Multi-layer hierarchical clustering of pedestrian trajectories for automatic counting of people in video sequences

David Biliotti[1], Gianluca Antonini[2] and Jean Philippe Thiran[2]

Information Engineering Department[1]
University of Siena
Siena, Italy, 53100

Signal Processing Institute[2]
Swiss Federal Institute of Technology
Lausanne, CH, 1015

Abstract

In this paper we propose an approach to count the number of pedestrians, given a trajectory data set provided by a tracking system. The tracking process itself is treated as a black box providing us the input data. The idea is to apply a hierarchical clustering algorithm, using different data representations and distance measures, as a post-processing step. The final goal is to reduce the difference between the number of tracked pedestrians and the real number of individuals present in the scene.

1 Introduction

The problem of detection and tracking of moving objects in video sequences has been widely tackled in the last decades. Despite the multitude of methods presented in literature ([1, 2, 3, 4, 5, 6, 7] among others), the problem of automatic counting of targets is far to be solved and most of the difficulties met with it are intrinsic in the initialization step (i.e., robust object detection). We do not provide here a detailed description of the state of the art of tracking techniques because out of the scopes of this paper. In this context, we assume to have the trajectory data set generated by a behavioral-based tracking system ([8, 9]) which over-estimates the real number of individuals present in the scene. We aim to investigate how we can improve the estimation of the real number of targets without entering into the tracking process itself. Our problem can be stated as a pure clustering problem for post-processing of trajectory data sets generated by a tracking system.

The three main steps of a clustering algorithm are 1) data representation; 2) distance measures between patterns and 3) grouping rules. The first step can be identified with all those transformations applied to the original data set in order to obtain more discriminant data representations. This problem includes the feature selection process and its final goal is to find good data representations providing at the same time dimensionality reduction. Typically, patterns are represented as multidimensional vectors, where each dimension is a single feature ([10, 11]). A popular feature extraction process is the principal component analysis which does not depend on labeled data, can be used directly and allows to obtain a dimensionality reduction on the data set. Each clustering technique is based on a distance function. The most popular metrics for continuous data representations are all the Minkowski metrics ([12]). They work well when the data set has isolated clusters but often they need a normalization of the data to avoid that the largest-scaled data dominate the others. Mahalanobis distance is used when the assumption is that the class conditional densities are multivariate Gaussian distributions. Interesting approaches are those proposed in [13] and [14] where similarity measures and metrics are defined based on the definition of specific relations between sets of points. Finally, based on the different grouping rules, we have several approaches. The partition of the data can be hard (partition into groups) or fuzzy (each pattern described by a degree of membership to different clusters, see [15, 16]). Hierarchical techniques split or merge data based on a certain criterion generating a cluster-tree structure ([17, 18, 19, 20, 21]). Partitional clustering algorithms divide data in a certain number of groups optimizing a clustering criterion ([22, 19, 20]). 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 ([23, 24]).

In this paper we propose an investigation of hierarchical clustering techniques applied to pedestrian trajectories for automatic counting of people in video sequences. The main idea is that trajectories arising from the same target are more similar to each other than trajectories related to other targets, also in the case of high target density. This can be useful to discriminate among two or more close pedestrians walking togheter. Different representations for trajectory data set are considered and different distance functions are investigated. Finally, comparative results are illustrated.

The paper is organized as follows: we start describing two different data representations in section 2. In section 3 we illustrate an overview of the multi-layer clustering algorithm and in section 4 the different distance functions we have used. Results are presented in section 5 and conclusions and final remarks in section 6.

2 Trajectory representation

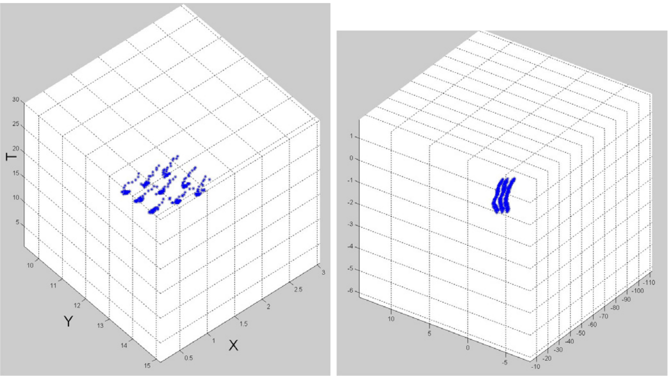
We consider two different trajectory representations and different clustering procedures to each of them, based on two different distance measures.

