UNIVERSITY
OF TRENTO

DIPARTIMENTO DI INGEGNERIA E SCIENZA DELL'INFORMAZIONE

38050 Povo – Trento (Italy), Via Sommarive 14
http://www.disi.unitn.it

SYNTACTIC MATCHING OF TRAJECTORIES FOR AMBIENT INTELLIGENCE APPLICATIONS

N. Piotto, N. Conci and F.G.B. De Natale

August 2009

Technical Report # DISI-09-044

Syntactic matching of trajectories for ambient intelligence applications

N. Piotto, *Student Member, IEEE,* N. Conci, *Member, IEEE,* and F.G.B. De Natale, *Senior Member, IEEE*

Abstract

In this paper we propose a novel approach for syntactic description and maching of object trajectories in digital video, suitable for classification and recognition purposes. Trajectories are first segmented by detecting the meaningful discontinuities in time and space, and are successively expressed through an ad-hoc syntax. A suitable metric is then proposed, which allows determining the similarity among trajectories, based on the so-called inexact or approximate matching. The metric mimics the algorithms used in bio-informatics to match DNA sequences, and returns a score, which allows identifying the analogies among different trajectories on both global and local basis. The tool can therefore be adopted for the analysis, classification, and learning of motion patterns, in activity detection or behavioral understanding.

Index Terms

trajectory analysis, trajectory representation, trajectory matching, ambient intelligence, visual surveillance.

I. Introduction

The growing interest in ambient-intelligence and the significant reduction in the price of image capture devices and digital signal processing systems, has contributed to the widespread adoption of video technologies in most monitoring and surveillance applications. On the other hand, large and distributed sensing architectures provide human operators with huge amounts of data (mostly real-time video) that quickly overwhelm the ability of the security personnel to analyze and react to events, especially in safety-critical applications. As a matter of fact, most of the available consumer products mainly focus on the recording of video sequences for after-event analysis that are useful as forensic tool, but disregard the primary benefit of surveillance systems as active and real-time prevention instruments. More sophisticated systems attempt to process data in real-time in order to detect significant events that need to be promptly

reported to the operator. This is the case of systems for decision support, where the automation of certain procedures allows real-time detection of relevant events. Such events are typically related to changes detected in the monitored area, which can be caused by human actions (e.g., entering/exiting the scene, accessing some specific areas), modifications of the environmental conditions (e.g., objects relocation, objects left unattended, changes in illumination, presence of shimmering lights or fire), or suspect behaviors (e.g., identification of specific movement patterns, interaction with objects in the scene). Most of these events are associated with the presence of moving entities like people and/or other objects in the scene. Nowadays, sophisticated and reliable object trackers can be found in the literature that make it possible to extract an accurate representation of the spatio-temporal trajectory of each object in a video sequence (see [1] [2]) also in very complex scenarios. Starting from the acquired trajectory, a common way to detect activities or behaviors consists in translating the trajectories of the moving objects into sets of descriptors, and successively comparing such descriptors with predefined (or learned) models. This approach has been widely used in many application fields such as smart environments (motion is analyzed to understand people presence and behaviors [3] [4] [5]), content-based video indexing and retrieval [6], gesture and gait analysis [7], and biometry [8].

Starting from a preliminary study proposed in [9], we present in this work a complete representation and matching framework, and provide an in-depth description of the relevant processing techniques and a thorough experimental validation, also in comparison with state-of-art approaches of the same class. The paper presents some related works in Section II, while Section III concentrates on the proposed architecture. Section IV focuses on the experimental validation on two different sets of trajectories in different indoor scenarios. Concluding remarks are drawn in Section V.

II. Related work

The algorithms used to describe and compare trajectories can be divided into three main categories: dynamic matching, statistical matching, and vector matching. Within the first category, dynamic time warping (DTW) is typically used in time-series comparison, but it has been successfully applied also to human trajectory matching because of its conceptual simplicity and versatility [1]. Very simple yet effective, DTW has a major drawback in its sensitivity to time differences, which leads to unreliable results when the trajectories to be matched are sampled at different temporal rates. Furthermore, DTW demonstrates a limited robustness also to noise and outliers as well as to shifts and scaling.

Statistical matching refers to methods that jointly process a wide set of trajectories to determine their distribution in a given feature space (e.g., spatial location, moving direction, speed, etc.). High

matching scores are assigned to trajectories whose behavior fits a prototype distribution. If the statistics of a given trajectory do not fit any of the prototypes, it is classified as anomalous. Johnson et al. [10] employed a sequence of flow vectors to represent the trajectory of the tracked object. An estimation of the statistical spatial distribution of these vectors is achieved by applying a vector quantizer. In particular, two concurrent neural networks are employed: the first processing stage identifies the sequence of vectors that best represents the target trajectory; in the second stage, the clustering is performed to group similar tracks. The major drawback of this technique is that it cannot handle partial tracks. An improved version of the algorithm has been developed in [11]; here, an autonomous tool that learns anomalous movements is conceived. The authors provide a learning module similar to the one used in [10], but ensuring higher accuracy in the clustering phase, and allowing for an automatic setup of the trajectory prototypes (clusters). Each prototype is assumed to have a Gaussian distribution, and the anomaly detection is carried out by statistically checking the fitness of the incoming track over the prototype model. As anticipated above, also clustering techniques for trajectory description and matching can be included within this category of approaches. A few meaningful examples are briefly summarized hereafter. The work in [12] describes a strategy for trajectory distance measurement and clustering based on a Hidden Markov Model (HMM). The track evolution and dynamic properties are captured within a state transition matrix by a continuous chain of HMMs. Through this approach the categorization of the paths can be achieved by taking into account the speed affinity together with the geometrical/spatial features. Another benefit is represented by the capability of the system to cope with the so-called uneven sampling instances (non-uniform sampling between consecutive points), which are typical of real-time tracking applications. In [13], the goal is to achieve a hierarchical clustering strategy that first identifies global similarities (referred to as general trend) and then performs a refined analysis of each coarse cluster. In this approach, wavelet decomposition is employed to tackle the presence of noise in the raw trajectory: after smoothing, a feature extraction phase is carried out, in which the trajectory resampling point set (TRPS) and the trajectory directional histogram (TDH) are retrieved. TDH is used to identify coarse trajectory clusters, while TRPS is used to refine them. More recently, the approach in [14] was proposed, which performs better than [13] in the presence of noise. It is based on an unsupervised clustering algorithm that uses a mean-shift to detect coarse clusters, followed by a merging procedure in which adjacent blobs are grouped and outliers are detected and deleted. Also in this case, the resampling algorithm does not allow identifying the local variations in terms of time and speed. Another trajectory clustering approach can be found in [15], where trajectories are organized in a tree structure, along with the corresponding occurrences that are used to detect anomalies.

