Counting Pedestrians in Video Sequences Using Trajectory Clustering

Gianluca Antonini and Jean Philippe Thiran, *Senior Member*

***Abstract*—In this paper, we propose the use of lustering methods for automatic counting of pedestrians in video sequences. As input, we consider the output of those detection/tracking systems that overestimate the number of targets. Clustering techniques are applied to the resulting trajectories in order to reduce the bias between the number of tracks and the real number of targets. The main hypothesis is that those trajectories belonging to the same human body are more similar than trajectories belonging to different individuals. Several data representations and different distance/similarity measures are proposed and compared, under a common hierarchical clustering framework, and both quantitative and qualitative results are presented.**

I. Introduction

TARGET detection and tracking are two related and well known problems in computer vision and image processing. Despite the multitude of methods presented in literature ([1]–[9] among others), the problem of automatic counting of targets is far to be solved. The problem is actually twofold. First, the target detection step represents by itself a very hard task, especially in complex and real scenarios. Sophisticated segmentation-based approaches are not yet fully reliable, especially when applied in bad illumination conditions and cluttered background. Second, the tracking over time of the detected targets introduces new complexity to the problem, leading to a possible errors' propagation. For the target selection step, an elegant approach has been proposed in [10], in the specific case of pedestrians. Here the detection step provides a set of *hypothetical* pedestrians, given by a subsampled version of the foreground region, obtained by background subtraction. The set of hypothetical targets are then tracked by correlation, giving rise to a set of hypothetical trajectories. A mathematical model for pedestrian walking behavior [11], [12], based on discrete choice analysis and calibrated on real pedestrian data, is used to filter the resulting trajectories, keeping only the most *human-like*. The main advantage of such a methodology is the fact that a target detection/recognition step is bypassed, reducing the complexity of the system, with a consistent gain in computational time. On the other hand, the price to pay as a consequence for the simple initialization procedure is the overestimation of the number of targets. It occurs when more points of the subsampled foreground belong to the same human body, giving rise to multiple trajectories for the same target. Other methods for detection and tracking presented in literature can generate the target over-estimation problem [13]. It can be the case of background subtraction based algorithms, which tend to split the objects into more parts; feature based tracking methods which normally assume a feature grouping step in order to make object hypotheses [14]; or motion segmentation based methods, where motion clusters are combined to generate object hypotheses [15].

In this work, we deal with the overestimation problem and we propose a comparative study between different approaches, based on clustering techniques, in order to provide a "bootstrap" method, to reduce the bias between the number of trackers and the real number of individuals present in the scene.

The paper is structured as follows. Section II provides a qualitative description of the problem while Section III contains a short literature review on clustering techniques. In Section IV the general framework is presented while the different methods we have compared are described in Sections V–VII. We report the results in Section VIII and we conclude with final remarks in Section IX.

II. Problem Definition

In the context of pedestrian tracking, the overestimation of targets gives rise to the generation of multiple trajectories, related to the same individual. The overestimation problem is related, but not identical, to the false positive problem. Namely, a false positive is informally defined as a tracker placed on an image region that does not correspond to a target of interest, being it a background region or another object we are not interested in. On the contrary, a target overestimation occurs when a target of interest is subject to a multiple detection, giving rise to multiple trackers, all of them being *correct*. In Fig. 1, we show a situation where multiple trackers are manually placed on three different individuals walking together, and manually tracked for a certain number of frames. These trackers are place on the head, the center of the body and on the feet, respectively. Informally speaking, in Fig. 1(a) it is hard to distinguish that the resulting nine trajectories belong to three different individuals. Fig. 1(b) gives a different viewpoint on the data, after adding the time dimension and having rotated the axes of the reference system.

We finally illustrate in Fig. 2 how the same trajectory dataset looks like, after that a combination of linear transformations [i.e., independent component analysis (ICA) plus rotation] has been applied.

We note that now it is easier to recognize that the nine trajectories belong to three well-defined clusters, corresponding to the original three individuals. This reasoning allows us to give a general formulation of the problem as an optimization problem.

Definition 2.1: Given a trajectory dataset $T = \{(x_i, y_i, t_i)\} \subseteq R^3$ generated by a tracking

Manuscript received October 4, 2005; revised March 14, 2006. This paper was recommended by Associate Editor E. Izquierdo.

The authors are with the Signal Processing Institute, Swiss Federal Institute of Technology Lausanne, CH-1015 Lausanne, Switzerland (e-mail: Gianluca.Antonini@epfl.ch; JP.Thiran@epfl.ch).

Digital Object Identifier 10.1109/TCSVT.2006.879118

1051-8215/$20.00 © 2006 IEEE

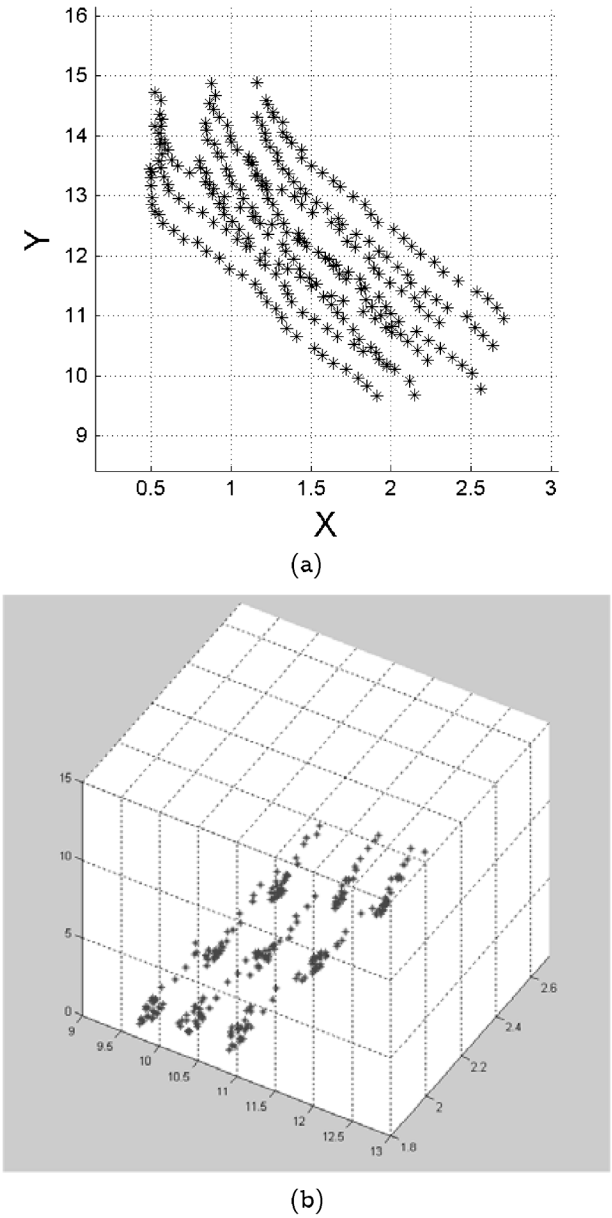


Fig. 1. Example of overestimated trajectories. (a) 2-D representation. (b) 3-D representation.

system, with $i = 1, \ldots, N$ where N is the total number of observations, and given a clustering algorithm r between the trajectories and an objective function J_r measuring the performances of the clustering algorithm, we are interested in finding the mapping $M : T \rightarrow T^{'} \subseteq R^3$ maximizing J_r.

The idea behind this general formulation of the problem is that counting targets from trajectories can be actually seen as providing a set of suitable (and general) transformations on the original trajectory dataset. Moreover, for a given association rule between the data, we are guaranteed to have the maximum discriminant power in the data association process. Of course, such a general formulation is intractable and it represents more a qualitative description of an intuitive process than a mathematical definition. Several simplifying assumptions have to be made in order to make the problem operational.

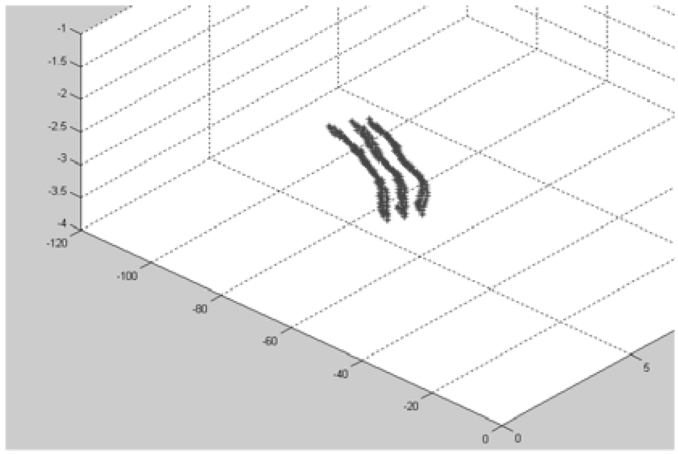


Fig. 2. Same dataset as in Fig. 1 after the application of a combination of linear transformations.

