# Automatic trajectory clustering for generating ground truth data sets

Julia Moehrmann, Gunther Heidemann
Intelligent Systems Group, Universität Stuttgart, Germany

## ABSTRACT

We present a novel approach towards the creation of vision based recognition tasks. A lot of domain specific recognition systems have been presented in the past which make use of the large amounts of available video data. The creation of ground truth data sets for the training of theses systems remains difficult and tiresome. We present a system which automatically creates clusters of 2D trajectories. The results of this clustering can then be used to perform the actual labeling of the data, or rather the selection of events or features of interest by the user. The selected clusters can be used as positive training data for a user defined recognition task – without the need to adapt the system. The proposed technique reduces the necessary user interaction and allows the creation of application independent ground truth data sets with minimal effort. In order to achieve the automatic clustering we have developed a distance metric based on the Hidden Markov Model representations of three sequences – movement, speed and orientation – derived from the initial trajectory. The proposed system yields promising results and could prove to be an important steps towards mining very large data sets.

**Keywords:** Hidden Markov Model, HMM based representation, clustering, ground truth data

## 1. INTRODUCTION

Due to emerging video and surveillance cameras, the importance of image recognition systems steadily increases. Image recognition plays an important role in surveillance applications and for context aware systems. As of today, image recognition remains specialized on the recognition task. In general, an image recognition system is developed for a special application and the features used for the recognition task are selected according to their suitability for the task. The identification of suitable features requires a high level of knowledge from the developer of the recognition system, therefore these systems cannot yet be applied on a large scale for a multitude of tasks. Additionally, the act of training image recognition systems with a sufficient amount of data is a laborious and expensive task. Identifying positive and negative data samples in the training data set demands numerous user interactions. We believe that the simplification of the training process is a crucial step towards the supply of image recognition systems on a larger scale. Therefore, this work focuses on the creation of ground truth data sets. A method for automated clustering of time-series data is presented which provides a major step towards reducing necessary user interaction and facilitating the creation of ground truth data sets for video data.

Currently, the creation of ground truth data sets presents one of the major problems in creating image recognition systems. In order to achieve sufficient classification results, very large image data sets need to be acquired and labeled. According to the kind of image data or events to be detected, it might be necessary to label individual images or sequences of images, e.g. within a video. Although this presents a major problem, research on image recognition has largely neglected this topic. Existing clustering techniques for time-series data tend to focus on the detection of unusual events or on the spatial distance between the trajectories. However, in order to cluster time-series data for application independent labeling tasks, the clustering technique must not make assumptions about the nature of used data trajectories. The clustering technique cannot use the number of occurrences as a criteria, since common data trajectories may be of interest, which occur more often than others. The proposed clustering technique, which applies the Hidden-Markov-Model (HMM) based distance metric,[1] presents a suitable method for an application independent clustering of time-series data. The goal of the proposed clustering technique is the definition of a generic distance metric which does not rely on prior assumptions about the nature

Further author information: Send correspondence to J. Moehrmann
E-mail: Julia.Moehrmann@vis.uni-stuttgart.de

of the data. In our test case, the proposed clustering technique successfully combined trajectories belonging to the same path.
Due to our motivation being the simplification of the creation of ground truth data sets, the clustering technique is expected to provide a larger number of clusters than necessary. In respect to the interactive labeling task, the error of two paths being identified as one path is worse than a few data trajectories not being assigned to their correct clusters. In the case of two paths being assigned to one cluster, the user has to identify every single trajectory and assign its true category. In the second case the user would only have to identify the few non-clustered trajectories.
The following section gives an overview on related work, section 3 discusses the basics of HMM-based representations, section 4 describes the experimental setup and the results obtained with the application of the proposed technique. Section 5 concludes this work with an overview on future work.