2.1 Independent Component representation

Independent Component Analysis (ICA) ([25, 26]) 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 statistically independent latent (unobservable) variables, the *sources*, by means of an unknown mixing matrix A. This model can be described by the following equation:

$$\mathbf{X} = A\mathbf{s} \tag{1}$$

where **X** represents the observations and **s** the sources. The number m of observations can differ from the number n of sources. For a general discussion on ICA we can assume, without loss of generality, that $m = n$. The basic hypothesis of the ICA model is the statistical independence of the latent variables. It is possible to show that *independence* is strictly related to *non-gaussianity*. So, the main assumption in ICA is the non-gaussianity of the source signals. ICA becomes interesting for our pourposes when we consider its geometrical interpretation. To better understand the characteristics of ICA, let us think to principal component analysis. PCA is a well known unsupervised statistical method to find useful data representations. Its goal is to find a 'better' basis so that in this new basis the data are uncorrelated. The solution chosen by PCA is an orthogonal matrix depending just on the second-order statistics of the data (i.e. the covariance matrix). ICA can be seen as the non-orthogonal extension of PCA. The chosen solution is based on the high-order statistics of the data and represents a non-orthogonal rotation finding directions with high concentrations of data. As a consequence, this transformation changes the relative distances between points affecting similarity and/or distance measures. For these reasons it can be quite useful in classification and clustering problems. We show in figure 1(a) some examples of trajectory data. These 9 trajectories are manually tracked placing three points on three pedestrians: one on the head, one on the body and a third one on the feet.



(a) Original trajectories

(b) ICA trajectories

Figure 1: The ICA transformation has reduced the distances between trajectories belonging to the same pedestrian.

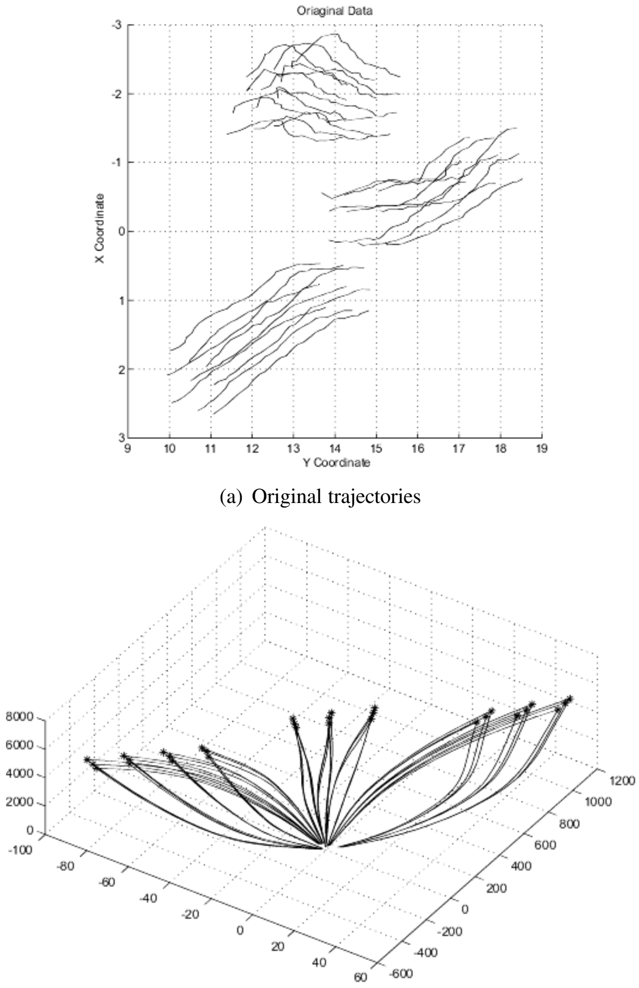
In figure 1(b) we show the same set of 9 trajectories as in figure 1(a) after ICA transformation. We note how the non-orthogonal rotation has improved the discriminant power reducing distances between that trajectories that belong to the same individual.

