TRAJECTORIES CLUSTERING IN ICA SPACE: AN APPLICATION TO AUTOMATIC COUNTING OF PEDESTRIANS IN VIDEO SEQUENCES

Gianluca Antonini and Jean Philippe Thiran

Swiss Federal Institute of Technology (EPFL)
Signal Processing Institute (ITS)
CH-1015 Lausanne, Switzerland
home page: http://ltswww.epfl.ch
{Gianluca.Antonini,JP.Thiran}@epfl.ch

ABSTRACT

In this paper we propose a method for the automatic counting of pedestrians in video sequences for (automatic) video surveillance applications. We analyse the trajectory data set provided by a detection/tracking system. When using classical target detection and tracking systems, it is weel known that the number of detected targets is overestimated/underestimated. A better representation for the trajectories is given in the ICA (Independent Component Analysis) transformed domain and clustering techniques are applied to the ICA-transformed data in order to provide a better estimation of the actual number of pedestrians which are present on the scene.

1. INTRODUCTION

In this paper we focus the attention on the output data of an automatic multi-object detection/tracking system, in the particular case of pedestrian tracking. Despite the multitude of methods presented in literature to tackle this problem (see [1, 2, 3, 4, 5, 6]), the detection and tracking of moving objects does not provide yet a reliable method of counting the number of the tracked objects. The discordance between the number of detected objects and the real number of targets in the scene depends on objective difficulties to define properly the target object as an image region respecting predefined properties. All the image segmentation methods are always strongly dependent on image illumination conditions, cluttered backgrounds and partial occlusions between targets. All the blob-detection based methods suffer of underestimating the real number of targets in the scene.

On the contrary, in our detection/tracking system more than one tracker is associated to the same pedestrian, giving an overestimation of the real number of individuals present in the scene. The contribution of this paper is to refine the detection/tracking results by analysing the computed trajectories, finding a better representation in the ICA transformed domain and applying clustering techniques to give a better estimation of the real number of pedestrians present in the scene.

The paper is structured as follows: in section 2 we define the problem in the context of trajectory analysis/clustering. In section 3 we give a short reference to related works in this field. In section 4 we describe our trajectory representation and we give a description of the main ICA concepts. In section 5 we present the two metric/similarity measures adopted to perform the clustering between the trajectory data and in section 6 we present our results. We conclude by presenting our final remarks.

2. INPUT DATA AND PROBLEM DEFINITION

Without entering into details, we summarize here how the input trajectory data are generated. We assume having a calibrated camera, giving a unique correspondence between the image plane and the top view reconstruction of the scene. A large number of hypothetical moving points (trackers) is initialized on the top view plan by means of a grid with a certain resolution. The corresponding projections of these points on the image plan, filtered using a foreground mask, are tracked by means of a visual correlation method (see figure 1). The resulting trajectories are re-projected on the top view plan and filtered again using behavioral constraints (see [7, 8, 9, 10]). [1].

At the end of the filtering stages we have a reduced set of trajectories, originated by the filtered trackers, most of which are correctly placed on the pedestrians. This approach guarantees a good detection of moving targets but introduces an overestimation in the number of the moving objects, caused by the initial grid used to initialize the system and by the errors introduced by visual correlation. More than one tracker can belong to the same moving re-

[1] The interested reader can find a detailed description of the detection/tracking algorithm at http://lts1pc19/index_page.html

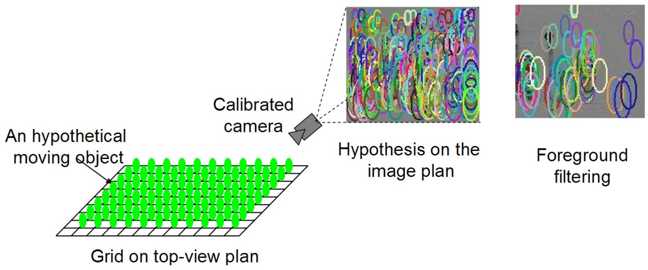


Figure 1: The top view grid used to initialize the algorithm.