Finally, vector-based matching techniques define similarities among trajectories on the basis of the distance between the feature vectors associated to each track [16]. The matching requires the mapping of each trajectory into a set of features, and a metric (e.g., Euler, Minkovsky, Hausdorff distances) as a measure of similarity [17]. Among the most interesting techniques that employ this approach, Chen et al. [18] introduced a trajectory retrieval system using a symbolic representation called movement pattern string (MPS). MPS approximates the real trajectory according to a predefined space quantization map and specific symbols are used to characterize the motion patterns. The authors also defined a similarity metric for trajectory matching, based on the edit-distance [19]. The work in [20] introduces another interesting video retrieval system that compares video clips according to the similarity of the trajectories of moving objects in the scene. Similarly to [18], the authors propose a hybrid method to capture the semantic meaning and the geometrical characteristics of each trajectory; the comparison is then performed through string matching. Since the above methods are thought for pure video retrieval, none of them takes into consideration the temporal references in encoding and matching the symbolic trajectories, although the temporal evolution of the track may represent a critical factor to characterize the behavior of a moving object. This problem is solved only partially in the retrieval system proposed in [21], where the goal is to bridge the semantic gap between the user queries and the trajectory representation. Here, the incoming samples are filtered and spatio-temporally clustered in order to learn activity models; the acquired models are then indexed in a hierarchical tree, where each child inherits the parents properties. The trajectory query interface is provided to final users at semantic level.

A more recent implementation that exploits the edit-distance is presented in [22]. Here, the object trajectories are processed and represented by a chain of symbols indicating the direction and velocity components (sampling time is assumed unitary and constant): the symbolic mapping of the path is then achieved by quantizing each component, in order to reduce the redundancy. Since no resampling or trajectory smoothing is applied, the symbolic mapping may lead to long symbol chains where each sample is encoded as a symbol.

In our paper we propose a new paradigm based on the edit-distance [19], which does not require the resampling of the trajectory and allows taking into account the time component as a key feature to describe the object motion. A selection of experimental tests will be presented, to demonstrate the effectiveness of the method. Furthermore, a comparison with state-of-art approaches will be proposed, referring in particular to Longest Common Sub-Sequence (LCSS, [17]) and a more recent method presented in [22].

A. General overview of the system

The implementation of an effective trajectory similarity metric requires a few preliminary considerations. The extraction of object trajectories from video data is typically imprecise due to environmental noise, segmentation errors, and occlusions: these uncertainties typically produce unreliable tracks containing gaps and misplacements. Moreover, the same spatial trajectory could be associated to different duration, speed and acceleration patterns. A good representation and matching strategy should be able to catch similarities and differences in all these respects, assigning the appropriate weight to each parameter. According to these considerations, the key idea of the matching scheme proposed in this work has been inspired by the alignment procedure adopted in bio-informatics to match genomic sequences [23] [24], also referred to as inexact or approximate matching. These techniques do not provide a hard matching (i.e., point by point as in DTW), since they rely on modifications of the edit-distance [19]. The edit-distance is based on the combination of elementary operations, such as deletion, insertion and substitution, together with the assignment of specific scores to each of them. These algorithms can be applied on different scales, and in Fig. 1 an example is shown where two different matching results are obtained over the same pair of genetic sequences considering the global and local alignment, respectively. The global alignment determines the score corresponding to the matching result over the whole sequence, while the local alignment calculates the score considering the most similar subsequence.

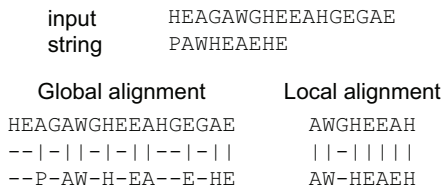


Fig. 1. Global and local alignments of a pair of DNA sequences.

Accordingly, we propose to segment the track in syntactic elements that represent significant substrings of the original trajectory which are used as basic symbols of a string representation. The structure of the symbols has been arranged according to a set of rules that ensure a flexible representation, as we will discuss in the following sections. The string-based representations are then aligned according the above strategies. An overview of the processing flow is shown in Fig. 2: raw trajectories are pre-processed to detect the spatio-temporal discontinuities, thus identifying a reduced set of meaningful trajectory segments. The concatenation of the obtained segments can thus be assumed to be an approximation of the original

trajectory. The quantization of each segment in terms of direction, velocity and time, lets mapping each level into a symbol, selected from a pre-defined codebook. Then, the matching between two trajectories can be expressed as the cost of aligning the corresponding strings of symbols. The major advantages of this representation and the matching strategy we propose, can be summarized in two main points:

- reduction in the complexity of representation and matching and capability of considering the invariance to scale, rotation, temporal or spatial shifts;
- temporal and spatial features jointly contribute to the score calculation, thus leading to a more accurate alignment, able to detect similarities on both global and local level.

Additionally, we highlight the capability of building the symbol string *on-the-fly*, thus making it possible to analyze the sequence and to evaluate the matching score in real-time, even if the complete trajectory is not available yet. The nature of the edit-distance turns out to be effective also in tackling the local noise; in fact, the best match is found when coupling close symbols and discarding the outliers, which are handled at the syntactic level. An outlier in the trajectory may generate a very brief sequence of wrong symbols (1-2) associated to gaps in the alignment process (dashed lines in Fig. 1).

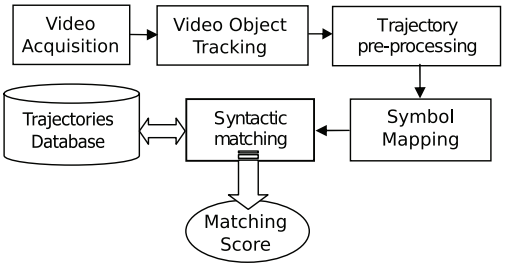


Fig. 2. Application flowchart.

III. The proposed approach

In this section we describe the proposed trajectory representation and matching algorithm. We would like to point out that video object tracking is beyond the scope of this paper. We therefore adopted a state of art methodology. The trajectories we use consist of the projection of the objects *centroid* on the floor, which represent the top-view of the object displacement in the environment. The tracking module we used is based on [25] for the background suppression stage, while the tracking algorithm uses a proximity criteria to detect adjacent blobs across frames based on their color appearance and distance. Since this would result in an inaccurate discrimination of objects in the presence of occlusions, we adopted a stereo

camera to derive the depth information, through which it is possible to project the blobs on the ground floor and merge them accordingly. Analogous results can be obtained by using multicamera systems. Fig. 3 shows an example of a moving object detection and the corresponding top-view trajectory ($x-z$ plane), where the coordinate (0,0) refers to the camera position.