2.2 The maximum of cross-correlation representation

In this representation we fix any trajectory t_1 of the data set as the reference trajectory. We compute the similarity measure between two data sets 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 2D discrete signals and write the cross-correlation function c between them as:

$$c(m,n) = t_1(-m,-n) * t_2(m,n) = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} t_1(-j,-k) t_2(m-j, n-k) \tag{2}$$

The two trajectories are represented by two matrices of size Mx2 and Nx2 respectively so the size of the full cross-correlation is $(M+N-1)$x3. We show in figure 2(b) the 3D 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



(a) Original trajectories

(b) Cross-correlation

Figure 2: The *maximum-of-cross-correlation* representation.

close spatial points. In a similar way, two strongly different trajectories will be mapped into two farther spatial points. In figure 2 we show this data representation. Three points are placed on each of ten pedestrians and the resulting 30 trajectories are manually grabbed. In figure 2(b) we see how the maximum of the cross-correlation between each pair of trajectories belonging to each person form a well defined cluster in the new domain.

3 Multi-layer hierarchical clustering: an overview

3.1 Hierarchical clustering algorithms

We have no *a-priori* knowledge about the real number of pedestrians present in the scene. So, the hierarchical approach represents a natural way of grouping data over a variety of scales. In our system we use two different hierarchical clustering techniques: *agglomerative* and *divisive*.

Agglomerative
Trajectories are paired into binary clusters, the newly formed clusters are grouped into larger clusters until a hierarchical tree is formed. The resulting tree can be analysed at different levels to find out different resulting clusters. An agglomerative algorithm yields a *dendrogram* representing the nested grouping of patterns and similarity levels at which groupings change. 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 measure. Let be u and v two clusters of size n_u and n_v respectively and let be x_{ui} the *i*th object in cluster u. We have:

$$d(u,v) = \frac{1}{n_u \cdot n_v} \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} dist(x_{ui}, x_{vj}) \qquad (3)$$

where the average paired distance between all the object pairs in the two clusters is used.

Divisive
Hierarchical divisive algorithms start with a single cluster of all the given objects and keep splitting the clusters based on some criterion to obtain a partition of singleton clusters. We report here the main steps of this algorithm ([27]):

1) from the whole set of objects we choose any one to be the first hub;
2) find the point which is farthest from this hub and make it the second hub;
3) for each remaining object, assign it to the closer hub;
4) to decide for a one more hub:
4.a) find the average distance between the two hubs d;
4.b) compute the distance from each object to its hub. If any distance is greater than d, we need another hub.
5) the new hub is the object which is farthest from its respective hub;
6) re-calculate the distance of each point to the new hub and reassing the point to the new hub in the case that the new distance is less than the distance with the previous assigned hub.
7) repeat the iteratively from step 4 until all points are

within a distance d from their hub or all points are themselves hubs.

3.2 The multi-layer structure

The idea for a multi-layer hierarchical clustering arises from the consideration that the comparison between trajectories can be performed from different point of views. Trajectories of different lengths rarely belong to the same person. Moreover, paths belonging to the same target likely start from close spatial points. In figure 3 we illustrate the tree structure of our clustering method.

First level Represents a length-based clustering where trajectories having the same length are grouped together.

Second level Represents the spatial clustering on the trajectories starting points. We assume that trajectories belonging to the same pedestrian start at close spatial positions.

Third level In the case of over-estimated trajectory data sets, each spatial cluster (i.e., a second-level cluster) corresponds to more pedestrians walking close to each other or is generated by many trackers placed on the same human body and its shadow. These situations represent the main source of errors in the target number estimation. At this level of the tree we attempt to count pedestrians.