gion having at the same time a good behavior (e.g. trackers placed on different parts of the same human body). We show in figure 2 some examples of this problem. This situation makes it impossible to give a good estimation of the actual number of people. This specific problem could

Figure 2: Three examples of multiple trackers represented by rectangles on the same pedestrian.

be approached from different points of view. A deeper image analysis on the moving regions could be useful, even if the low resolution and bad quality of sub-images make the task quite difficult. In line with the approach used to detect pedestrians, we propose in this paper to analyse the dynamic of moving targets, i.e. their trajectories, looking for similarities between trajectories that belong to the same moving region. We attempt to cluster the resulting trajectories assuming that trackers placed on the same pedestrian will generate trajectories much similar to each other.

3. RELATED WORK

Different approaches have been developed in the literature to give measures of similarity between trajectories and/or defining a metric or quasi-metric to compare them. A classical approach widely used in time-series analysis is the DTW (Dynamic Time Warping, see [11, 12]). The main idea behind DTW is to find an alignment of two time series on a common time-axis. Another classical approach is to use a vector-form for trajectories and use a p-norm to compute distances (see [13]). This method does not deal directly with outliers while most of the metrics used to compare data sets are sensitive to this phenomenon. On the contrary, it is possible to gain in simplicity and performances because it allows dimensionality reduction. A better approach would handle the natural dimensionality of the data directly. This leads to model-based clustering techniques, where each cluster will be described by a probability density function and the density of the data will be a mixture of functions. One example of this kind of approach is the mixture of regression models ([14]). A lot of work has been performed in the data mining community, mainly focusing on finding better distance measures to indexing items in databases ([15, 16]). Finally, interesting approaches are those proposed by ([17] and [18]) where similarity measures and metrics are defined based on the definition of specific relations between sets of points.

4. TRAJECTORY REPRESENTATION

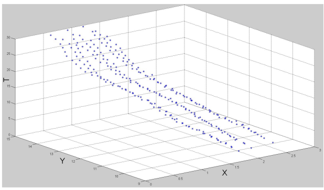
The basic idea of our method is to use a generative probabilistic approach in order to give a better representation of our data, where the presence of outliers is reduced. We consider a trajectory simply as a set of points, where each point is expressed with 3 coordinates (x, y, t), the two plane coordinates x and y and the time t. We show in figure 3 some examples of trajectory data. We note how a 3-D representation gives more discriminant power to group the trajectory data than the 2-D *xy*-plane projection. However, we see in fig. 3(c) an example of how it can be quite difficult to associate a trajectory with the corresponding pedestrian.
Looking at trajectories as 3-D data distributions, we can see that they are quite sparse. It is well known that when the sources are sparse, independent component analysis can be seen as a probabilistic method to find an interesting *non-orthogonal rotation* that concentrates and better represents the data.

4.1. Independent Component representation

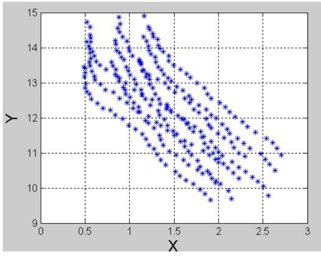
Independent Component Analysis (ICA) ([19, 20, 21]) is a generative model where a set of random variables, the *observations*, are supposed to be generated by a mixing process starting from another set of statistical independent latent (unobservable) variables, the *sources*, by means of an unknown mixing matrix A. This model can be described by the following equation:

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

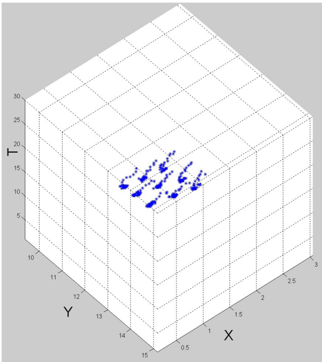
where X represents the observations and s the sources. The number m of observations can differ from the number n of sources. For a general discussion on ICA we



(a) Original trajectories



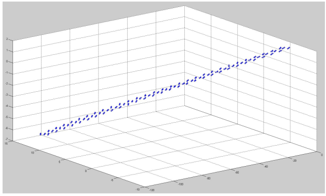
(b) *xy*-plane projection



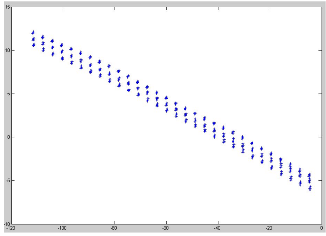
(c) Rotated axes

Figure 3: We show here an example of 9 trajectories: they are manually grabbed from 3 pedestrians who walk close to each other. Three trackers have been placed, respectively, on the head, body-center and feet for each individual.