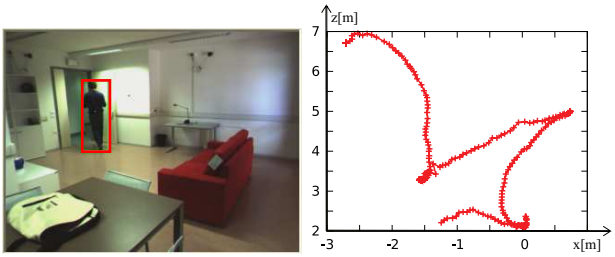


Fig. 3. Object tracking and top-view trajectory.

A. *Trajectory segmentation and characterization*

Starting from the raw trajectory extracted by the tracker, an on-line filtering is applied in order to identify the spatio-temporal discontinuities in the path (trajectory pre-processing in Fig. 2). The input of the pre-processor unit is:

$$T_i = \{x_j^i, z_j^i, t_j\};\ j = 0...N \tag{1}$$

where x_j^i and z_j^i determine the top-view position of the i-th tracked object at the time t_j as shown in Fig. 3, and N is the number of trajectory samples. To detect sharp velocity discontinuities in the object motion, and in particular stops/re-start events, the coordinates of the object $\left(x_k^i, z_k^i\right)$ are evaluated in the time window $[t_k, t_{k+l}]$. If the object position does not change within the selected time interval, $P_k^i = (x_k^i, z_k^i, t_k)$ is marked as a *temporal breakpoint*. Since the *centroid* of the object is subjected to small position variations due to noise, a guard area of radius ρ proportional to the object size, is used to check the stop condition [26]. Considering an indoor scene, the characteristics of human motion, and an acquisition rate of 25 samples per second, in our tests we set the radius ρ in the range $[0.5, 1]$ meters, and a time frame l in the range $[50, 75]$ frames (equivalent to 2-3 sec).

As far as the spatial analysis is concerned, two separate procedures are implemented (Fig. 4). The former detects sharp direction variations by analyzing a temporal window of three consecutive samples: the current point $P_k^i(x_k^i, z_k^i, t_k)$ and two previous observations $P_{k-1}^i(x_{k-1}^i, z_{k-1}^i, t_{k-1})$ and $P_{k-2}^i(z_{k-2}^i, y_{k-2}^i, t_{k-2})$.

The interpolating lines $r_{k-1}(x)$ and $r_k(x)$ are then calculated, being the lines passing through P^i_{k-2}-P^i_{k-1} and P^i_{k-1}-P^i_k:

$$r_{k-1}(x) = m_{k-1}x + q_{k-1} \tag{2}$$

$$r_k(x) = m_k x + q_k \tag{3}$$

where m_{k-1} and m_k represent the slope of the lines connecting the two points, and q_{k-1} and q_k are the corresponding offsets.

Then, the angle β_k (Fig. 4(a)), is calculated according to (4) and it is compared with a predefined threshold β_{th}: if the angle exceeds the threshold, P^i_k is marked as a *spatial breakpoint*.

$$\beta_k = tan^{-1}\left|\frac{m_{k-1} - m_k}{1 - m_{k-1}m_k}\right| \tag{4}$$

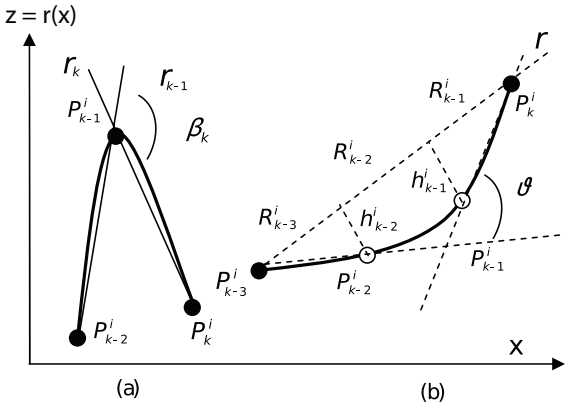


Fig. 4. (a) Local variation angle and (b) cumulative variations leading to a significant direction change.

The above criterion (derivative) cannot detect cumulative changes in direction generated by successive small variations, (Fig. 4(b)). An integrative criterion has been therefore implemented to calculate the area γ subtended by the trajectory, starting from the last breakpoint up to the current sample:

$$\gamma_{(k-g,k)} = \frac{1}{2}\sum_{q=k-g}^{k}[(h^i_q + h^i_{q+1})(\left|R^i_{q+1} - R^i_q\right|)] \tag{5}$$

In (5), h^i_q is the Euclidean distance between the current sample P^i_q and the line r that connects P^i_{k-g} (last breakpoint) with P^i_k(current sample); R^i_q is the projection of the sample P^i_q on r. Again, if the resulting area $\gamma_{(k-g,k)}$ exceeds a given threshold (γ_{th}), the sample P^i_k is marked as a spatial breakpoint (filled dots in Fig. 4(b)). The choice of the two thresholds β_{th} and γ_{th} will be analyzed in the Section

IV based on the evaluation of the accuracy of the system on a set of training sequences. As a rule of thumb, we can state that β_{th} determines the reactivity to local variations, while γ_{th} affects the sensitivity to long-term deviations. Since pedestrians tend in general to walk along smooth trajectories, the most important threshold is usually γ_{th}.

B. Key points symbolic mapping

The above described spatio-temporal analysis identifies a chain of breakpoints B_m^i (being $m = 1...M$ the number of detected breakpoints for the i-th object), along the original trajectory. Each pair of successive breakpoints identifies a rectilinear segment $S_m^i = B_m^i \leftrightarrow B_{m+1}^i$ that approximates a portion of the original path. Accordingly, the approximated trajectory can be represented (and reconstructed) by an appropriate description of the segment chain $\{S_m^i\} : m = 1 \dots M-1$. In this representation, each segment S_m^i is characterized by its orientation θ_m^i, its velocity v_m^i, and the relevant temporal interval Δt_m.

The above parameters are determined as follows: direction and duration are calculated with respect to the previous segment (6) (7), while speed is computed as the length of the segment divided by its duration (8).

$$\theta_m^i = \quad \beta_m^i \tag{6}$$

$$\Delta t_m = \quad t_m - t_{m-1} \tag{7}$$

$$v_m^i = \quad \frac{d(B_m^i, B_{m-1}^i)}{\Delta t_m} \tag{8}$$

β_m^i is calculated according to (4) in B_m^i; t_{m-1} and t_m are the absolute time references corresponding to B_{m-1}^i and B_m^i, respectively, and $d(a,b)$ is the Euclidean distance. The approximated trajectory description for the i-th object is then given by (9):

$$T_i^* = \{\theta_m^i, v_m^i, \Delta t_m\};\ m = 1...M \tag{9}$$

This representation is inherently invariant to rotation and translation and fulfills several requirements. In fact, only the coordinates and orientation of the first segment refer to an absolute positioning, then this information can be easily discarded to achieve invariance to translation and orientation. Similarly, if the temporal discontinuities of the trajectory are not relevant, stops can be removed by simply dropping samples with null speed. As it can be noticed, these features allow performing different types of matching such as, for instance, identifying trajectories with similar geometry but different speed, or detecting similar behaviors (e.g., *zig-zag* moving patterns) in different locations of the room.