III. Clustering Literature Review

A natural way to scale down the complexity of the general optimization problem proposed above is to keep the main idea that those trajectories, originating from trackers that belong to the same target, are similar to each other. Moreover, the set of suitable transformations on the data is reduced to a specific and well founded set of different data representations. This intuition leads to a reformulation of the problem in terms of a pure trajectory clustering problem. Many research efforts have been made in this domain during the last three decades. A huge amount of literature exists on this subject and a lot of different methods have been defined. While a full review of the problem is clearly out of the scope of this paper, we focus the attention on the main aspects of this data analysis methodology. From a general point of view, any clustering method is based on three main steps: data representation, distance/similarity measures between patterns and the choice of a grouping rule.

A. Data Representation

All the feature selection/extraction methods belong to the set of transformations that map the original dataset into a more suitable representation. Using a time series representation for the data, we could be interested in removing the offset or the trend, rescale or again look for recursiveness of patterns in time [16], [17]. Linear statistical generative models can be used [e.g., principal component analysis (PCA) or ICA]. In these approaches, a new basis is found which better represents the statistical properties of the data [18]–[20] and allows for dimensionality reduction. If linear models do not adapt well to the data at hand, nonlinear dimensionality reduction techniques such as local linear embedding [21] and ISOMAP [22], [23] can be applied. There are no general guidelines suggesting methods to obtain a good data representation. The experience of the analyst and the data at hand represent the main sources of information.

B. Distance/Similarity Measures

Clustering approaches are based on fuzzy concepts, such as *nearness* or *relatedness*. To quantify these ideas, the choice of a distance and/or a similarity measure between patterns is necessary. Besides the most popular distance measures, such as the

Minkowski and Mahalanobis metrics, interesting approaches are those proposed in [24] and [25], where similarity measures and metrics are defined based on the definition of specific relations between sets of points. An approach widely used in time series analysis is the dynamic time warping (DTW) [26], [27]. The main idea behind DTW is to find an alignment of two time series on a common time axis. A lot of work has been performed in the data mining community, mainly focusing on finding better distance measures to indexing items in databases [28], [29]. Recently, several researchers have used the Hausdorff distance in a point set matching context [24], [30] while in the database retrieval domain an interesting similarity measure is the longest common subsequence (LCSS) [31].

C. The Grouping Rule

A *class* is defined as a source of patterns whose distribution in the feature space is governed by a probability density, specific to the class. Clustering techniques group patterns in such a way that classes thereby obtained reflect the different pattern generation process. Hard clustering approaches [32]–[35] assign a class label l_i to each pattern x_i, identifying its class. The set of the labels for a pattern set S is $L = (l_1, \ldots, l_n)$ with $l_i = 1, \ldots, k$ where k is the number of clusters. Fuzzy clustering procedures assign to each input pattern x_i a fractional degree of membership f_{ij} to each output cluster j [36]–[38]. Hierarchical clustering approaches produce a nested series of partitions based on a criterion for merging or splitting clusters. Such methods are more suitable in those cases where no *prior* knowledge provides information on the number of clusters. The first family of these algorithms, *agglomerative*, begins with each pattern in a distinct (singleton) cluster and successively merges clusters together until a stopping criterion is satisfied [39], [40]. The second, *divisive*, begins with all patterns in a single cluster and performs splitting until a stopping criterion is riched [41]. Partitional clustering algorithms divide data in a certain number of groups optimizing a clustering criterion [34], [42], [43]. The choice of the number of groups is made based on the *a priori* knowledge on the data at hand. Additional techniques for the grouping operation include probabilistic methods where the underlying assumption is that the patterns to be clustered are drawn from one of several distributions. The goal is to identify the parameters of each of such distributions. Most of the work has been done assuming a maximum likelihood estimation for mixture of Gaussians distributions [35], [44].

IV. Proposed Methods

A. Multilayer Structure

The idea for a multilayer hierarchical clustering arises from the consideration that the comparison between trajectories can be performed from different viewpoints. Trajectories of different lengths rarely belong to the same person. Moreover, paths belonging to the same target likely start from close spatial points. These assumptions are justified on the base of specific experimental conditions, described later on in the paper. The same assumptions result in a limitation when the focus is on the tracking problem, representing here the data collection tool. In this context, we assume a pure data analysis perspective, where we try to describe the available dataset making hypotheses based on the real phenomenon that generated them. Errors generated during the automatic tracking step are not analyzed here. In Fig. 3, we illustrate the conceptual tree structure of our clustering method.

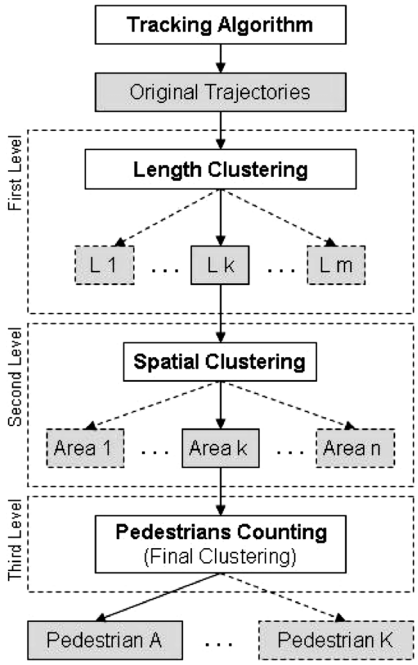


Fig. 3. Overview of the proposed multilayer clustering.

First level: Represents a length-based clustering, where trajectories having the similar length are grouped together. It is actually possible that pedestrians stay in the scene for different amount of time, yielding to trajectories of different lengths **Second level**: Individuals enter the scene at different spatial locations. A preclustering on the trajectory starting positions is performed, in order to separate different groups. We assume here that those trajectories belonging to the same pedestrian and/or to close individuals walking together start at close spatial positions. **Third level**: While the first two levels represent simple preprocessing operations on the original dataset, the actual counting task is performed at the third level. The discrimination between oversampled pedestrians and different individuals walking close to each other requires a more detailed analysis.

We approach the problem by comparing and testing different data representations and distance/similarity measures, under a common hierarchical clustering framework.

B. Compared Approaches

In Section III, the clustering problem is identified with the choice of a data representation, a distance/similarity measure and a grouping rule. In Table I, we report the different techniques that we combine and test.

- *Clustering with ICA and time series representations:* The aim here is to compare the two representations using both the Hausdorff distance and the LCSS similarity measures [45]. ICA is a generative statistical model, indicated for clustering analysis on sparse data. It reduces the influence

TABLE I
SET OF DIFFERENT DATA REPRESENTATIONS AND DISTANCE/SIMILARITY MEASURES THAT HAVE BEEN COMBINED AND TESTED, UNDER A COMMON HIERARCHICAL AGGLOMERATIVE CLUSTERING FRAMEWORK

Data represent.	Dist./simil. measure	Grouping rule
ICA	Euclidean	Hierarc. agglomer.
TS	Longest Common SubSequence (LCSS)	
MCC	Hausdorff	
ICA = Independent Component Analysis TS = Time series MCC = Maximum of cross-correlation		

of outliers, grouping the data around the independent components. The goal is to show that a distance measure sensible to the presence of outliers (the Hausdorff distance) performs well if used with a suitable representation. The time series representation does not reduce the presence of outliers, requiring a more complex similarity measure, such as the LCSS. The different combinations are tested on two different datasets and the results are reported in Section VIII-B.

- *Clustering with the MCC representation* A new representation is proposed, based on the cross-correlation between pairs of trajectories ([46]). The idea is that two identical trajectories are equally distant from a reference one. Mapping pairs of trajectories with their maximum correlation value allows to reduce the dimensionality of the data to a set of three-dimensional (3-D) points, where spatially close points represent trajectories which are similar to a reference one. We use the Euclidean distance with the maximum of cross correlation (MCC) representation, testing the method on two datasets. The relative results are reported in Section VIII-C.

V. DATA REPRESENTATIONS

Time Series: A trajectory dataset in its original representation can be considered as a time series of 2-D spatial points. Each point is represented by a *triplet* (x, y, t), the two plane coordinates (x, y) and the time step t. We have used two common preprocessing techniques with this data representation. The first one is the *linear trend* removal. The trend in a time series represents the mean slope and can be computed with standard techniques, such as linear/nonlinear regression. Intuitively speaking, removing the trend can be considered as a way to highlight fluctuations around the mean slope. The advantage of trend removal is that slight nonstationarities can be (partially) addressed. In the case of pedestrians walking in normal (no panic) conditions, we can expect *a priori* a certain degree of regularity and highly non linear time series should be unlikely (see [47] and [48]). As a consequence, we use a linear regression model to estimate and remove the linear trend. Another family of techniques widely used working with time series is represented by *smoothing* algorithms. These techniques are used to remove irregularities in the data and provide a clearer view of the underlying behavior of the series. When the trend has been removed a *single smoothing* algorithm can be used

$$f_t = \alpha y_{t-1} + (1-\alpha) f_{t-1} \tag{1}$$

with $0 < \alpha \leq 1$ and $t \geq 3$. The α parameter is called the smoothing constant, f_t is the smoothed value and y_t the original value of the series at time t. We can also perform smoothing accounting for the trend at the same time, using *double exponential smoothing*. The equations describing the model are

$$f_t = \alpha y_t + (1-\alpha)(f_{t-1} + b_{t-1}) \tag{2}$$

$$b_t = \gamma(f_t - f_{t-1}) + (1-\gamma) b_{t-1} \tag{3}$$

where the same notation as before has been used and b_t represents the trend at time t. The first smoothing equation adjusts f_t directly for the trend of the previous period b_{t-1}. The second smoothing equation then updates the trend, which is expressed as the difference between the last two values. The equation is similar to the basic form of single smoothing, but here applied to the updating of the trend. α and γ are the two smoothing constants, used to smooth the observation and the trend, respectively. They are are bounded in the interval $[0, 1]$. We report in Fig. 4 an example of the effect of such a preprocessing techniques on the same manually tracked trajectory dataset shown in Fig. 1(a).