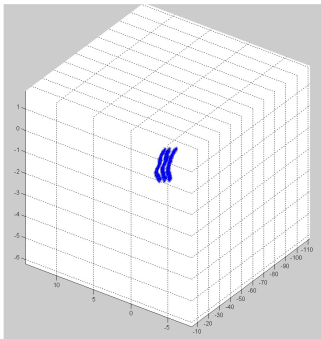
can assume, without loss of generality, that $m = n$. The basic hypothesis of the ICA model is the statistical independence of the latent variables. It is possible to show that *independence* is strictly related to *non-gaussianity*. So, the main assumption in ICA is the non-gaussianity of the source 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 4 the same set of 9 trajectories as in figure 3 after ICA. We note how the non-orthogonal rotation has improved



(a) ICA trajectories



(b) 2-D projection



(c) Rotated axes

Figure 4: The same data set represented in the ICA space. The solution found by ICA algorithm change the relative distances between points giving more disciminant power to cluster the trajectories.

the discriminant power reducing distances between that trajectoriy's points that belong to the same individual. In figure 4(c) become evident the three main trajectories.

5. CLUSTERING ALGORITHM IN ICA SPACE

Having no *a-priori* knowledge about the number of pedestrians in the scene we proceed by grouping trajectories using a hierarchical clustering algorithm. This approach represents a natural way of grouping data over a variety of scales. 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. We proceed in our experiments using the *Hausdorff* distance and the *LCSS* similarity measure between trajectories as *pairwise distances* between observations. Given n trajectories, the pairwise distance information is represented by a vector of length $n(n-1)/2$. Different methods exist to obtain the hierarchical tree structure from the pairwise distance information. We compare the results obtained using four different algorithms: *complete*, *average*, *centroid* and *ward*. We repeat the two clustering procedures both in the original and ICA transformed domain showing that the ICA transformation gives better results with both the distance and similarity measures [2].

5.1. The Housdorff distance

Using the same notation as [18], the Hausdorff distance d_h between two sets A and B is defined as:

$$d_h(A,B) = \max(\max_{a\in A}(\min d(a,b)|b\in B), \max_{b\in B}(\min d(a,b)|a\in A)) \quad (2)$$

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. 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.

5.2. Longest Common Sub-Sequence

The second measure we use (a similarity measure) is based on the *LCSS*, i.e. *longest common sub-sequence*, as defined in [22]. Keeping the same notation as the referenced paper, 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 $A = ((a_{x,1}, a_{y,1}), ..., (a_{x,n}, a_{y,n}))$ and $B = ((b_{x,1}, b_{y,1}), ..., (b_{x,m}, b_{y,m}))$, let *Head*(A) and *Head*(B) be two sequences defined as:

$Head(A) = ((a_{x,1}, a_{y,1}), ..., (a_{x,n-1}, a_{y,n-1}))$

$Head(B) = ((b_{x,1}, b_{y,1}), ..., (b_{x,m-1}, b_{y,m-1}))$.

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

$$\begin{cases} 0 \text{ if A or B is empty} \\ 1 + LCSS_{\delta,\epsilon}(Head(A), Head(B)), \\ \text{if } |a_{x,n} - b_{x,m}| < \epsilon \text{ and } |a_{y,n} - b_{y,m}| < \epsilon \text{ and } |n-m| \leq \delta \\ \max(LCSS_{\delta,\epsilon}(Head(A), B), LCSS_{\delta,\epsilon}(A, Head(B))), \\ \text{otherwise} \end{cases}$$

Definition 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*:

$$S1(\delta,\epsilon,A,B) = \frac{LCSS_{\delta,\epsilon}(A,B)}{\min(m,n)} \quad (3)$$

[2]The independent components are estimated using the FastICA Matlab package (http://www.cis.hut.fi/projects/ica/fastica/)

This similarity measure is more robust to outliers in trajectory data, can operate on trajectories of different lengths and can be efficiently computed by means of dynamic programming.