TABLE I

QUANTIZATION LEVELS.

Variable	Range	Quantization Levels
θ_m	$[-180^\circ \ +180^\circ]$	$\theta_0 \ ... \ \theta_{11}$
v_m	$[0 \ v_{max}]$	$v_0 \ ... \ v_3$
Δt_m	$]0 \ \infty]$	$\tau_0 \ ... \ \tau_3$

The last step to achieve a complete syntactic representation consists in mapping each segment into symbols. This can be obtained by properly quantizing the parameters $\{\theta_m^i, v_m^i, \Delta t_m\}$, in order to make the symbols enumerable. Since the application we address in our tests is people tracking in indoor environments, owing to the limited speed and typical movements of the target, we quantized the direction θ_m in 12 non-uniform levels, while speed and time components have been quantized in 4 levels (see Table I). The choice of the values associated to each level are discussed in Section IV.

Fig. 5 shows a time-space diagram, in which the original and the reconstructed trajectories are plotted. Markers represent the breakpoints detected by the segmentation algorithm.

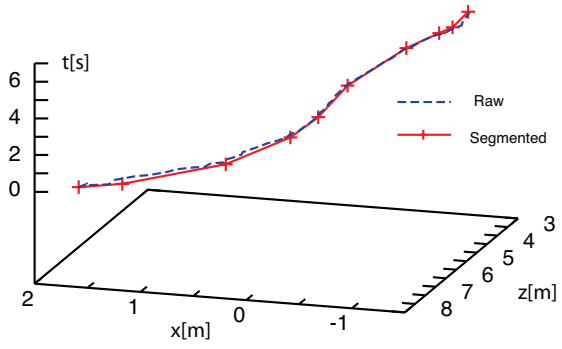


Fig. 5. Path segmentation.

C. Trajectory alignment and matching

The goal of the syntactic matching engine is to find the best alignment among strings that represent different trajectories, and to calculate the corresponding similarity score. Depending on the application, the trajectories to be matched can be part of a pre-defined database (e.g., knowledge-based behavioral analysis), queries sketched by the user (e.g., content-based video retrieval), or actions automatically

learned by the system (e.g., behavior classification and automatic detection of anomalous events). Thanks to the syntactic representation of the object paths, the proposed matching procedure is very fast and efficient, similarly to what text processors do in detecting and correcting errors.

The alignment algorithm we present in this paper relies on the so-called edit-distance. Basically, the difference between two strings of symbols is measured as the minimum-cost set of elementary actions (i.e., insertion, deletion and substitution) required to transform one sequence into the other. To achieve this goal, a cost (weight) is associated to each operation according to its relevance in the transformation. This makes it possible to assign different weights to different types of actions (e.g., a substitution can have higher impact than a deletion) and to the symbols involved in the action. The association of an operation to a weight is achieved by using a substitution matrix (i.e. a look-up-table). The total cost of the transformation is the sum of the single weights. The edit-distance has been chosen against other metrics because of its flexibility, and in particular the possibility of easily adapting the matching to different application requirements. In fact, any kind of matching rule can be implemented by properly adjusting the substitution matrix. Another significant advantage of the selected approach is the capability of operating *on-the-fly*, i.e., processing the samples as soon as they are acquired. As to the specific implementation, we introduce a modified version of the theory presented in [23] and [24]. The concept we apply is similar, in the sense that we assign a score to each symbols pair: the higher the score, the better the matching (equal symbols receive maximum score). The best global matching is the one that maximizes the global score. The identification of the most suitable substitution matrix is *application-dependent*: for instance, the entries in the matrix employed in [23] (DNA sequences alignment) are defined on the basis of the biological similarity among amino-acids (A, T, G, C). The alignment procedure adopted in this work provides the simultaneous matching in three different domains, namely space, speed, and time. Since these three parameters have different impact on the matching, the score is calculated starting from three separate substitution matrices, one for each parameter, and then adding the single scores to achieve the overall substitution cost.

As far as the substitution matrices are concerned, we provide here a simple example that can explain the rationale behind the choice of the entries. Let us assume that a trajectory is described through five symbols: move forward (f), slight turn right (st_r), slight turn left (st_l), sharp turn right (St_r), and sharp turn left (St_l). While performing the matching, it is likely that the cost of substituting f with st_r is larger than the cost of substituting f with St_r, since a slight turn is more similar to a straight path than to a sharp turn. The cost of substitution pair St_r-St_l should be clearly even larger. Time and speed parameters behave a little differently, since they have theoretically no upper bound. In this case, we assume that the

score drops down to zero, once exceeding a given distance, meaning that the two symbols are no more correlated. It has to be pointed out that the tuning of the matrices has an important semantic impact: for instance, if speed should be ignored in the matching, the corresponding matrix will have the same value for all entries, thus becoming irrelevant in the comparison phase. Section IV provides a detailed explanation of our experimental setup, together with actual examples of matrix configurations. Given the matrices, the alignment procedure can be described as follows. Let us take two generic strings of symbols, A and B, of length N_A and N_B, respectively. A two-dimensional array, commonly called F matrix, is created with dimension $N_A \times N_B$. The F matrix is iteratively filled by assigning to each entry $F(i,j)$ the score of the optimum alignment between the symbols of string A and string B, according to the following algorithm:

```
F(0,0) = 0
F(0,j) = w * j
F(i,0) = w * i

foreach (i,j)
  F(i,j) = max(F(i-1,j-1)+S(A(i),B(j)),
               F(i,j-1)+w,
               F(i-1,j)+w)
```

where $S(A(i), B(j))$ represents the function that calculates (for example) the orientation score between $A(i) = \{\theta_i\}$ and $B(j) = \{\theta_j\}$.

At the end of the process, the last entry $F(N_A, N_B)$ returns the best global alignment between the two strings. Successively, a trace-back procedure allows retrieving the sequence of elementary operations that lead to the specific alignment (i.e., the best path to go back to $F(0,0)$).

The calculation of the F matrix is the most computationally expensive phase: dynamic programming techniques have been introduced to reach a good trade-off between space and time complexities. In our solution the running time and the used memory is $O(nm)$, even though more efficient implementations can lead to significant reductions in complexity.