Independent Component Analysis: The main idea here is to consider trajectories as sequences of 3-D points, (x, y, t), generated by a stochastic process. Walking pedestrians give rise to trajectories which are well different one from the other. Even if two persons follow the same spatial path, they do that at different times, leading the two trajectories to be separated when using a 3-D representation. This fact leads the trajectory dataset to be sparse. These heuristics find a natural mathematical formalization in probabilistic generative models, which are well known in literature, widely used in almost any scientific domain involving statistical computation and analysis. ICA [18], [19] in particular is a generative model where a set of random variables, the *observations*, are supposed to be generated by a mixing process, starting from another set of statistical independent latent (unobservable) variables, the *sources*, by means of an unknown mixing matrix A. This model can be described by the following equation:

$$\mathbf{X} = A\mathbf{s} \tag{4}$$

where X represents the observations, s the sources, and A is the mixing matrix. The basic hypothesis of the ICA model is the statistical independence of the latent variables. This property can be derived using an information-theoretic framework. We define the mutual information I between m scalar random variables $y_i, i = 1, \ldots, m$ as follows:

$$I(y_1, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}) \tag{5}$$

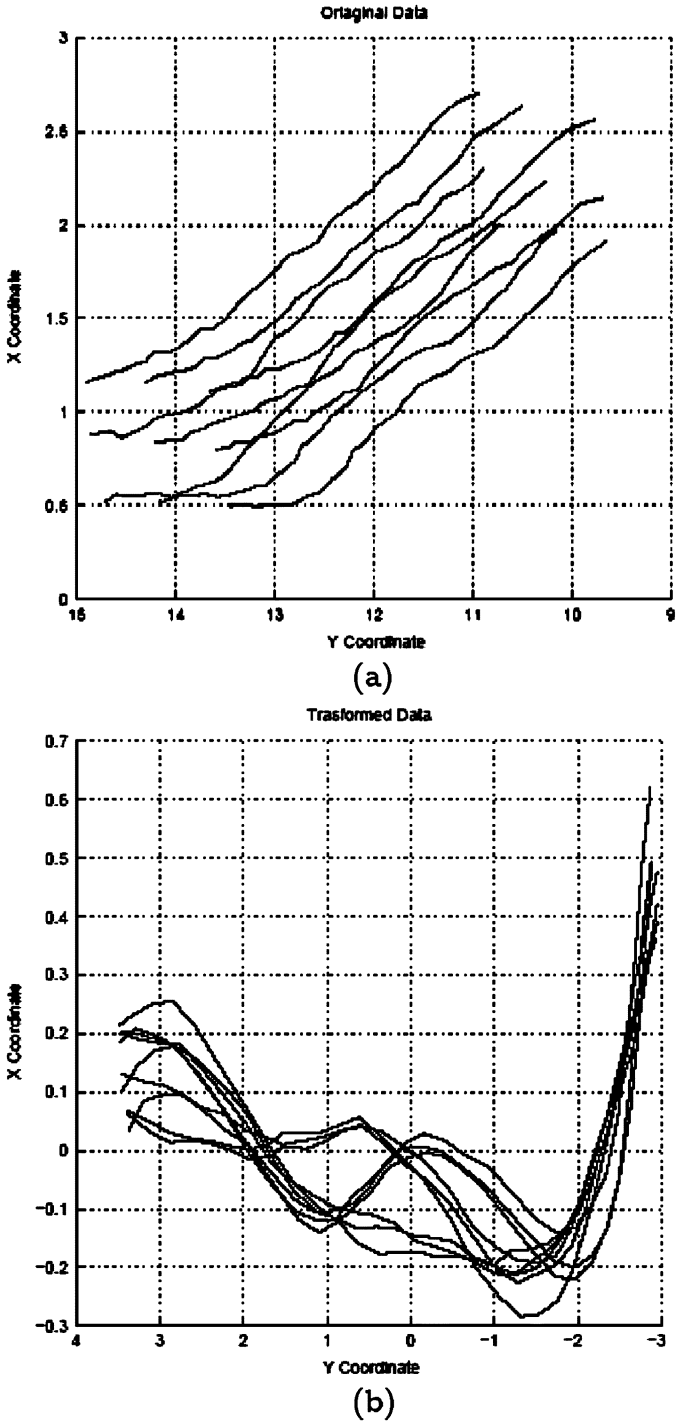


Fig. 4. Nine trajectories are generated by three individuals. The trajectories are better grouped into three bundles after a preprocessing step. (a) Original 2-D data. (b) Preprocessed data.

where H represents the differential entropy. The mutual information is equivalent to the Kullback–Leibler divergence between the joint density of $\mathbf{y}$ and the product of the marginal densities of the y_i. This measure is zero if and only if the variables y_i are statistically independent. It is possible to show that constraining the y_i to be uncorrelated and of unit variance, the mutual information is equal to

$$I(y_1, \ldots, y_m) = C - \sum_i J(y_i) \tag{6}$$

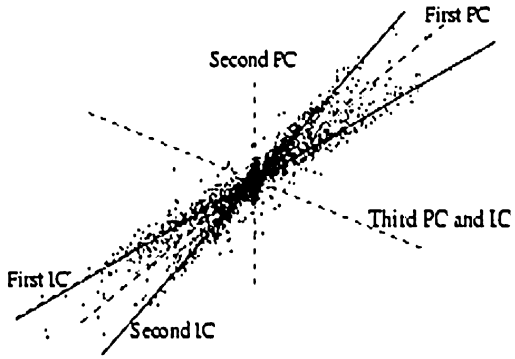


Fig. 5. ICA versus PCA.

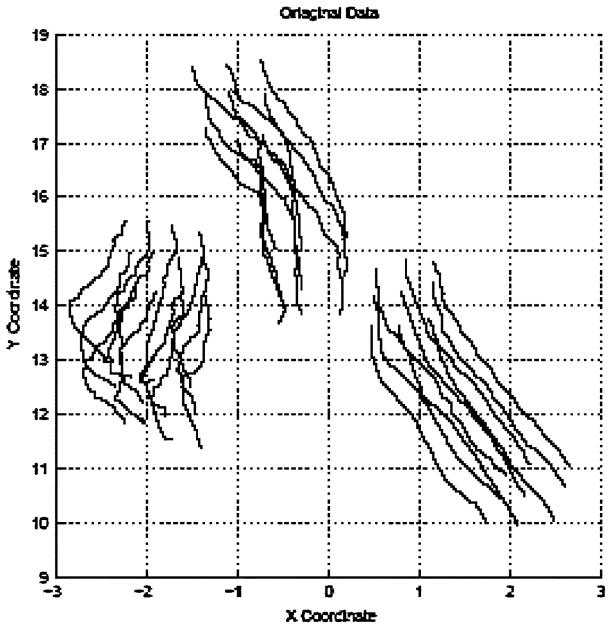


Fig. 6. New set of 30 trajectories, manually tracked, corresponding to ten pedestrians.

where C is a constant and J represents the negentropy, defined as

$$J(\mathbf{y}) = H(\mathbf{y}_{\text{gauss}}) - H(\mathbf{y}) \tag{7}$$

It is well known from information theory that Gaussian variables have the maximum entropy among all the variables with equal variance. We obtain that minimizing the mutual information is equivalent to maximize the negentropy, which actually means to maximize the *non-gaussianity* of the random variables [20]. So, the main assumption in ICA is the non-gaussianity of the source signals.

Geometrical Interpretation: ICA becomes interesting for our purposes when we consider its geometrical interpretation, compared to PCA. While the PCA solution is given by orthogonal axes representing the directions of maximum variance in the data, ICA can be seen as the nonorthogonal extension of PCA. In Fig. 5, this property is illustrated (Fig. 5 is taken from [49]).