## 2. RELATED WORK

Liao[2] presented a survey on clustering techniques of time-series data. The clustering techniques are divided into three categories, raw-data based, feature based and model based techniques. Although feature based techniques provide a way to reduce the high-dimensional raw data to a set of relatively low dimensional features, these approaches are generally application dependent, with a certain set of features being appropriate for a certain application. Since our goal is to provide an application independent clustering technique for the simplification of labeling tasks, these approaches are unsuitable. The raw-data based approaches are better suitable for application independent tasks, but usually require the data to be uniformly sampled, i.e. the trajectories being of the same length. Application independent time-series data cannot be subject to this condition without loss of information. However, sub-sampling or interpolating time-series data in order to create data sets of equal length are common techniques to overcome this problem.[3–5]
Distance metrics based on spatial distances have been applied to time-series clustering tasks.[3] However, the necessity to handle the high-dimensional raw data may prove to be a drawback, depending on the dimensionality in the specific application. Additionally, the spatial distance may not be appropriate for certain tasks, which has to be investigated if the goal is an application-independent technique. A simple example is shown in Figure 1, where the spatial distances do not fit the intuitive path associations. Additionally, Hervieu discussed the need for distance metrics for time-series data to take temporal causality into account by comparing their approach to the Bhattacharyya distance between histograms.[6] Anjum et al.[7] present a fuzzy clustering approach which aims at clustering trajectories without the need for prior assumptions. Their approach uses Mean-shift over several feature spaces to obtain a fuzzy clustering. However, issues related with labeling tasks were not considered in this work.
Different distance metrics have been compared by Zhang et al.[8] according to their applicability to outdoor surveillance scenarios. Although Zhang et al. found that a Hidden Markov Model (HMM) based distance metric is less successful than common distance metrics (Euclidean, Hausdorff) in outdoor scenarios, the HMM-based approach was shown to be successful in other research studies.[1,9] The HMM-based approach represents each trajectory by the Hidden Markov Model which creates this trajectory. The distance metric calculates the similarity between two trajectories using the cross-likelihood ratio.[10] A major advantage of the HMM-based distance metric is the ability to handle time-series data of varying length, due to the nature of HMMs. Another advantage is the ability of this approach to identify similar trajectories according to their shape. The example given in Figure 1 can be correctly solved using this approach.
Bashir et al.[11] used an HMM representation of features extracted using Curvature Scale Space (CSS) and Centroid Distance Function (CDF) for activity recognition. Jiang et al.[9] used the HMM-based approach together with the Bayesian Information Criterion (BIC) to identify unusual events in a dynamic hierarchical clustering process. Although this approach is aimed at clustering time-series data independently from the application, the authors assume that the overall goal of the pattern recognition task is the detection of unusual events. Similarly, Xiang et al.[12] and Zhong et al.[13] are interested in the detection of unusual events only. This assumption is valid for most surveillance scenarios, however, it is not for the task of preparing the time-series data for the manual labeling. With the growing amount of available memory and the propagation of context-aware systems, common events and information increasingly become of interest. The more pattern and image recognition enhances its

abilities, the more likely it is that image-based sensors will replace physical sensors in the future and monitor everyday events.

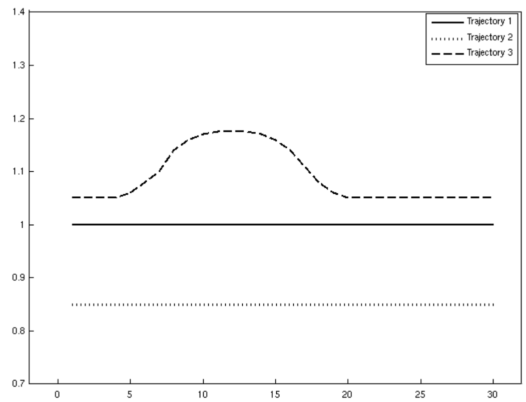


Figure 1. Three uniformly sampled trajectories, with trajectory 1 and trajectory 2 belonging to the same path and trajectory 3 describing a different path. The mean euclidean distance is calculated as $dist_{AB} = \frac{1}{N}\sum_{n=1}^{N} \|A(n) - B(n)\|$. The distance between trajectories 1 and 2 is $dist_{12} = 0.15$, the distance between trajectories 1 and 3 is $dist_{13} = 0.0902$. The example shows that spatial proximity is not necessarily suitable for the identification of similar paths.

