 716– 719, Aug. 2004.

[14] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining application. In *Proc. of ACM SIGKDD International Conference on Knowledge Discovery and Data mining (KDD 2000)*, pages 285–289, 2000.

[15] J. Lou, Q. Liu, T. Tan, and W. Hu. Semantic interpretation of object activities in a surveillance system. In *Proc. of ICPR*, volume 2, 2002.

[16] K. Mardia and P. Jupp. *Directional Statistics*. Wiley, '00.

[17] A. Mecocci and M. Pannozzo. A completely autonomous system that learns anomalous movements in advanced video-surveillance applications. In *Proc. of ICIP*, volume 2, pages 586–589, Sept 2005.

[18] F. Porikli and T. Haga. Event detection by eigenvector decomposition using object and frame features. In *Proc. of CVPR Workshop*, volume 7, pages 114–121, 2004.

[19] 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.

[20] M. Sato. Fast learning of on-line EM algorithm. Technical Report TR-H-281, ATR Human Information Processing Research Laboratories.

[21] C. Stauffer and W. Grimson. Learning pattern of activity using real-time tracking. *IEEE Trans. on PAMI*, 22(8):747–757, Aug. 2000.

[22] M. Vlachos, G. Kollios, and G. D. Discovering similar multidimensional trajectories. In *Proc. of International Conference on Data Engineering (ICDE 2002)*, pages 673–685, 2002.