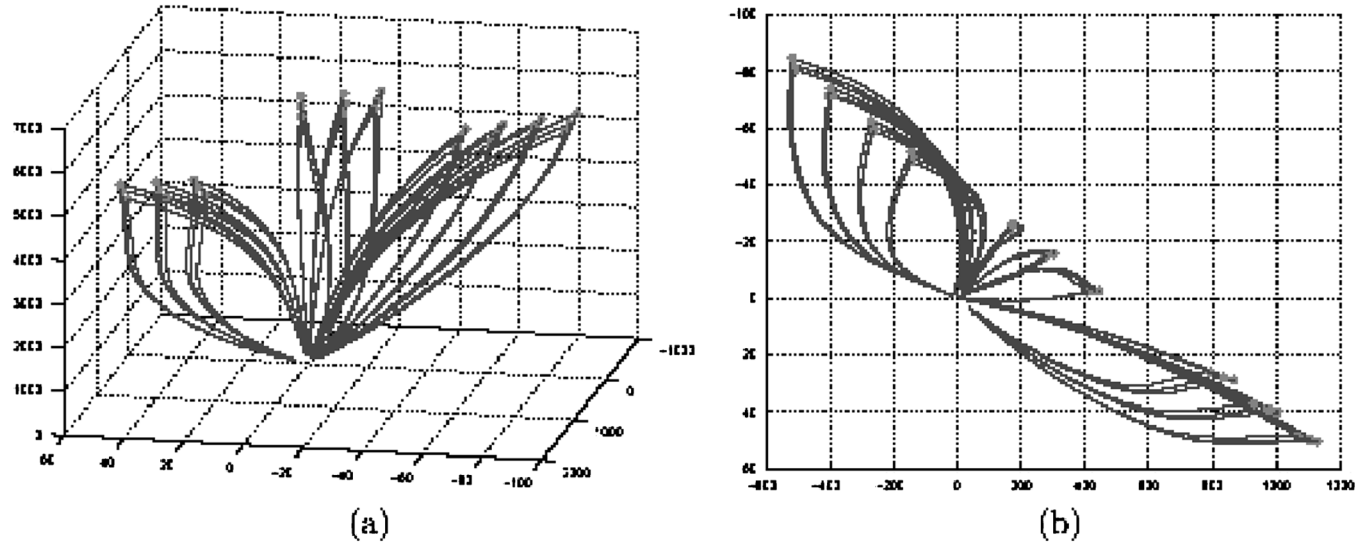


Fig. 7. MCC in 3-D and the x-y 2-D projection.

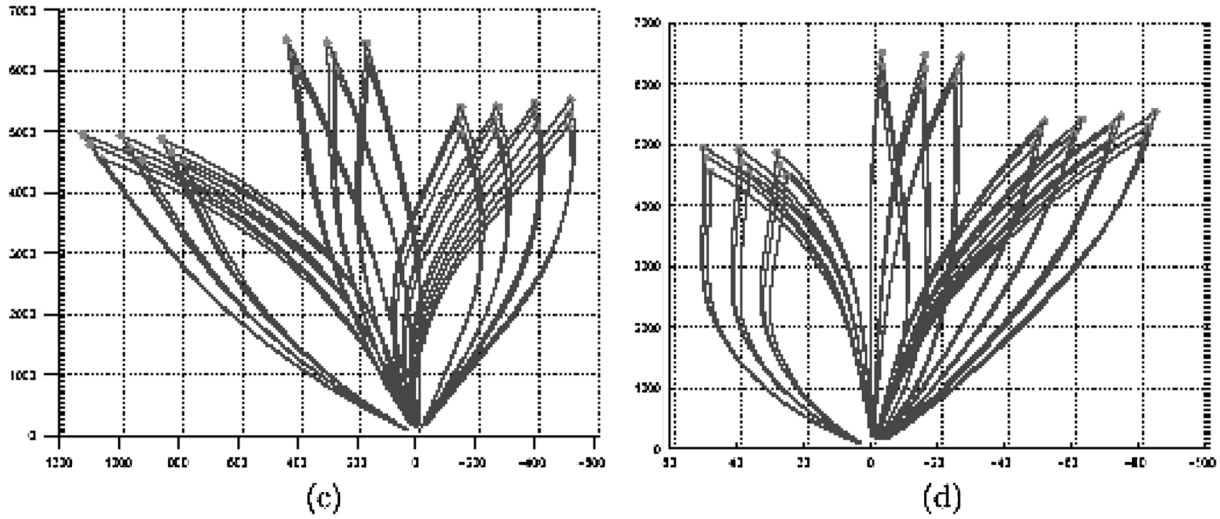


Fig. 8. Other 2-D projections.

When the sources are sparse, ICA provides a better probabilistic model of the data than PCA, which better identifies where the data concentrate. The chosen solution is based on the high-order statistics of the data and represents a nonorthogonal rotation. As a consequence, this transformation can change the relative distances between points, affecting similarity and/or distance measures. For these reasons, it can be quite useful in classification and clustering problems. Fig. 1(b) and Fig. 2. illustrate an example of trajectories projected in the ICA space.

The limitation of this representation resides in an ambiguity intrinsic in the ICA model. In (4), both $\mathbf{s}$ and A are unknown. We can change the order of the independent components keeping untouched the validity of the model. Therefore, the components are estimated up to a permutation matrix. When the ICA model is used, for example, as a dimensionality reduction method (by means of a previous PCA step, where a certain number of eigenvalues of the covariance matrix are kept) this does not change the results. On the contrary, in our case we use the ICA model to estimate a transformation matrix, changing the space where the data are represented. Permuting the order of the estimated components is the same as inverting the axis of the new representation system, changing the data representation itself. This fact leads to different clustering results. One solution can be to keep the ICA estimation that optimizes the clustering. In our specific case, having three independent components, the number of permutations is 3. As a consequence, it is possible to choose the order which maximizes the clustering performances. This ambiguity in the ICA model can seriously deteriorate the performances when such a model is applied to high dimensional datasets, where the number of permutations become huge.

Maximum of Cross-Correlation (MCC): We introduce here the MCC representation. The idea is simply the realization that two identical trajectories are always equally far from a reference one. This simple fact is used here. We fix any trajectory t_1 of the dataset as the reference trajectory. We compute the similarity measure between two trajectories as the cross-correlation function between them. We can look at two trajectories t_1 of length M and t_2 of length N as two real two-dimensional (2-D)

discrete signals, and write the cross-correlation function c between them as

$$c(m,n) = t_1(-m,-n) * t_2(m,n) = \sum_{j=0}^{M-1}\sum_{k=0}^{N-1} t_1(-j,-k)t_2(m-j,n-k). \quad (8)$$

The two trajectories are represented by two matrices of size $M\times 2$ and $N\times 2$, respectively, so the size of the full cross-correlation is $(M+N-1)\times 3$. We show in Fig. 7(a) the 3-D representation of the output c where the axes represent the three columns of the cross-correlation. The new trajectory representation is obtained mapping each pair of trajectories with the *maximum* of their cross-correlation. The intuitive idea is that, independently from the chosen reference trajectory t_1, the maximum of the cross-correlation between two *similar* trajectories t_2 and t_3 with t_1 maps t_2 and t_3 into two close spatial points. In a similar way, two strongly different trajectories will be mapped into two farther spatial points. In Fig. 6, a new set of 30 trajectories, manually tracked from ten pedestrians, is illustrated. The individuals are walking in 3 different groups composed by three, three, and four persons, respectively. Looking at Fig. 6 is easy to identify the three groups, but it is not easy at all to count the ten pedestrians. Figs. 7 and 8 illustrate the 3-D MCC with all the 2-D projections.

This representation presents several advantages over the others. First, it can handle trajectories of different lengths, in a quite easy manner, being the cross-correlation operator independent on the number of points. Second, it allows to map a couple of trajectories into one 3-D point. This remains true for any dimensionality of the dataset and represents a drastic dimensionality reduction. Third, it allows to reduce the clustering problem to a much simpler spatial clustering, which can be handled, with a certain accuracy, by means of the simple Euclidean metric.

VI. Distance/Similarity Measures

Hausdorff Distance: The Hausdorff distance is a metric between nonempty compact point sets. Let $X_1 = (x_{11},\ldots,x_{1m})$ and $X_2 = (x_{21},\ldots,x_{2n})$ be two finite point sets. The Hausdorff distance $H(X_1,X_2)$ is defined as follows:

$$H(X_1,X_2) = \max(h(X_1,X_2), h(X_2,X_1)) \quad (9)$$

where $h(X_1,X_2)$ is the *direct* Hausdorff distance between X_1 and X_2, defined as

$$h(X_1,X_2) = \max_{x_1\in X_1} D(x_1,X_2) \quad (10)$$

where $\forall x_1 \in X_1, D(x_1,X_2)$ is defined as

$$D(x_1,X_2) = \min_{x_2\in X_2} d(x_1,x_2). \quad (11)$$

It identifies the point $x^* \in X_1$ that is farthest (using a prespecified norm d, usually the Euclidean distance) from any point in X_2 and measures the distance from x^* to its nearest neighbor in X_2. Essentially, $h(X_1,X_2)$ ranks each point in X_1 based on its distance from the nearest point in X_2 and then uses the largest ranked such point (x^*) as the distance measure. Similarly, we can define $h(X_2,X_1)$. The Hausdorff distance is the maximum between the direct and inverse distances. As it is well known, this metric is very sensitive to outliers so smoothing operations or other kind of transformations, as for example the ICA representation, are usually performed before to compute the distance. On the other hand it has also some quite good properties. First, it represents a metric and not just a similarity. Second, we can easily apply this measure to sets of different sizes.