4 Distance measures

4.1 Hausdorff distance

In the ICA representation the result of the transformation is a new 3D data distribution in the coordinate system defined by the estimated independent components. [1] To compute the distances between the ICA-trajectories we use the well known Hausdorff distance. Using the same notation as [14], the Hausdorff distance d_h between two sets A and B is defined as:

$$d_h(A,B) = \max(\max_{a\in A}(\min d(a,b)|b \in B), \max_{b\in B}(\min d(a,b)|a \in A)) \quad (4)$$

where $d(.,.)$ represents a point-distance function (normally the Euclidean metric). As it is well known, this metric is very sensitive to outliers. The ICA transformation attempts to reduce this sensitivity modifying the relative distances between the trajectory points. 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.

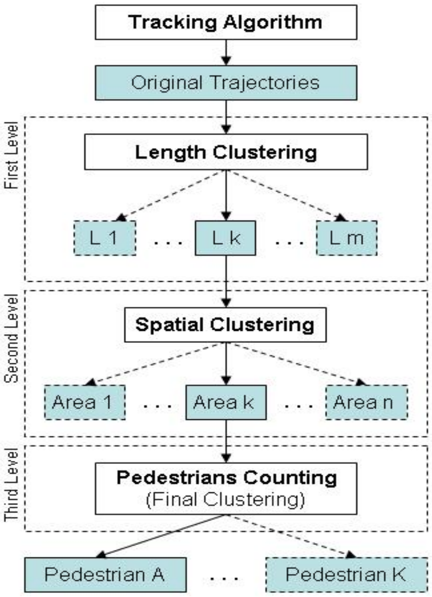


Figure 3: An overview of the multi-layer clustering

4.2 The Euclidean distance

In the case of the *maximum-of-cross-correlation* representation we use the hierarchical divisive clustering based on the Euclidean distance. Infact the representation gives a set of 3D spatial points representing the maximum of the cross-correlation between two given trajectories and the grouping is performed looking at their 3D spatial distribution.

5 Results

We have tested our algorithm on two real outdoor sequences. [2] We summerize the results in tables 1 and 2. The columns **e1** and **e2** represent the *missed* pedestrians and the *over-counted* pedestrians, respectively. By over-counted we mean a pedestrian with more than one resulting cluster over himself. We refer to *missed* pedestrian when no clusters are attached to him. We do not report the errors coming from wrong trajectories generated by the tracking system.

The first data set is composed by 31 trajectories distributed on 11 pedestrians (figure 4(a)). The density of the

[1] We use the FastICA matlab toolbox for the ICA model estimation.

[2] The elaborated and original video sequences can be found at http://ltswww.epfl.ch/ltsftp/antonini/

targets in the scene is high. In particular, we note that the group of four pedestrians walking together (figure 5(a)) is highly over-estimated by the detection/tracking algorithm. The numerical results are presented in table 1. The clustering results on the trajectories are shown in figures 4(b) and 4(c) while visual examples are shown in figures 5(b) and 5(c).

num traj	**num clsuters**	**num ped**	**e1**	**e2**
Indipendent Component Analisys:				
31	14	11	0	3
Cross-correlation:				
31	12	11	0	1