The adopted cost function is expressed in (10) and (11):

$$\delta_k = F_k(i,j);\ k = \theta, t, v \tag{10}$$

$$\Psi(i,j) = \frac{\alpha_\theta \delta_\theta}{Q_\theta} + \frac{\alpha_t \delta_t}{Q_t} + \frac{\alpha_v \delta_v}{Q_v}\ with\ \alpha_\theta + \alpha_t + \alpha_v = 1 \tag{11}$$

where α_θ, α_t, and α_v are the feature weighting coefficients, Q_θ, Q_t, Q_v, are normalization factors

corresponding to the maximum score associated to each feature, and δ_θ, δ_t and δ_v are global scores for each single feature, obtained by applying the substitution matrices. As the alignment algorithm proceeds, the temporary score is normalized over the whole number of elementary operations required to align the substrings. It is to be noticed that (10) and (11) provide the global alignment without considering the initial rotation and translation. In order to retrieve the absolute direction and position and use them as inputs to the alignment algorithms, two additional parameters Ψ_{pos} and Ψ_{dir} need to be considered:

$$\Psi_{pos} = \frac{T}{Q_{pos}} \tag{12}$$

$$\Psi_{dir} = \frac{R}{Q_{dir}} \tag{13}$$

$$T = F_{pos}(1,1) \tag{14}$$

$$R = F_{dir}(1,1) \tag{15}$$

Q_{pos} and Q_{dir} represent again normalization factors (the maximum score for initial direction and translation alignment, respectively), while T and R correspond to the score for initial position and direction variations, respectively. T and R are calculated using appropriate substitution matrices that specifically match the first point of the trajectory.

Finally, the final spatio-temporal matching is achieved by combining (12), (13), (14), (15) in a weighted sum, as in (16).

$$\begin{aligned} \Psi_{global}(N_a, N_b) = \psi\Psi(N_a, N_b) + \\ \psi_{pos}\Psi_{pos} + \\ \psi_{dir}\Psi_{dir} \end{aligned} \tag{16}$$

where again $\psi + \psi_{pos} + \psi_{dir} = 1$.

According to the application requirements, the roto-translation parameters can be appropriately weighted. In the specific case where $\psi_{pos} = \psi_{dir} = 0$, only the general shape of the trajectory is considered, no matter its absolute positioning and orientation.

As far as the score function $S(A, B)$ is concerned, we have imposed a non-linear distribution. The score evaluation is performed using the recursive function reported in (17) that calculates the score between two symbols $A(i)$ and $B(j)$, where i and j are the symbol quantization levels. Through this representation it is possible to fill the matrix entries for each feature. As expected, the highest values are along the main diagonal, gradually decreasing as soon as the distance between symbols increases, in a cyclic fashion.

$$S(A(i), B(j)) = S(i, j) = S(i, j-1) + |i - j| \tag{17}$$

The above weight assignment produces symmetric matrices, so that the cost of inverting a symbol pair is equivalent in the two directions.

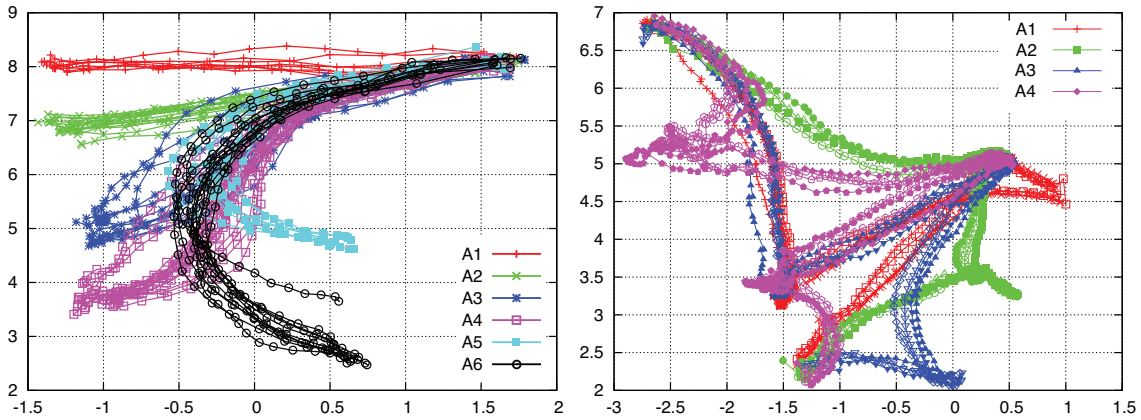
IV. Experimental results

The proposed strategy has been tested in an indoor environment, using a standard PC connected to a stereo camera. The tests concerned human activity monitoring. The acquired trajectories refer to a 2D top-view of the person motion as detected by the tracker, i.e., the location of the person with respect to the floor of the observed room. To validate the proposed method we adopted two different data sets. The first data set (*MMlab*) refers to relatively simple trajectories where the starting point is common for all sequences. It is composed by 112 tracks divided in 6 different actions and including 35 anomalous paths. The environment used for testing is shown in Fig. 6(a) and the set of trajectories is reported in Fig. 7(a). A second data set (*Application Lab*) refers to more complex trajectories acquired in an experimental smart environment, dedicated to develop technologies for assisted living. The laboratory is fully equipped with furniture, in order to simulate real moving patterns corresponding to typical activities (e.g., move from the sofa to the kitchen, bring an object and take it back to the sofa). In this case the set includes 100 trajectories grouped into 4 different clusters (see Fig. 6(b) and Fig. 7(b)) and including 42 anomalous paths. In both figures the coordinates (0,0) refer to the camera position and the coordinates of the points always refer to the $x - z$ plane, namely, the top-view of the room. The two sets of anomalous paths are shown in Fig. 8. Experiments required the proper setup of the parameters, and in particular the

(a) (b)

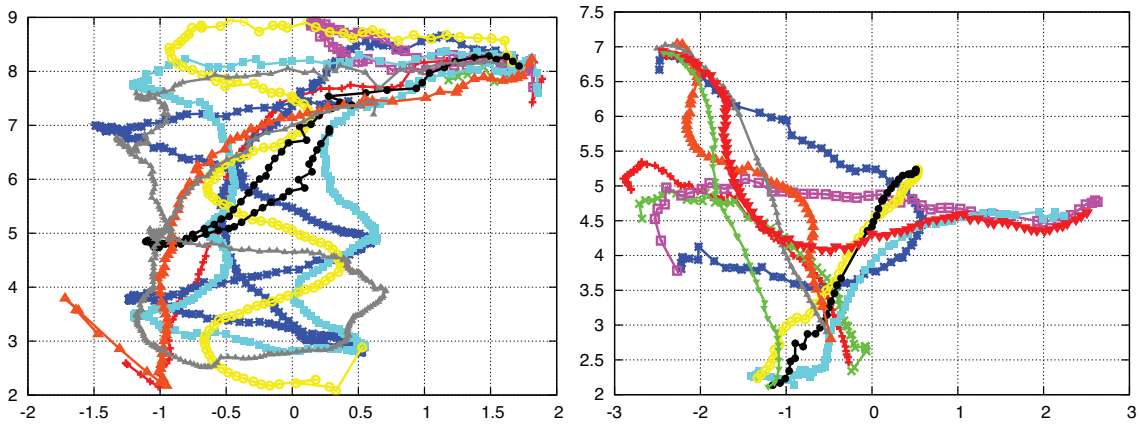
Fig. 6. Snapshots of the environments used for validation.

thresholds for trajectory segmentation. As anticipated in Section III-A, spatial and temporal thresholds are chosen in order to maximize the accuracy of the classification of activities. To this purpose, a prototype is defined for each class as the cluster medoid (i.e., the path with average minimum distance from all the others in the same cluster) and the patterns are classified according to the minimum distance prototype.