Longest Common Subsequence: Longest common subsequence (LCSS) is a similarity measure derived from the Levenshtein distance, also known as edit distance measure [50]. The edit distance is a measure of the similarity between two strings, given by the number of deletions, insertions, or substitutions required to transform one string into the other. In this spirit, and using the notation used in [31], we use what the authors call the $S1$ similarity measure. It does not extend to translations because in our case two parallel trajectories with similar shapes may represent two different individuals. Given two trajectories $A = ((a_{x,1},a_{y,1}),\ldots,(a_{x,n},a_{y,n}))$ and $B = ((b_{x,1},b_{y,1}),\ldots,(b_{x,m},b_{y,m}))$, let Head(A) and Head(B) be two sequences defined as

$$\text{Head(A)} = ((a_{x,1},a_{y,1}),\ldots,(a_{x,n-1},a_{y,n-1}))$$
$$\text{Head(B)} = ((b_{x,1},b_{y,1}),\ldots,(b_{x,m-1},b_{y,m-1})).$$

Definition 6.1: Given an integer $\delta \geq 0$ and a real number $0<\epsilon<1$ the $LCSS_{\delta,\epsilon}(A,B)$ is defined as follows:

$$\begin{cases} 01 \text{ if A or B is empty} \\ \\ 1 + \text{LCSS}_{\delta,\epsilon}(\text{Head(A)},\text{Head(B)}), \\ \text{if } |a_{x,n} - b_{x,m}| < \epsilon \text{ and} \\ |a_{y,n} - b_{y,m}| < \epsilon \text{ and } |n-m| \leq \delta \\ \\ \max(\text{LCSS}_{\delta,\epsilon}(\text{Head(A)},\text{B}), \\ \text{LCSS}_{\delta,\epsilon}(\text{A},\text{Head(B)})), \\ \text{otherwise.} \end{cases}$$

Definition 6.2: Given two trajectories A and B and given $\epsilon \in (0,1)$ and $\delta \geq 0$, the similarity measure $S1$ is defined as follows:

$$S1(\delta,\epsilon,A,B) = \frac{\text{LCSS}_{\delta,\epsilon}(A,B)}{\min(m,n)} \quad (12)$$

The constant δ controls how far in time we can go in order to match a given point from one series to a point in the other time series. ϵ is the matching threshold. LCSS similarity has the very nice property of matching two sequences stretching them, without rearranging the order and allowing for some *unmatched* elements. This is not allowed for example using Euclidean distance or DTW, which require all the elements to be matched, including the outliers. For this reasons, LCSS is normally better in presence of outliers.

VII. Grouping Rule

Our aim is to reduce the bias in the number of targets as estimated by the tracking system. We do not know *a priori* how

many pedestrians are present in the scene. As a consequence, the hierarchical approach represents a natural way of grouping data over a variety of scales. We use *agglomerative* techniques. In this approach, trajectories are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is created. The resulting tree can be analyzed at different scales, to find out different resulting data partitions. An agglomerative algorithm yields a *dendrogram* representing the nested groups of trajectories and the similarity levels at which the grouping changes. Given n trajectories, the pairwise distance information is represented by a vector of length $n(n-1/2)$. The linking method we use to generate the hierarchical tree is based on the average distance measures. Let u and v two clusters of size n_u and n_v, respectively, and let be x_{ui} the ith object in cluster u. We have

$$d(u,v) = \frac{1}{n_u \cdot n_v} \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} \text{dist}(x_{ui}, x_{vj}) \tag{13}$$

where the averaged pair distance between all the object pairs in the two clusters is used. More details are reported in the next section.

VIII. Results

We report in this section the quantitative results obtained by applying the different trajectory clustering procedures to different datasets. We have conducted two experiments, using globally four different datasets. The first experiment, reported in Section VIII-B, compares the clustering results obtained with time series and ICA representations, using the Hausdorff distance and the LCSS similarity measures. The second experiment in Section VIII-C compares the ICA representation with the MCC representation, using the Hausdorff distance and the Euclidean metric, respectively.

The trajectory data used in the experiments refer to two outdoor sequences. In both the scenarios, specific main pedestrian flows are imposed by the architecture of the scenes. The sketches of the layouts are reported in Fig. 9(a) and (b). These spatial configurations justify the simplistic hypotheses discusses in Section IV, which are at the base of the first and second levels in the multilayer hierarchical framework, reported in Fig. 3.

The results are compared defining two kind of errors. We call **e1** the number of *missed* pedestrians, meaning that no clusters refer to an individual. We call **e2** the *over-counted* individuals, meaning those pedestrians having more than one resulting cluster over themselves. We do not consider as an error those clusters which refer to trajectories that are not placed on the pedestrian bodies. Such an error comes actually from the trajectory collection process (i.e., the tracking system) and cannot be corrected with the proposed clustering procedures.

A. Clustering Parameters

The cluster tree (*dendrogram*) generated by the hierarchical clustering algorithm proposes a data structure which can be observed at different scales. In Fig. 10, we report an example,

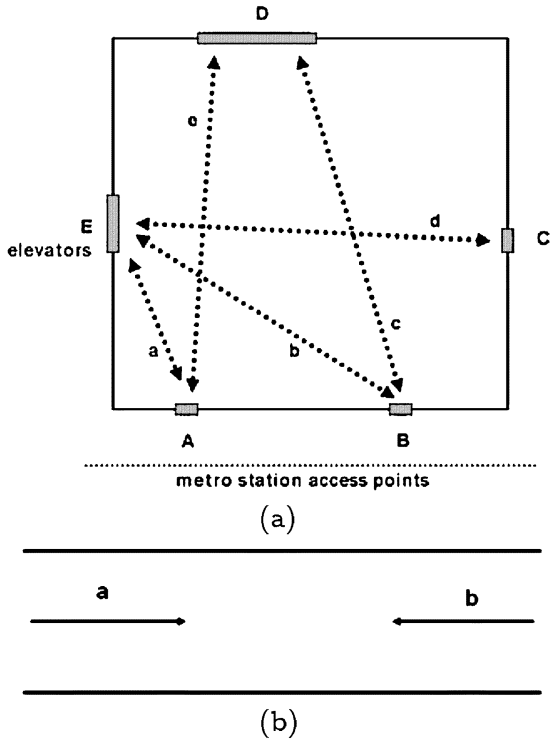


Fig. 9. Main pedestrian flows are reported here for both the sequences. For the more complex *flon* sequence, five main walking directions are present. In the simpler *monaco* case, only two main flows are present. (a) Sketch of the *flon* scene. (b) Sketch of the *monaco* scene.

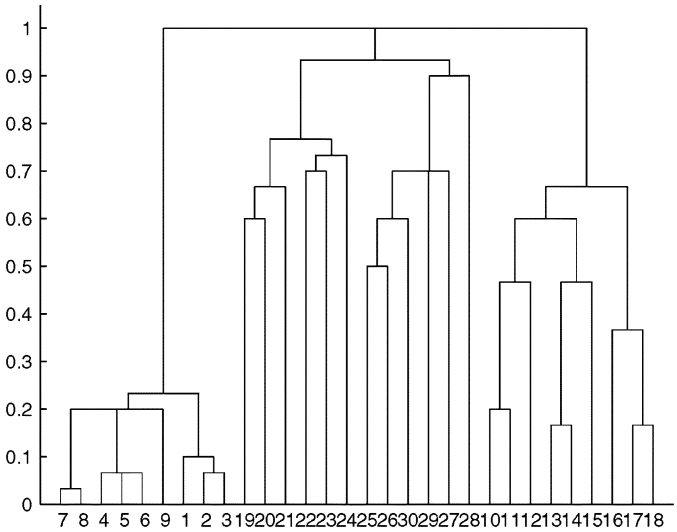


Fig. 10. Dendrogram referring to a time series representation with LCSS similarity, for the first trajectory dataset used in Test 1.

where the x axes corresponds to the patterns that are linked together while the y axes reports the distance between the patterns. The depth in the tree, corresponding to a certain cluster structure, is fixed through a *cutoff* threshold value. This value refers in our case to the inconsistency coefficient between the data and it characterizes each link in the cluster tree by comparing its length with the average length of other links at the same level of the hierarchy. The higher the value of this coefficient, the less similar the objects connected by the link. In our

TABLE II
COPHENETIC CORRELATION COEFFICIENT FOR THE MCC REPRESENTATION USING THE TWO DATASETS OF TEST 1

Dataset with 30 trajectories	
Linking method	Cophenetic correlation
single	0.7953
complete	0.8333
average	0.8336
centroid	0.8336
ward	0.8323
Dataset with 15 trajectories	
Linking method	Cophenetic correlation
single	0.8787
complete	0.8898
average	0.8941
centroid	0.8941
ward	0.8926

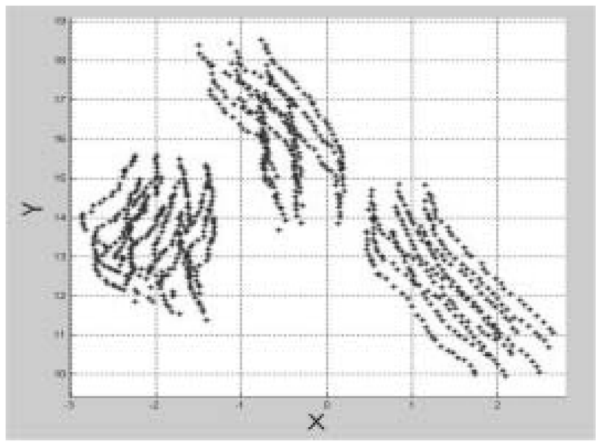
experiments, we use cutoff values in the range 0.7–1. We have verified by visual inspection that such a range assures the best tradeoff between the number of missed pedestrians and the accuracy of the clusters centroids over time, with respect to the available datasets.