5.3. The hierarchical tree structure

As already said, different algorithms can be used to create the cluster-tree structure, starting from the pairwise distance vector. They can be 'agglomerative', meaning that groups are merged, or 'divisive', in which one or more groups are split at each stage. Moreover, conventional heuristic methods or more complex model-based methods (where maximum likelihood citerion is used to merge different groups) are used at each step to merge or split the current cluster-tree structure. In this paper we focus our attention on data representation for trajectory clustering. In this spirit we use simple conventional agglomerative methods. We shortly describe in the following the four used linking methods and we remind the interested reader to [23] and [24], for more complex clustering approaches.

Let be 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:

- **complete:** this method uses the largest distance between two objects in two groups:

$$d(u,v) = \max(dist(x_{ui}, x_{vj})) with i = 1, ..., n_u, j = 1, ..., n_v \quad (4)$$

- **average:** uses the average paired distance between all the object pairs in the two clusters

$$d(u,v) = \frac{1}{n_u \cdot n_v} \sum_{i=1}^{n_u} \sum_{j=1}^{n_v} dist(x_{ui}, x_{vj}) \quad (5)$$

- **centroid:** two groups are compared using the distance relative to their centroids. The centroid x_{uc} for the cluster u is defined as:

$$x_{uc} = \frac{1}{n_u} \sum_{i=1}^{n_u} x_{ui} \quad (6)$$

- **ward:** this method uses the incremental sum of squares

$$d(u,v) = \frac{n_u n_v d_{uv}^2}{n_u + n_v} \quad (7)$$

  where d_{uv} is computed with the centroid method.

6. RESULTS

In our experiments we use two sets of trajectories. The first one is composed by 30 trajectories manually grabbed and the second one consists in 15 trajectories obtained with our detection/tracking system. We show in the following the obtained results.

Test 1
The manually tracked points that generate our first data set are placed on 10 different pedestrians, 3 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 10 pedestrians walk divided in goups of respectively 3, 3 and 4 persons, as we can see in figure 5. The goal is to correctly cluster the 30 trajectories in 10 different clusters. We show in figure 5 the trajectories.
The results of the first data set are summarized in tables 1 and 2 where column **e1** represents the *missed* pedestrians and column **e2** represents the number of *over-counted* pedestrians. By *over-counted* we mean a pedestrian with more than one resulting cluster over himself. We talk about *missed* pedestrian when no clusters refer to him.

num traj	**clustering alg**	**num clsuters**	**num ped**	**e1**	**e2**
Hausdorff distance:					
30	*complete*	12	10	1	3
	average	13		1	4
	centroid	11		2	3
	ward	10		1	4
***LCSS* similarity**:					
30	*complete*	11	10	/	1
	average	10		1	1
	centroid	13		/	3
	ward	11		/	1

Table 1: Results obtained using the Hausdorff metric and *LCSS* similarity in the original space

Test 2
The second data set represents a subset of trajectories computed by our detection/tracking system. We show the data in figure 6 and the clustering results in tables 3 and 4.

6.1. Comments

Tables 1, 2, 3, and 4 present different interesting points to discuss. The results for the first data set clearly show 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 non-orthogonal 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. The complete, average and centroid linking algorithms for the data in the original space and using LCSS (see table 3) show how the selected threshold is too high to let LCSS similarity finding differences in the data distribution. The same experiments in the transformed domain show how the grouping power has been improved, changing the representation but keeping the same clustering threshold. This threshold, which indicates the level in the hierarchical tree, has been fixed in all our tests at the value of 0.8. Of course, the results can be improved with an appropriate tuning of the threshold value. We report finally in the next figure the images related to the centroid linking algorithm. The average trajectory for each clus-

num traj	**clustering alg**	**num clsuters**	**num ped**	**e1**	**e2**
Hausdorff distance:					
30	*complete*	10	10	/	/
	average	10		/	/
	centroid	10		/	/
	ward	10		/	/
***LCSS* similarity**:					
30	*complete*	10	10	/	/
	average	10		/	/
	centroid	10		/	/
	ward	10		/	/