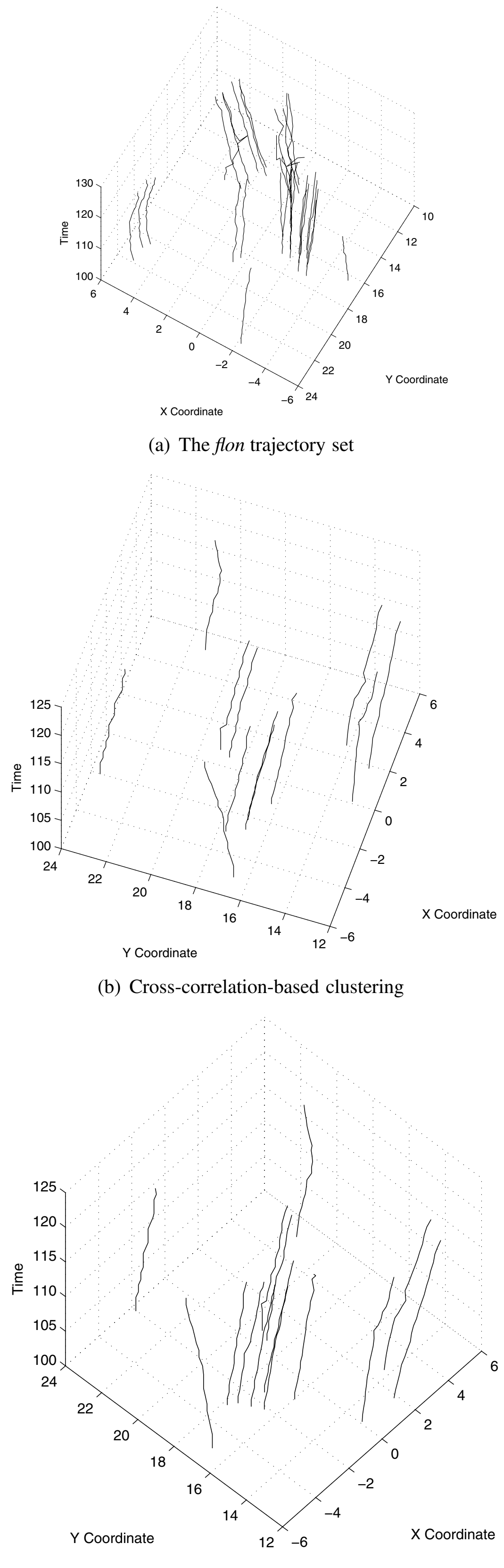
Table 1: Results for the *flon* sequence.

The second data set is strongly over-estimated by the detection/tracking system. Eight pedestrians are present in the scene but the trajectories obtained are 43. We report in table 2 the relative numerical results. The clustering results on the trajectories are shown in figures 6(b) and 6(c) while visual examples are shown in figures 7(b) and 7(c).

num traj	**num clsuters**	**num ped**	**e1**	**e2**
Indipendent Component Analisys:				
43	17	8	0	4
Cross-correlation:				
43	9	8	0	1

Table 2: Results for the *monaco* sequence.

The two different representations we have used give comparable results. We observe better performances related to the max-of-cross-correlation representation. Other advantages are its simplicity, reduction of dimensionality and low computational cost. The limitation of the ICA approach resides in an ambiguity intrinsic in the ICA model. In equation 1 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 this doesn't change the results. On the contrary, in our case we use the ICA model to estimate a transformation matrix to change the representation of the data. Permuting the order of the estimated components is the same as invert the axis of the new representation system, changing the data representation itself. This fact leads to different clustering results.