Another set of parameters in the clustering procedure are related to the approach used to create the links between patterns. Depending on the way intermediate groups of patterns are merged or splitted, several possibilities are available (single, average, centroid, complete, ward) differing in the way they compare the objects who belong to the intermediate groups to be compared. Details about these methods can be found in [51], [52], and [45]. In order to choose a linking method, we use the *cophenetic* correlation coefficient. It is based on the cophenetic distance between two observations, represented in a dendrogram by the height of the link at which those two observations are first joined. That height is the distance between the two subclusters that are merged by that link. The cophenetic correlation for a cluster tree is defined as the linear correlation coefficient between the cophenetic distances obtained from the tree, and the original distances (or dissimilarities) used to construct the tree. Thus, it is a measure of how faithfully the tree represents the dissimilarities among observations, showing values close to 1 for high quality solutions [53], [54]. We report in Table II the related values for the MCC representation using the two datasets of Section VIII-B. They show slightly better performances of the clustering algorithm when the average and centroid linking methods are used.

B. Test 1

In this first test, we compare the time series representation with the ICA representation. The Hausdorff distance and the LCSS similarities are used with both the representations. The aim of this experiment is to show that better results can be obtained using a more suitable data representation, which reduces the presence of outliers. Actually, also a metric as the Hausdorff one, extremely sensible to outliers, can perform well when it is used with an opportune data representation. Two sets of trajectories are used. The first one is composed by 30 trajectories

(a)



(b)

Fig. 11. First dataset used in Test 1. (a) Trackers used to collect the data. (b) Trajectories generated by the trackers.

TABLE III
RESULTS OBTAINED USING THE HAUSDORFF METRIC AND LCSS SIMILARITY WITH A TIME SERIES REPRESENTATION

num traj	clust. alg	num clust.	num ped	e1	e2
30	*Time series & Hausdorff*	13	10	1	4
30	*Time series & LCSS*	10	10	1	1

manually grabbed and the second one consists in 15 trajectories obtained with the behavioral model-based tracking system, proposed in [10]. The manually tracked points that generate our first data set are placed on ten different pedestrians, three for each of them, and are placed on the head, the body's center and on the middle of feet of the individuals. The selected ten pedestrians walk divided in groups of three, three, and four persons, respectively, as we can see in Fig. 11(a). The goal is to correctly cluster the 30 trajectories in ten different groups. We show in Fig. 11(b) the trajectories.

The results on the first dataset are summarized in Tables III and IV. In Tables V and VI we report the results obtained using the second dataset. Tables III–VI present different interesting points to discuss. The results for the first data set clearly show

TABLE IV
RESULTS OBTAINED USING THE HAUSDORFF METRIC AND LCSS SIMILARITY IN THE ICA SPACE

num traj	clust. alg	num clust.	num ped	e1	e2
30	*ICA & Hausdorff*	10	10	/	/
30	*ICA & LCSS*	10	10	/	/

TABLE V
RESULTS OBTAINED USING THE HAUSDORFF METRIC AND LCSS SIMILARITY WITH A TIME SERIES REPRESENTATION

num traj	clust. alg	num clust.	num ped	e1	e2
15	*Time series & Hausdorff*	6	6	2	2
15	*Time series & LCSS*	1	6	5	/

TABLE VI
RESULTS OBTAINED USING THE HAUSDORFF METRIC AND LCSS SIMILARITY IN ICA SPACE

num traj	clust. alg	num clust.	num ped	e1	e2
15	*ICA & Hausdorff*	6	6	1	1
15	*ICA & LCSS*	6	6	1	1

TABLE VII
RESULTS FOR THE *FLON* SEQUENCE

num traj	num clust.	num ped	e1	e2
Independent Component Analisys:				
31	14	11	0	3
Cross-correlation:				
31	12	11	0	1

how the ICA transformation improves the clustering. We can see it also in the respective results using the Hausdorff and LCSS metric/similarity. The differences of the respective results in the original space are removed in the ICA space, where the Hausdorff distance performs as well as the LCSS similarity measure. This is an implicit indication that the nonorthogonal rotation has reduced the presence of outliers in the trajectories, concentrating the data along the independent directions. We remark the same qualitative improvements for the second data set.

C. Test 2

The results illustrated in the previous section show that a suitable data representation can overcome the drawbacks related to a specific metric. As we have already said in Table V, the ICA representation presents some limitations, due to the nature of the ICA model itself. The independent components are defined up to a permutation matrix. This fact can create problems when we use such components to change the representation of our data. In this section we compare the ICA representation with the MCC. We use two different datasets, both of them obtained using the model-based tracker presented in [10]. The first one used in this experiment consists in 31 trajectories distributed

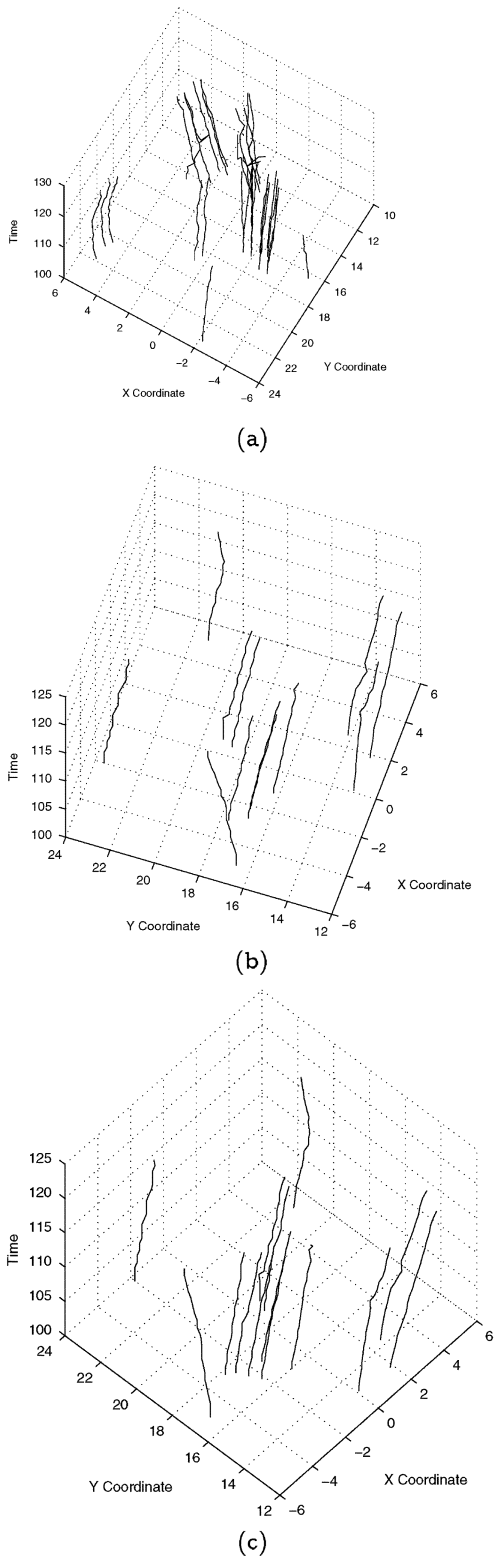


Fig. 12. Results of the clustering on the *flon* trajectory data set. (a) *flon* trajectory set. (b) Cross-correlation-based clustering. (c) ICA-based clustering.

on 11 pedestrians [Fig. 12(a)]. The density of the targets in the scene is high. In particular, we note that the group of four pedestrians walking together [Fig. 13(a)] is highly overestimated by the detection/tracking algorithm. The numerical results are presented in Table VII. The clustering results on the trajectories are

(a)

(b)

(c)

Fig. 13. Visual examples for the *flon* sequence. (a) Final trajectory points without clustering. (b) Final trajectory points after the max-of-crosscorrelation clustering. (c) Same example after the ICA clustering.

TABLE VIII
RESULTS FOR THE *MONACO* SEQUENCE

num traj	num clust.	num ped	e1	e2
Independent Component Analisys:				
43	17	8	0	4
Cross-correlation:				
43	9	8	0	1

shown in Fig. 12(b) and (c) while visual examples are shown in Fig. 13(b) and (c).