(a) (b)

Fig. 7. (a) Known actions for MMLab and (b) Application Lab data sets.



(a) (b)

Fig. 8. (a) Anomalous paths for MMLab and (b) Application Lab data sets.

In the application domain we address, it can be observed that humans tend to perform smooth trajectories instead of abrupt sharp turns within small time windows. Therefore, we have studied the performance of the segmentation scheme by varying γ_{th} for a fixed β_{th} = 30 degrees and selected the value, which returned the highest classification accuracy. To this purpose we employed the first data set. Results are plotted in Fig. 9 and report the accuracy in terms of recall and precision at different values of γ_{th}, the former corresponding to the average true positive rate, while the latter being TP/(TP+FP), where TP and FP are the numbers of true and false positives, respectively. In our scenarios, in order to discriminate among different actions, the resulting threshold is very small and it basically imposes to

fragment the trajectory in segments of around half meter. Such a fine segmentation is due to the specific geometry of the environment, which does not impose any constraint in movement (unlike for example in vehicular applications, where cars move along specific directions), and the detection of specific actions must be carried out with a finer granularity.

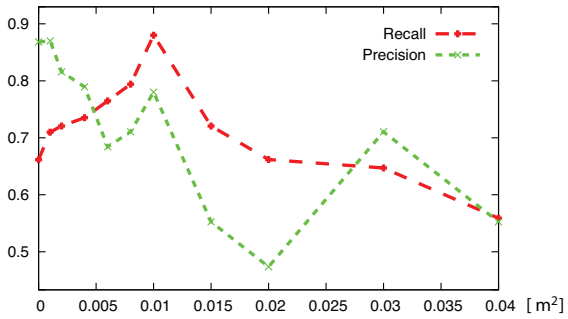
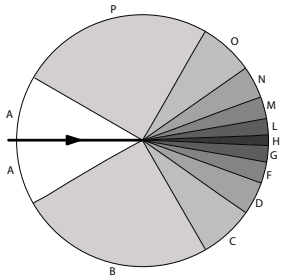


Fig. 9. Performance vs. γ_{th} variation

The correct setup of the parameters for spatial segmentation are clearly the most significant, since they refer to the geometrical displacement of the object in the scene; the choice of the temporal thresholds is instead more arbitrary. In our experiments an object is considered stopped if its *centroid* remains within the same guard area (of radius $\rho = 0.3m$), for at least 2 seconds.

To determine the symbols, and referring to the direction, we adopted a non-uniform quantization for a better description of small direction variations. This is a reasonable assumption, considering that in normal walking, sharp direction changes are more unlikely. Fig. 10 sketches the adopted quantization scheme where the bold arrow refers to the incoming direction. The deviation with respect to the outgoing sector is depicted with increasing gray levels. Each level is then associated to the corresponding symbol (see table in Fig. 10 (b)).

As far as speed is concerned, and referring to Table I, one level has been reserved for the null velocity (stop), while the last level covers the range from v_3 up to v_{max}, which specifies the maximum possible velocity. Since no maximum value can be foreseen, the level is used to discriminate velocities that exceed 5 km/h. A similar approach is applied to time, where the maximum level is set for temporal intervals exceeding the stop threshold (2 seconds in our case). The selected symbols for speed and time are {Q, R, S, T} and {W, X, Y, Z}, respectively. According to the function $S(\cdot)$ of (17), the resulting substitution matrices are shown in Fig. 11. Again, different scenarios such as vehicular applications would require appropriate settings (for instance much higher velocity threshold). It is to be pointed out, however, that



range [degree]	Associated symbol
150: -150	A
-150: -60	B
-60 : -35	C
-35 : -20	D
-20 : -10	F
-10 : -2.5	G
-2.5 : 2.5	H
2.5 : 10	L
10 : 20	M
20 : 35	N
35 : 60	O
60 : 150	P

Fig. 10. (a) Quantization levels for direction and (b) the corresponding symbols.

the selection of parameters used for human targets proved to be very robust across different tests in different environmental situations as shown in the following results.

	A	B	C	D	F	G	H	L	M	N	O	P
A	12,	11,	9,	5,	0,	0,	0,	0,	0,	5,	9,	11
B	11,	12,	11,	9,	5,	0,	0,	0,	0,	0,	5,	9
C	9,	11,	12,	11,	9,	5,	0,	0,	0,	0,	0,	5
D	5,	9,	11,	12,	11,	9,	5,	0,	0,	0,	0,	0
F	0,	5,	9,	11,	12,	11,	9,	5,	0,	0,	0,	0
G	0,	0,	5,	9,	11,	12,	11,	9,	5,	0,	0,	0
H	0,	0,	0,	5,	9,	11,	12,	11,	9,	5,	0,	0
L	0,	0,	0,	0,	5,	9,	11,	12,	11,	9,	5,	0
M	0,	0,	0,	0,	0,	5,	9,	11,	12,	11,	9,	5
N	5,	0,	0,	0,	0,	0,	5,	9,	11,	12,	11,	9
O	9,	5,	0,	0,	0,	0,	0,	5,	9,	11,	12,	11
P	11,	9,	5,	0,	0,	0,	0,	0,	5,	9,	11,	12

(a)

	Q	R	S	T
Q	12,	7,	2,	0
R	7,	12,	7,	2
S	2,	7,	12,	7
T	0,	2,	7,	12

(b)

	W	X	Y	Z
W	12,	7,	2,	0
X	7,	12,	7,	2
Y	2,	7,	12,	7
Z	0,	2,	7,	12

(c)

Fig. 11. (a) Direction, (b) speed, and (c) time substitution matrices.

We present hereafter a selection of results to demonstrate the capability of the system to identify similarities among trajectories, with different spatio-temporal configurations enabling/disabling the invariance to rotation and translation. In the first set of tests we compare two paths that are: (i) similar in space but denoting minor differences in time (T1, Fig. 12-a); (ii) similar in space but with remarkable differences in time (T2, Fig. 12-b); (iii) significantly different in both space and time (T3, Fig. 12-c). In particular, comparing T1 and T2 it is possible to notice that the execution of the same path at different speed results in a compression of the graph in the time axis.

The results obtained by applying equations (10) and (11) are reported in Table II in terms of normalized score in space, space-speed and full space-temporal domains, respectively. The scores are obtained by setting different weights in (11) as shown in Table III. The final values are normalized with respect to the total number of elementary operations required to transform one symbolic string into the other.