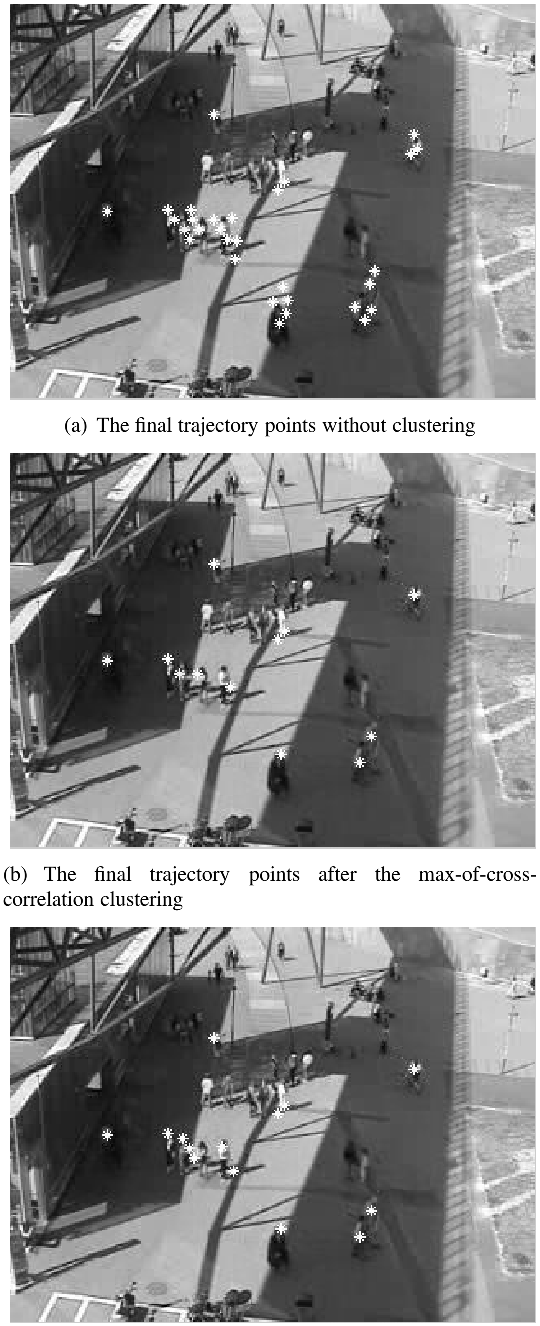


(a) The *flon* trajectory set

(b) Cross-correlation-based clustering

(c) ICA-based clustering

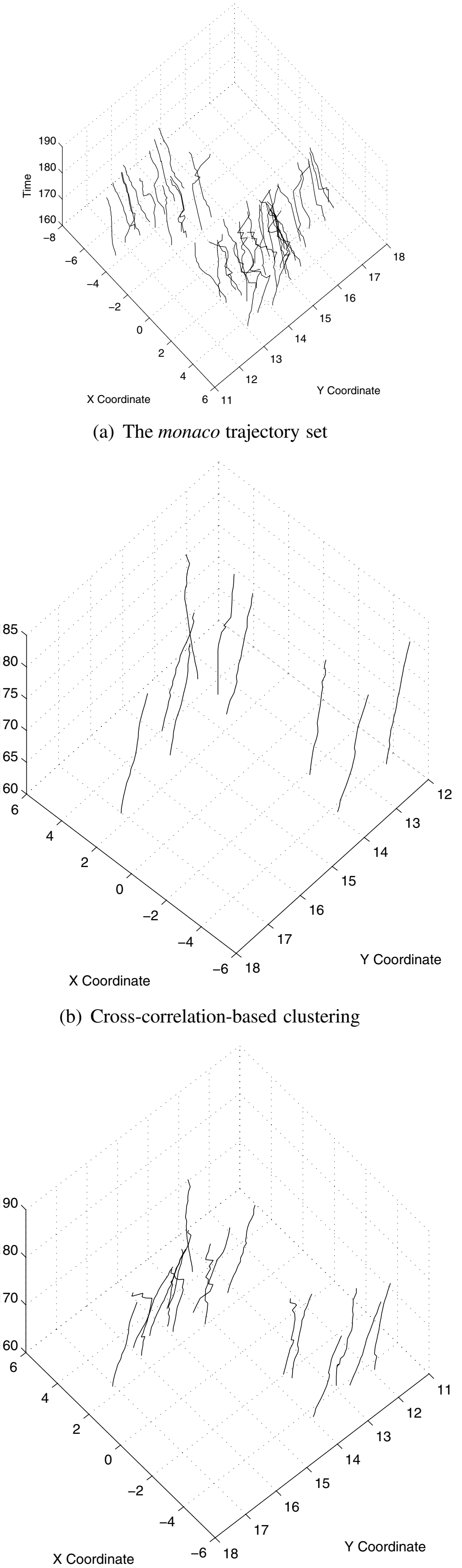
Figure 4: The results of the clustering on the *flon* trajectory data set.


(a) The final trajectory points without clustering

(b) The final trajectory points after the max-of-cross-correlation clustering

(c) The same example after the ICA clustering

Figure 5: Visual examples for the *flon* sequence.