Table 2: Results obtained using the Hausdorff metric and *LCSS* similarity in ICA space

num traj	**clustering alg**	**num clusters**	**num ped**	**e1**	**e2**
Hausdorff distance:					
15	*complete*	6	6	2	2
	average	6		2	2
	centroid	7		1	2
	ward	6		2	2
***LCSS* similarity**:					
15	*complete*	1	6	5	/
	average	1		5	/
	centroid	1		5	/
	ward	5		2	1

Table 3: Results obtained using the Hausdorff metric and *LCSS* similarity in the original space

num traj	clustering alg	num clusters	num ped	e1	e2
Hausdorff distance:					
	complete	6		1	1
15	*average*	6	6	1	1
	centroid	6		1	1
	ward	6		1	1
***LCSS* similarity**:					
	complete	4		3	1
15	*average*	6	6	1	1
	centroid	6		1	1
	ward	6		1	1

Table 4: Results obtained using the Hausdorff metric and *LCSS* similarity in ICA space

ter has been computed and the red markers represent the starting points of such trajectories.

6.2. Limitations

The limitation of this approach resides in an ambiguity intrinsic in the ICA model. In equation 1 both **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. 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. In our case, this choice has been done by visual inspection.

7. CONCLUSION AND FUTURE WORKS

In this paper we have shown a possible approach to solve the problem of counting targets in detection/tracking systems of moving objects, taking pedestrians as specific application. This classical image processing task has been approached as a trajectory clustering problem. We have shown that despite the multitude of methods present in the literature in different domains such as time series analysis and data mining, a simple changing in the data representation can improve the final results. We consider the trajectories as 3D data distributions and we use independent component analysis to estimate the transformation matrix. In the ICA space the data appear better grouped around the independent directions and the presence of outliers is reduced. With this new representation, standard distance measures between sets of points, such as the Hausdorff distance, can perform as well as more sophisticated similarity measures such as the *LCSS*.

8. ACKNOWLEDGMENT

This work is supported by the Swiss National Science Fundation under the NCCR-IM2 project and by the Swiss CTI under project Nr. 6067.1 KTS, in collaboration with VisioWave SA, Ecublens, Switzerland.

9. REFERENCES

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

[2] L.J. Van Gool K. Nummiaro, E. Koller-Meier. Object tracking with an adaptive color-based particle filter. In *DAGM-Symposium Pattern Recognition*, pages 353–360, 2002.

[3] A. W. Senior. Tracking with probabilistic appearance models. In *ECCV workshop on Performance Evaluation of Tracking and Surveillance Systems*, pages 48–55, 2002.

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

[5] Hedvig Sidenbladh, Michael J. Black, and David J. Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *ECCV (2)*, pages 702–718, 2000.

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

[7] Michel Bierlaire, Gianluca Antonini, and Mats Weber. Behavioral dynamics for pedestrians. In K. Axhausen, editor, *Moving through nets: the physical and social dimensions of travel*, pages 1–18. Elsevier, 2003.

[8] G.Antonini, M.Bierlaire, and M.Weber. Simulation of pedestrian behavior using a discrete choice model calibrated on actual motion data. In *4th STRC Swiss Transport Research Conference*, Monte Verita, Ascona, Switzerland, 2004.

[9] G.Antonini, S.Venegas, J.P.Thiran, and M.Bierlaire. A discrete choice pedestrian behavior model for pedestrian detection in visual tracking systems. Technical Report ITS-03-04, Signal Processing Institute, EPFL, 2004. http://lts1pc19/antonini/index.html.

[10] 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. Technical Report ITS 07.04, Signal Processing Institute, EPFL, 2004. http://lts1pc19/antonini/index.html.

[11] D.Berbdt and J.Clifford. Using dynamic time warping to find patterns in time series. In *In Proc. of KDD Workshop*, 1994.