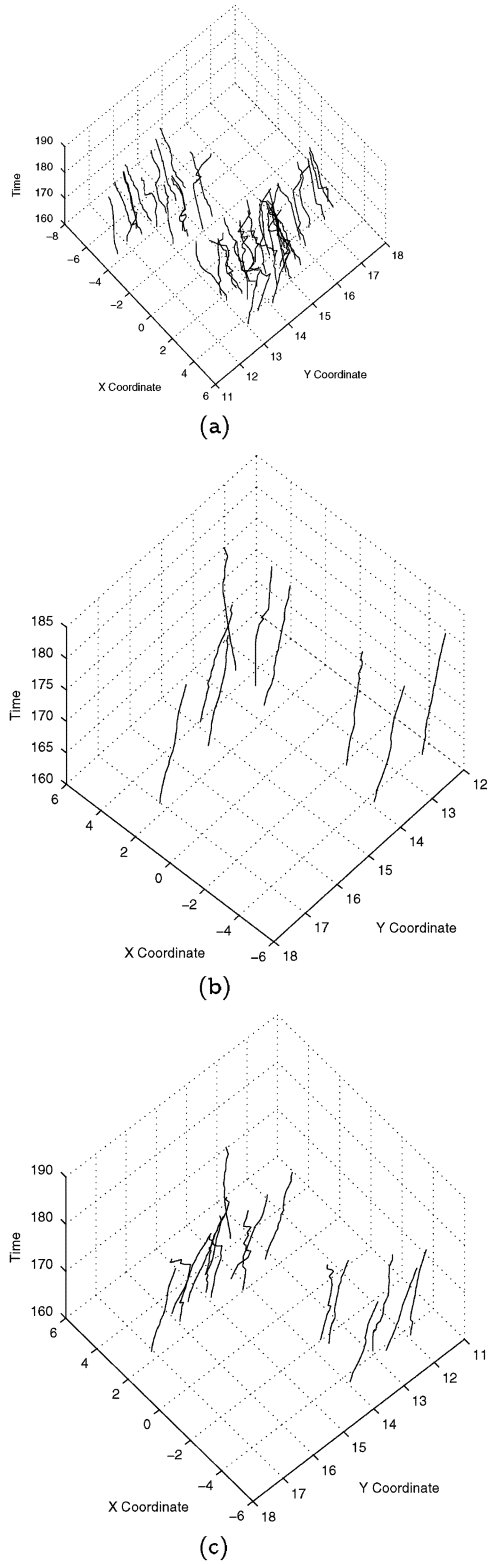


Fig. 14. Results of the clustering on the *monaco* trajectory data set. (a) *monaco* trajectory set. (b) Cross-correlation-based clustering. (c) ICA-based clustering.

The second dataset used in this experiment is strongly overestimated by the detection/tracking system. Eight pedestrians are present in the scene but the trajectories obtained are 43. We report in Table VIII the relative numerical results. The clustering results on the trajectories are shown in Fig. 14(b) and (c) while visual examples are shown in Fig. 15(b) and (c).

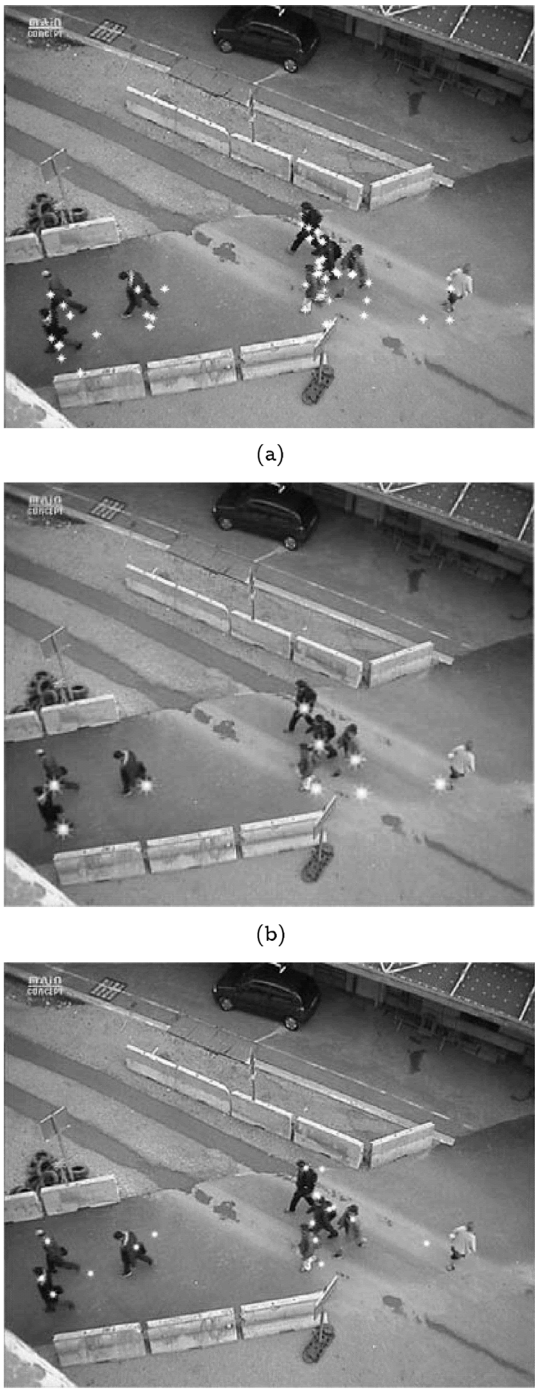


Fig. 15. Visual examples for the *monaco* sequence. (a) Final trajectory points without clustering. (b) Final trajectory points after the max-of-crosscorrelation clustering. (c) Same example after the ICA clustering.

For both the datasets used in this test, the MCC representation performs better than ICA. Given the large overestimation of the number of targets, both the methods provide 0 missed pedestrians. However, the MCC approach shows a better capability in clustering those trajectories belonging to the same individual, resulting in a lower number of over-counted pedestrians.

IX. Conclusion

In this paper we have presented a comparative study of clustering methods for automatic counting of pedestrians in video sequences. The aim is to reduce the bias in the real number of targets present in the scene, as estimated by a general tracking system that overestimates the target number. We do not focus on the errors coming from the detection/tracking steps but rather we attempt to exploit the information provided by it. Allowing for redundance in the detection/tracking step (target overestimation) and correcting with trajectory clustering in a post-processing step, can reduce the problems coming from object detection in cluttered, real environments. At first, the datasets are analysed based on the length and starting point positions of the trajectories. On the resulting *preclustered* datasets, different data representations and distance/similarity measures have been used. More specifically, we first apply both the Hausdorff distance and LCSS similarity for the ICA and time series representations. The results presented in Section VIII-B show that the ICA space provides a more suitable representation with respect to the original space-time domain, reducing the presence of outliers. The second experiment presented in Section VIII-C shows that the *maximum-of-cross-correlation* mapping allows for better clustering results with respect to ICA, at a lower computational cost. The trajectory clustering problem is reduced to a simpler 3-D spatial clustering using the Euclidean metric. The clustering approach to count targets is independent from the algorithm used for tracking, given that it overestimates the number of targets.

Future improvements will consist in relax some of the empirical assumptions made here to analyse the scene under investigation, in order to obtain a more generalizable method.

References

[1] M. Isard and A. Blake, "Contour tracking by stochastic propagation of conditional density," *ECCV*, vol. 1, pp. 343–356, 1996.

[2] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *J. Comput. Graph. Stat.*, vol. 5, no. 1, pp. 1–25, 1996.

[3] M. Isard and A. Blake, "Condensation-conditional density propagation for visual tracking," *Int. J. Comput. Vis.*, vol. 1, no. 29, pp. 5–28, 1998.

[4] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L. V. Gool, "Color-based object tracking in multi-camera environments," in *Proc. 25th Pattern Recognit. Symp. DAGM 2003*, Sep. 2003, pp. 591–599.

[5] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "Object tracking with an adaptive color-based particle filter," in *Proc. Symp. Pattern Recognit. DAGM*, Sep. 2002, pp. 353–360.

[6] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla, "Learning a kinematic prior for tree-based filtering," in *Proc. British Mach. Vis. Conf.*, Norwich, UK, Sep. 2003, vol. 2, pp. 589–598.

[7] N. Johnson and D. Hogg, "Learning the distribution of object trajectories for event recognition," in *Proc. 6th British Cofn. Mach. Vis.*, 1995, pp. 583–592.

[8] R. Rosales and S. Sclaroff, "3D trajectory recovery for tracking multiple objects and trajectory-guided recognition of actions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 117–123.

[9] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, pp. 747–757, Aug. 2000.

[10] G. Antonini, S. Venegas, M. Bierlaire, and J. P. Thiran, "Behavioral priors for detection and tracking of pedestrians in video sequences," *Int. J. Comput. Vis.*, vol. 69, no. 2, pp. 159–180, 2006.