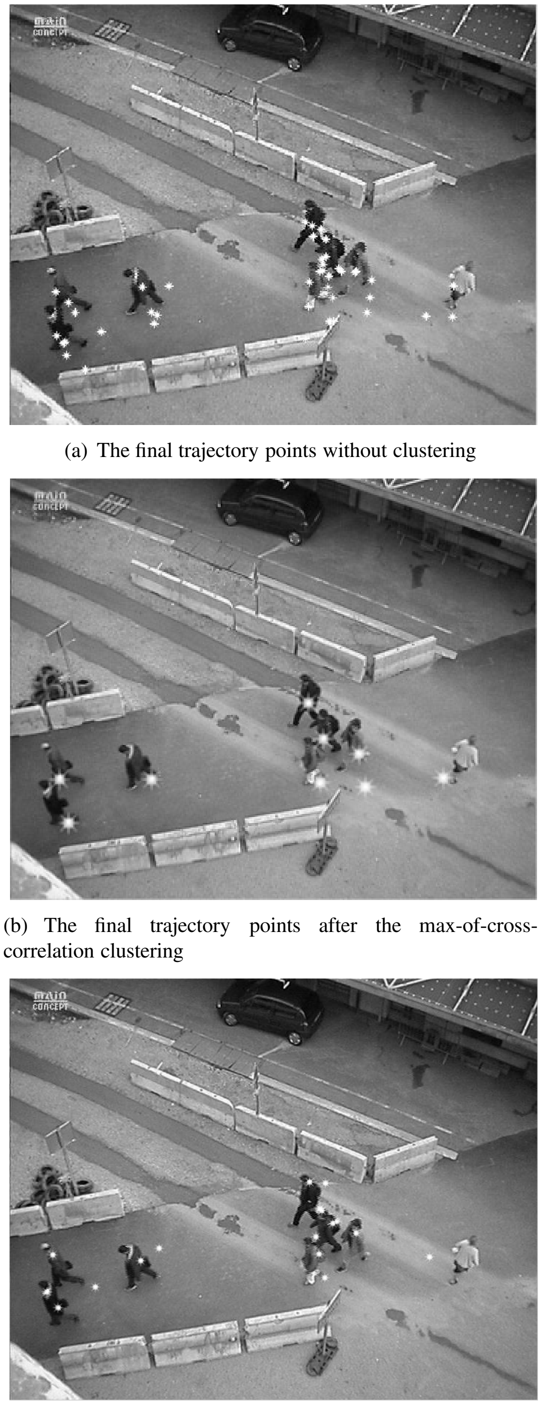


(a) The *monaco* trajectory set

(b) Cross-correlation-based clustering

(c) ICA-based clustering

Figure 6: The results of the clustering on the *monaco* trajectory data set.



(a) The final trajectory points without clustering

(b) The final trajectory points after the max-of-cross-correlation clustering

(c) The same example after the ICA clustering

Figure 7: Visual examples for the *monaco* sequence.

6 Conclusion and future works

In this paper we have presented a multi-layer hierarchical clustering method for post-processing of the trajectory data set generated by a detection/tracking system for pedestrians. Such systems provide an over-estimation of the number of targets present in the scene. We do not focus on the errors coming from the detection/tracking system but rather we attempt to exploit the information provided by it. At first, the data set is analysed based on the length and starting point position of trajectories. On the resulting *pre-clustered* data set, two different data representations have been used. We apply the Hausdorff distance for ICA-transformed trajectories, using an agglomerative hierarchical clustering. The second representation is based on a *maximum-of-cross-correlation* mapping for each pair of trajectories. This allows us to reduce the dimensionality of the problem working with a spatial distribution of resulting 3D points. We use here a divise clustering technique, more indicated for spatial-based data representations. The results of the two approaches show that the cross-correlation-based method is computationally more effective and provides a unique solution for the data representation problem. The same is not true for the ICA transformation where the independent components are estimated up to a permutation matrix, forcing us to examine any possible combination of them. The system is independent from the detection/tracking method which generates the trajectories.