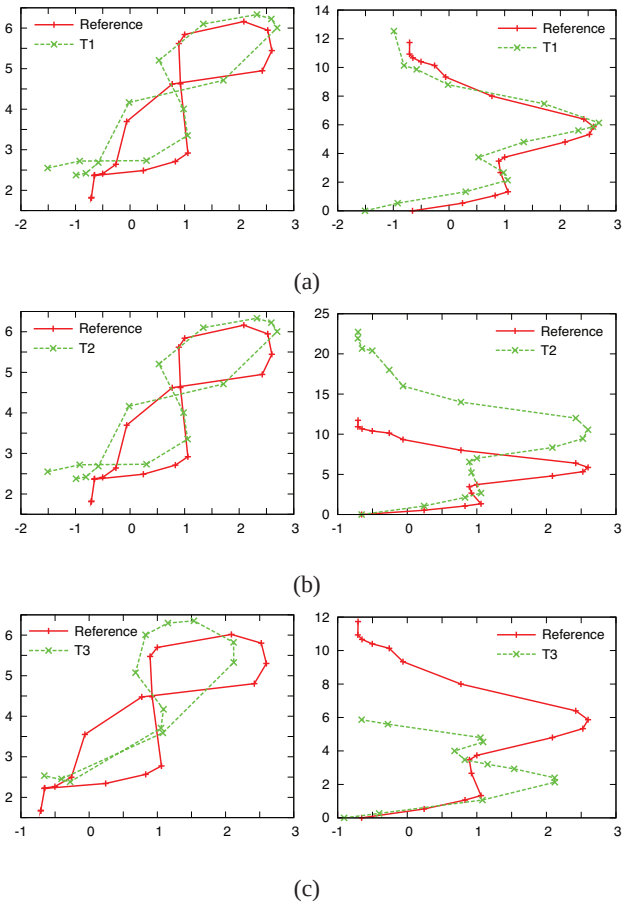


Fig. 12. Sample trajectories with different space (left) and time (right) characteristics: (a) similar paths in both spatial and temporal domain; (b) remarkable differences in time; (c) remarkable differences in space and time.

TABLE II

MATCHING SCORES IN DIFFERENT SPATIO-TEMPORAL CONFIGURATIONS.

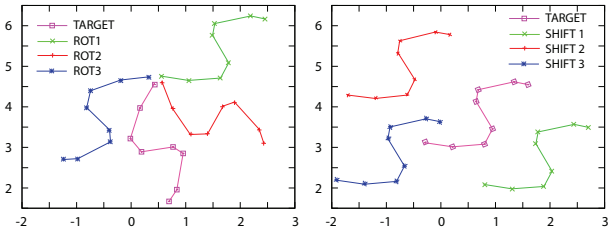
Trajectories	Space	Space-Speed	Spatio-Temp.
Reference vs T1	0.88	0.76	0.75
Reference vs T2	0.88	0.57	0.54
Reference vs T3	0.55	0.36	0.4

TABLE III

WEIGHTS ASSOCIATED TO DIFFERENT MATCHING SCHEMES.

	α_θ	α_v	α_t
Space	1	0	0
Space-speed	0.5	0.5	0
Spatio-temp.	0.33	0.33	0.33

The second set of tests aims at demonstrating the effectiveness of the described method while considering global rotation and/or translation. To this purpose, a random path (purple) has been selected as reference and compared with equally shaped paths that differ in: (i) initial direction (Fig. 13-a), and (ii) initial position (Fig. 13-b). Numerical scores are reported in Table IV and Table V.



(a) (b)

Fig. 13. (a) Copies of same paths with different initial directions, and (b) different locations.

TABLE IV

MATCHING SCORES FOR ROTATION INVARIANCE.

	Ψ_{dir}	Ψ_{global}
TARGET vs ROT1	0.91	0.96
TARGET vs ROT2	0	0.5
TARGET vs ROT3	0.41	0

In this last part of the section, we present the results achieved by processing the two data sets with the objective of detecting anomalies. The setup of the experiments reflects the same parameters configuration adopted to derive the best segmentation thresholds, as explained in the first part of this section. In this case, after finding the best match with the available classes, the matching score is evaluated. If it exceeds

TABLE V

MATCHING SCORES FOR TRANSLATION INVARIANCE.

	Ψ_{pos}	Ψ_{global}
TARGET vs SHIFT1	0.2	0.6
TARGET vs SHIFT2	0.2	0.6
TARGET vs SHIFT3	0.8	0.9

a given threshold (set to 70% in our tests) the trajectory is assumed to belong to the cluster; otherwise the path is tagged as anomalous.

We compared the performances of our method with two different state-of-art algorithms. The first is the one in [17] and refers to a common metric for sequence comparisons using the Longest Common Sub-Sequence. The second method is the one in [22], due to the fact that it shares some common principles with our work. It is to be noted that our technique does not require a uniform sampling of the points as required by the other two methods. Furthermore, it exploits the temporal information as a critical data for trajectory segmentation and matching.

TABLE VI

PERFORMANCE COMPARISONS FOR PATHS IN THE MMLAB DATA SET.

	method in [17]	method in [22]	proposed method
Recall	0.97	0.92	1
Precision	0.69	0.72	0.78
Accuracy	0.79	0.82	0.87

TABLE VII

PERFORMANCE COMPARISONS FOR PATHS IN THE APPLICATION LAB DATA SET.

	method in [17]	method in [22]	proposed method
Recall	0.94	0.91	0.97
Precision	0.67	0.69	0.83
Accuracy	0.75	0.76	0.88

In Table VI and Table VII we report the numerical results obtained from the two data sets and applying the three methods. Again, the evaluation parameters reported are Recall and Precision. Additionally, being the anomaly detection a binary classification problem, we show also the Accuracy, defined as (TP + TN)/(TP + TN + FP + FN). As it can be observed, the proposed method performs in general better than the competitors. In particular, the improvements in terms of accuracy are of 8% and 5% with respect to [17] and [22] in the *MMLab* data set. For more complex trajectories (*Application Lab*), the improvements are more consistent: 13% and 12%.

V. Conclusions

In this paper we presented a new approach to perform syntactic matching of trajectories, as a basis for applications such as activity detection, event analysis, or content-based video retrieval. Starting from the acquisition of the path in the $x - z$ plane, the meaningful spatio-temporal discontinuities are identified. Trajectory segments are then quantized and converted into symbols corresponding to the variations in terms of direction, speed, and time, with respect to the previous sample. The resulting syntax is used to compare different trajectories, adopting a bio-inspired approximate matching algorithm based on the so-called *edit-distance*. Experimental validation concerned the analysis of human walk patterns in indoor environments. Results confirm the good performance of the method in dealing with different data sets, and its flexibility in managing the invariance to translation and rotation. Moreover, the comparison with state of art approaches of the same class showed a significant performance enhancement.