[11] G. Antonini, S. Venegas, J. P. Thiran, and M. Bierlaire, "A discrete choice pedestrian behavior model for pedestrian detection in visual tracking systems," in *Adv. Concepts Intel. Vis. Syst. (ACIVS)*, Brussels, Belgium, Sep. 2004.

[12] G. Antonini, M. Bierlaire, and M. Weber, "Discrete choice models of pedestrian walking behavior," *Transport. Res. B*, vol. 40, pp. 667–687, 2006.
[13] D. M. Gavrila, "The visual analysis of human movement: A survey," *Comput. Vis. Image Understand.*, vol. 73, no. 1, pp. 82–98, Jan. 1999.
[14] B. Heisele, U. Kressel, and W. Ritter, "Tracking non-rigid, moving objects based on color cluster flow," in *Proc. Comput. Vis. Pattern Recognit.*, 1997, pp. 253–257.
[15] B. Heisele and C. Wohler, "Motion-based recognition of pedestrians," in *Proc. 14th Int. Conf. Pattern Recognit.*, 1998, pp. 1325–1330.
[16] R. Agrawal, K. I. Lin, and H. S. Sawhney, "Fast similarity search in the presence of noise, scaling, and translation in times-series databases," in *Proc. VLDB*, 1995, pp. 490–501.
[17] G. P. Box and G. M. Jenkins*, Time Series Analysis, Forecasting and Control*. San Francisco, Ca: Holden-Day, 1970.
[18] A. J. Bell and T. J. Sejnowski, "An information maximisation approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, no. 6, pp. 1129–1159, 1995.
[19] J. V. Stone, "Independent component analysis: An introduction.," *Trends Cogn. Sci.*, vol. 6, no. 2, pp. 59–64, 2002.
[20] A. Hyvrinen and E. Oja, "Independent component analysis: Algorithms and applications.," *Neural Netw.*, vol. 13, no. 4–5, pp. 411–430, 2000.
[21] S. R. L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.
[22] M. Balasubramanian, E. L. Shwartz, J. B. Tenenbaum, V. de Silva, and J. C. Langford, "The isomap algorithm and topological stability," *Science*, vol. 295, no. 5552, p. 7, Jan. 2002.
[23] R. Souvenir and R. Pless, "Isomap and nonparametric models of image deformation," in *Proc. Motion05*, 2005, pp. 195–200.
[24] T. Eiter and H. Mannila, "Distance measures for point sets and their computation," *Acta Informatica*, vol. 34, no. 2, pp. 109–133, 1997.
[25] J. Ramon and M. Bruynooghe, "A polynomial time computable metric between point sets," *Acta Informatica*, vol. 37, no. 10, pp. 765–780, Aug. 2001.
[26] D. Berbdt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *Proc. KDD Workshop*, 1994, pp. 229–248.
[27] E. Keogh and M. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proc. 6th Int.Conf. Knowledge Discovery Data Mining*, Boston, MA, 2000, pp. 285–289.
[28] P. K. Agarwal, L. Arge, and J. Erickson, "Indexing moving points," in *Proc. 19th ACM Symp. Principles Database Syst. (PODS)*, 2000, pp. 175–186.
[29] S. Perng, H. Wang, S. Zhang, and D. S. Parker, "Landmarks: A new model for similarity-based pattern querying in time series databases," in *Proc. ICDE*, 2000, pp. 33–42.
[30] B. Guo, K.-M. Lam, K.-H. Lin, and W.-C. Siu, "Human face recognition based on spatially weighted hausdorff distance," *Pattern Recognit. Lett.*, vol. 24, no. 1–3, pp. 499–507, 2003.
[31] M. Vlachos, G. Kollios, and D. Gunopulos, "Discovering similar multidimensional trajectories," in *Proc. ICDE*, 2002, pp. 673–684.
[32] B. King, "Step-wise clustering procedures.," *J. Amer. Stat. Assoc.*, vol. 69, pp. 86–101, 1967.
[33] M. R. Anderberg*, Cluster Analysis for Applications*. New York: Academic, 1973.
[34] R. C. Dubes and A. K. Jain, "Clustering techniques: The user dilemma," *Pattern Recognit.*, vol. 8, pp. 247–260, 1976.
[35] A. K. Jain and R. C. Dubes*, Algorithms for Clustering Data*. Upper Saddle River, NJ: Prentice-Hall, 1988.
[36] L. A. Zadeh, "Fuzzy sets," *Inf. Contr.*, vol. 8, pp. 338–353, 1965.
[37] J. C. Bezdek*, Pattern Recognit. With Fuzzy Objective Function Algorithms*. New York, NY: Plenum, 1981.
[38] R. N. Dave, "Generalized fuzzy c-shells clustering and detection of circular and elliptic boundaries," *Pattern Recognit.*, vol. 25, no. 7, pp. 713–722, 1992.
[39] A. Strehl and J. Ghosh, "Value-based customer grouping from large retail datasets," in *Proc. SPIE Conf. Data Mining Knowl. Discov.*, Orlando, FL, 2000, pp. 33–42.
[40] M. Wallace and S. Kollias, "Robust, generalized, quick efficient agglomerative clustering," in *Proc. 6th Int. Conf. Enterprise Inf. Syst.*, Porto, Portugal, 2004, pp. 409–416.
[41] D. Boley, "Principal direction divisive partitioning," *Data Mining Knowl. Discov.*, vol. 2, no. 4, pp. 325–344, 1998.
[42] J. Mao and A. K. Jain, "A self-organizing network for hyperellipsoidal clustering (HEC)," *IEEE Trans. Neural Netw.*, vol. 7, no. 1, pp. 16–29, Jan. 1996.
[43] M. J. Symons, "Clustering criterion and multivariate normal mixtures," *Biometrics*, vol. 37, pp. 35–43, 1977.
[44] T. Mitchell*, Machine Learning*. New York: McGraw-Hill, 1997.
[45] G. Antonini and J. P. Thiran, "Trajectories clustering in ica space: An application to automatic counting of pedestrians in video sequences," in *Adv. Concepts Intell. Vis. Syst. (ACIVS)*, Brussels, Belgium, Sep. 2004.
[46] D. Biliotti, G. Antonini, and J. P. Thiran, "Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences," in *Proc. IEEE Motion 2005*, Beckenridge, CO, Jan. 2005, pp. 523–529.
[47] A. Schadschneider, "Cellular automaton approach to pedestrian dynamics—Theory," in *Proc. Pedestrian Evacuation Dynamics*, 2002, pp. 75–86.
[48] D. Helbing, I. J. Farkas, P. Molnar, and T. Vicsek, "Simulation of pedestrian crowds in normal and evacuation simulations," in *Proc. Pedestrian Evacuation Dynamics*, 2002, pp. 21–58.
[49] M. Bartlett, J. Movellan, and T. Sejnowski, "Face recognition by independent component analysis," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1450–1464, Jun. 2002.
[50] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," *Sov. Phys. Dokl.*, vol. 10, no. 8, pp. 707–710, 1966.
[51] C. Fraley and A. E. Raftery, "How many clusters? which clustering method? answers via model-based cluster analysis," *Comput. J.*, vol. 41, no. 8, pp. 578–588, 1998.
[52] H. H. Bock, "Probability models and hypotheis testing in partitioning cluster analysis," in *Clustering and Classification*. Singapore: World Scienctific, 1996, pp. 377–453.
[53] R. V. Hogg and J. Leodolter*, Engineering Statistics*. New York: MacMillen, 1987.
[54] Statistics Toolbox. Release 5.0, MathWorks, 2005.

Gianluca Antonini was born in Acquapendente, Italy, in April 1973. He received the M.S. degree in telecommunication engineering from the University of Siena, Siena, Italy, in 2000 and the Ph.D. degree from the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland in 2005 for work on mathematical modeling of pedestrian behavior and its integration in computer vision applications.

In 2004, he spent one year at the Intelligent Transportation System Program , Massachusetts Institute of Technology, Cambridge. His current research interests include statistical and behavior modeling, computer vision, machine learning, and artificial intelligence.

Jean-Philippe Thiran (M'90–SM'03) was born in Namur, Belgium, in August 1970. He received the electrical engineering and Ph.D. degrees from the Université Catholique de Louvain (UCL), Louvain-la-Neuve, Belgium, in 1993 and 1997, respectively.

From 1993 to 1997, he was the Coordinator of the Image Analysis Group of the Communications and Remote Sensing Laboratory, UCL, mainly working on medical image analysis. In 1998, he moved to the Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, as Senior Lecturer. He is now an Assistant Professor and the Leader of the Image Analysis Group at the Signal Processing Institute, EPFL. His current scientific interests include image segmentation, prior information in image analysis, variational methods in image analysis, multimodal signal processing, medical image analysis, including multimodal image registration, segmentation, computer-assisted surgery, diffusion MRI, etc. He is author or coauthor of 45 journal papers, 86 papers in the proceedings of the main international conferences and holds four international patents.

Dr. Thiran was the Co-Editor-in-Chief of the *Signal Processing International Journal*.