We are currently working to integrate the clustering process to obtain an on-line feedback on the tracking system. We aim to periodically re-initialize the trackers around the centroids of each cluster.

References

[1] G.Kitagawa. Monte carlo filter and smoother for non-gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25, 1996.

[2] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *International Journal on Computer Vision*, 1(29):5–28, 1998.

[3] K. Nummiaro, E. Koller-Meier, T. Svoboda, D. Roth, and L.J. Van Gool. Color-based object tracking in multi-camera environments. In *DAGM03*, pages 591–599, 2003.

[4] N. Johnson and D. Hogg. Learning the distribution of object trajectories for event recognition, 1996.

[5] C.R. Wren and A.P. Pentland. Dynamic models of human motion. In *In Proceedings of FG98*, 1998.

[6] F. Jurie and M. Dhome. Real time 3d template matching. In *CVPR01*, pages 791–796, 2001.

[7] D.DeCarlo and D.Metaxas. Optical flow constraints on deformable models with applications to face tracking. *Int.J.of Computer Vision*, 38(2):99 – 127, 2000.

[8] G.Antonini, S.Venegas, J.P.Thiran, and M.Bierlaire. A discrete choice pedestrian behavior model for pedestrian detection in visual tracking systems. In *ACIVS 2004*, September 2004.

[9] S.Venegas, G.Antonini, J.P.Thiran, and M.Bierlaire. Bayesian integration of a discrete choice pedestrian behavioral model and image correlation techniques for automatic multi object tracking. In *2004 International Conference on Image Processing*. IEEE, October 2004.

[10] R.O.Duda and P.E.Hart. *Pattern Classification and Scene Analysis.* Wiley and Sons, Inc., New York, NY, 1973.

[11] Byoung-Kee Yi and Christos Faloutsos. Fast time sequence indexing for arbitrary lp norms. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pages 385–394. Morgan Kaufmann Publishers Inc., 2000.

[12] S.Lipschutz. *Theory and Problems of Discrete Mathematics*. McGraw- Hill, Inc., New York, NY, 1976.

[13] T.Eiter and H.Mannila. Distance measures for point sets and their computation. *Acta Informatica*, 34(2):109–133, 1997.

[14] J. Ramon and M. Bruynooghe. A polynomial time computable metric between point sets. *Acta Informatica*, 37(10):765–780, August 2001.

[15] R.N.Dave. Generalized fuzzy c-shells clustering and detection of circular and elliptic boundaries. *Pattern Recogn.*, 25:713–722, 1992.

[16] J.C.Bezdek. *Pattern Recognition With Fuzzy Objective Function Algorithms.* Plenum Press, New York, NY, 1981.

[17] R.A.Baeza-Yates. Introduction to data structures and algorithms related to information retrieval. *Information Retrieval: Data Structures and Algorithms*, pages 13–27, 1992.

[18] P.H.Sneath and R.R.Sokal. *Numerical Taxonomy.* Freeman, London, UK, 1973.

[19] M.J.Symon. Clustering criterion and multi-variate normal mixture. *Biometrics*, pages 35–43, 1977.

[20] R.C.Dubes and A.K.Jain. Clustering techniques: The users dilemma. *Pattern Recognition*, 8:247–260, 1976.

[21] J.H.JR.Ward. Hierarchical grouping to optimize an objective function. *J. Am. Stat. Assoc.*, 58:236–244, 1963.

[22] J.Mao and A.K.Jain. A self-organizing network for hyperellipsoidal clustering (hec). *IEEE Trans. Neural Netw.*, 7:16–29, 1996.

[23] T.Mitchell. *Machine Learning.* McGraw- Hill, Inc., New York, NY, 1997.

[24] A.K.Jain and R.C.Dubes. *Algorithms for Clustering Data.* Prentice-Hall Inc., Upper Saddle River, NJ., 1988.

[25] A. J. Bell and T. J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, 7(6):1129–1159, 1995.

[26] A. Hyvrinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[27] R.Clason. Finding clusters: an application of the distance concept. *The Mathematic Teacher*, April 1990.