References

[1] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. on System, Man and Cybernetics, Part C,*, vol. 34, no. 3, pp. 334–352, 2004.

[2] A. Yilmaz, O. Javed, and M. Shah, "Object tracking: a survey," *ACM Journal of Computing Surveys*, vol. 38, no. 4, pp. 13.1–13.45, 2006.

[3] G.L. Foresti, "Object recognition and tracking for remote video surveillance," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 9, pp. 1045–1062, 1999.

[4] Q. Wang, W. Shin, X. Liu, Z. Zeng, C. Oh, B.K. AlShebli, M. Caccamo, C.A. Gunter, E. Gunter, J. Hou, K. Karahalios, and L. Sha, "I-living: An open system architecture for assisted living," in *IEEE Int'l Conf. on Systems, Man and Cybernetics*, 2006, vol. 5, p. 4268 4275.

[5] F.G.B. De Natale, A. Katsaggelos, O. Mayora, and Y. Wu eds., "Special issue on signal processing technologies for ambient intelligence in home-care applications," *EURASIP Journal of Advances in Signal Processing*, 2007.

[6] J.W. Hsieh, S.L. Yu, and Y.S. Chen, "Motion-based video retrieval by trajectory matching," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 16, no. 3, pp. 396–409, 2006.

[7] J.P. Wachs, H. Stern, and Y. Edan, "Cluster labeling and parameter estimation for the automated setup of a hand-gesture recognition system," in *IEEE Int'l Conf. on Systems, Man and Cybernetics*, 2005, vol. 35, pp. 932–944.

[8] L. Wang, H. Ning, T. Tan, and W. Hu, "Fusion of static and dynamic body biometrics for gait recognition," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 14, pp. 149–158, 2004.

[9] N. Piotto, N. Conci, and F. G. B. DeNatale, "Syntactic matching of pedestrian trajectories for behavioral analysis," in *IEEE Multimedia Signal Processing, MMSP*, 2008.

[10] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," *Image and vision computing*, vol. 14, pp. 609–615, 1996.

[11] A. Mecocci and M. Pannozzo, "A completely autonomous system that learns anomalous movements in advanced videosurveillance applications," *IEEE Int'l Conf. on Image Processing, ICIP.*, vol. 2, pp. II–586–9, 2005.

[12] F. M. Porikly, "Trajectory distance metric using hidden markov model based representation," *European Conf. on Computer Vision*, 2004.

[13] X. Li, W. Hu, and W. Hu, "A coarse-to-fine strategy for vehicle motion trajectory clustering," *Int'l Conf. on Pattern Recognition, ICPR.*, vol. 1, pp. 591–594, 2006.

[14] N. Anjum and A. Cavallaro, "Unsupervised fuzzy clustering for trajectory analysis," *IEEE Int'l Conf on Image Processing, ICIP.*, vol. 3, pp. III –213–III –216, 16 2007-Oct. 19 2007.

[15] C. Piciarelli, G.L. Foresti, and L. Snidaro, "Trajectory clustering and its application for video surveillance," in *IEEE Conf. on Advanced Video and Signal Based Surveillance*, 2005, pp. 40–45.

[16] N. Imran, O. Javed, and M. Shah, "Multi feature path modeling for video surveillance," in *Proc. of the 17th Int'l Conf. on Pattern Recognition*, 2004, pp. 716–719.

[17] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," *Int'l Conf. on Data Engineering.*, pp. 673–684, 2002.

[18] L. Chen, M. T. Özsu, and V. Oria, "Symbolic representation and retrieval of moving object trajectories," in *Proc. of ACM SIGMM int'l workshop on Multimedia Information Retrieval*. 2004, pp. 227–234, ACM.

[19] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," Tech. Rep. 8, 1966.

[20] J. B. Zheng, D.D. Feng, and R.C. Zhao, "Trajectory matching and classification of video moving objects," *IEEE Multimedia Signal Processing, MMSP*, pp. 1–4, 2005.

[21] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank, "Semantic-based surveillance video retrieval," *IEEE Trans. on Image Processing*, vol. 16, no. 4, pp. 1168–1181, April 2007.

[22] S. Calderara, R. Cucchiara, and A. Prati, "A dynamic programming technique for classifying trajectories," *IEEE Conf. on Image Analysis and Processing, ICIAP.*, pp. 137–142, 2007.

[23] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970.

[24] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, vol. 147, no. 1, pp. 195–197, 1981.

[25] R. Cucchiara, C. Grana, G. Neri, M. Piccardi, and A. Prati, "The sakbot system for moving object detection and tracking," *10th ACM Int'l Conf. on Multimedia*, pp. 223–226, 2002.

[26] N. Brandle, D. Bauer, and S. Seer, "Track-based finding of stopping pedestrians - a practical approach for analyzing a public infrastructure," *IEEE Intelligent Transportation Systems Conference, 2006. ITSC '06.*, pp. 115–120, 2006.

Nicola Piotto (S08) received the BS and MS in Telecommunications Engineering from University of Trento, Italy, in 2004 and 2007, respectively. In 2005 and 2006 he joined different collaborations within the Department of Engineering and Computer Science (DISI) where he is attending the Ph.D. school in Telecommunications. His interests include image and multimedia signal processing, with particular attention to computer vision applications for video surveillance, monitoring and activity analysis.

Nicola Conci (S'04-M'08) received the BS and MS in Telecommunication Engineering from the University of Trento, Italy, in 2002 and 2004 respectively. From the same University he received the Ph.D in 2007. In 2007 he was visiting student at the Image Processing Lab. at University of California, Santa Barbara. Since 2008 he has been a post-doc researcher in the Multimedia and Vision research group at Queen Mary, University of London (UK). His research interests are related to video analysis and computer vision for video surveillance applications and behavioral understanding. In 2006 he received the Best Student Paper Award at the international ACM conference Mobimedia held in Alghero (Italy).

Francesco G.B. De Natale (M'96-SM'03), M.Sc. in Electronic Engineering, 1990, Ph.D. in Telecommunications, 1994, University of Genoa, Genoa, Italy. He is a Professor of Telecommunications and Head of the Department of Information Engineering and Computer Science (DISI) at the University of Trento, Italy. He is also responsible for the Multimedia Signal Processing and Understanding Lab. His research interests include multimedia signal processing, analysis, and transmission, with particular attention to image and video data. Dr. De Natale was General Co-Chair of the Packet Video Workshop in 2000, Technical Program Co-Chair of the IEEE International Conference on Image Processing (ICIP) in 2005 and the IEEE International Conference on Multimedia Services Access Networks (MSAN, now Mobimedia) in 2005. In 1998, he was co-recipient of the IEEE Chester-Sall Best Paper Award.