[12] E.Keogh and M.Pazzani. Scaling up dynamic time warping for datamining applications. In *In Proc. 6th Int.Conf. on Knowledge Discovery and Data Mining*, Boston, MA, 2000.

[13] 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.

[14] Scott Gaffney and Padhraic Smyth. Trajectory clustering with mixtures of regression models. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 63–72. ACM Press, 1999.

[15] P.K.Agarwal, L.Arge, and J.Erickson. Indexing moving points. In *Proc. of the 19th ACM Symp. on Principles of Database Systems (PODS)*, pages 175–186, 2000.

[16] S.Perng, H.Wang, S.Zhang, and D.S.Parker. Landmarks:a new model for similarity-based pattern querying in time series databases. In *In Proceedings if ICDE*, pages 33–42, 2000.

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

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

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

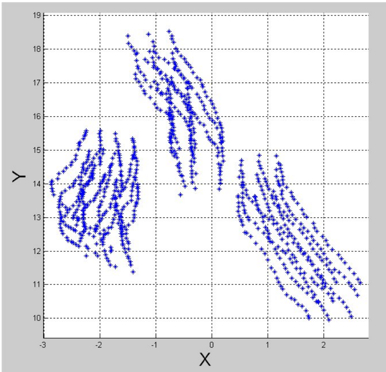
[20] J. V. Stone. Independent component analysis: An introduction. *Trends in Cognitive Sciences*, 6(2):59–64, 2002.

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

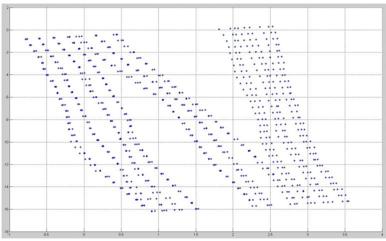
[22] M.Vlachos, G.Kollios, and D.Gunopulos. Discovering similar multidimensional trajectories. In *ICDE*, 2002.

[23] Chris Fraley and Adrian E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

[24] H. H. Bock. Probability models and hypotheis testing in partitioning cluster analysis. In *Clustering and Classification*, pages 377–453. World Science Publishers, 1996.



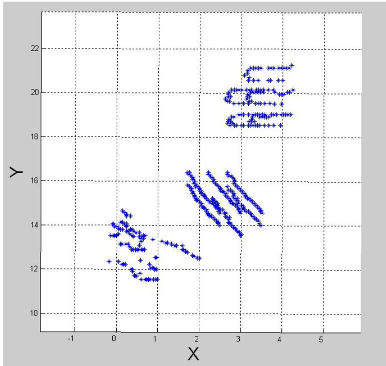
(a) Projection on the *x*-*y* plane



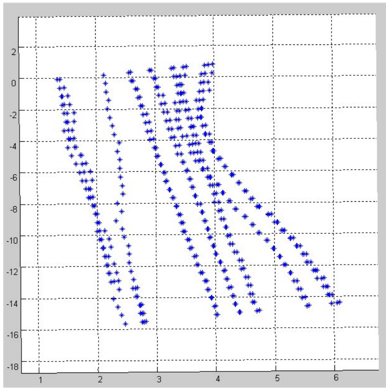
(b) 2D projection in the ICA space

(c) The corresponding starting points placed on pedestrians

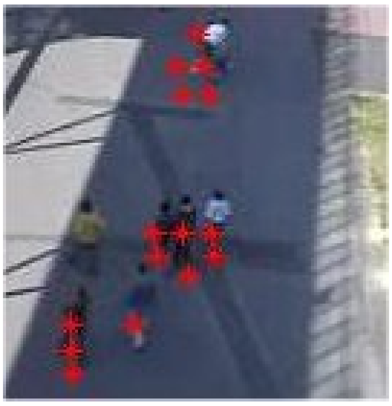
Figure 5: Illustration of the manually grabbed trajectories.



(a) Projection on the *x*-*y* plane



(b) 2D projection in the ICA space



(c) The corresponding starting points placed on pedestrians

Figure 6: Illustration of the automatically tracked trajectories.