## 3. HMM-BASED REPRESENTATION

This section discusses the background of the HMM-based distance metric. A trajectory is denoted as $T_i$ and the HMM which models this trajectory as $\theta_T^i$. The trajectory is given as a sequence of positions $T_i = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$, with $N$ being the length of the trajectory. The distance, i.e. the dissimilarity, between the HMMs $\theta^i$ and $\theta^j$ is denoted as $dist_M(\theta^i, \theta^j)$. Equation (1) gives the dissimilarity of two HMMs $\theta_T^i$ and $\theta_T^j$ using the cross-likelihood ratio.[10] The dissimilarity of two trajectories $T_i$ and $T_j$ based on their HMM representations is given as $dist_T(i, j)$ in Equation (2).

$$\begin{aligned} dist_M(\theta^i, \theta^j) &= logL_i^i + logL_j^j - logL_i^j - logL_j^i \quad (1)\\ dist_T(i, j) &= dist_M(\theta_T^i, \theta_T^j) \quad (2)\end{aligned}$$

with

$$L_i^j = P(T_i, \theta_T^j), \quad (3)$$

being the probability that trajectory $T_i$ is generated by HMM $\theta_T^j$. Equivalently, $L_i^i$ denotes the probability that trajectory $T_i$ is being generated by its corresponding HMM $\theta_T^i$. The idea behind Equation (1) is that if $T_i$ and $T_j$ are almost identical trajectories, the probability of $T_i$ being generated by $\theta_T^j$ is high and vice versa. Therefore a small value of $dist_T(i, j)$ indicates a large similarity between the trajectories.
As described by Porikli,[1] the extraction of features, like speed and orientation features, from the trajectory may be appropriate for shape-based recognition tasks. For application independent clustering tasks, speed and orientation features provide information which can be used to identify trajectories belonging to the same path. The calculation of the speed sequence $T\Delta_i$, i.e. the speed sequence of trajectory $T_i$, is given in Equation (4).

$$\begin{aligned} T\Delta_i &= [(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2]^{\frac{1}{2}}, n = 2..N \quad (4)\\ dist_S(i, j) &= dist_M(\theta_{T\Delta}^i, \theta_{T\Delta}^j) \quad (5)\end{aligned}$$

Similarly, Porikli proposed the use of an orientation sequence to identify similar shapes as in translations or changes in direction. The orientation sequence $T\phi_i$ in Equation (6).

$$\begin{aligned} T\phi_i &= tan^{-1}\frac{y_n - y_{n-1}}{x_n - x_{n-1}}, n = 2..N. \quad (6)\\ dist_O(i, j) &= dist_M(\theta_{T\phi}^i, \theta_{T\phi}^j) \quad (7)\end{aligned}$$

The distance metrics $dist_S(i,j)$ and $dist_O(i,j)$, using the speed and orientation sequences, are defined equivalently to $dist_T(i,j)$ (see Equations (5) and (7)).

The quality of the trajectories to be clustered has a great influence on the results of the clustering. Trajectories may be raw data trajectories or they may have been processed and smoothed. However, an additional smoothing step might lead to a loss of information. Instead of using the original pixel level data, trajectories are segmented according to image blocks of predefined sizes. In detail, the original image is segmented into blocks of size $w$. The original positions in the trajectory sequences are then mapped onto these blocks according to

$$\widetilde{T}_i = \left\lfloor \frac{T_i(n)}{w} \right\rfloor + 1, n = 1..N, \tag{8}$$

with $N$ being the length of trajectory $T_i$. The size of block size $w$ which is appropriate for the clustering task cannot be determined beforehand. It greatly depends on the smoothness of the initial trajectories and on the individual task, where a smaller or larger block size may by chance be appropriate. Since a single metric will never be able to fulfill all requirements and deliver a perfect clustering without making assumptions about the kind of trajectories or the recognition task, the solution is to choose a set of varying block sizes and calculate the clustering for all of them. Afterwards the results are summarized and shown to the user who has to select the result which meets his or her expectations best. This approach was chosen because persons are able to decide whether a clustering result is good or bad at practically one glance, due to his or her knowledge about the expected results. This decision does not even require the user to take a closer look at the individual clusters and possible errors, since the overall impression should usually be enough to chose a suitable block size. However, user studies about the presentation of the preliminary results and the number of results to be displayed are necessary and will be a topic of future research as discussed in Section 5. In the remainder of this work, segmented trajectories will be used only. $T_i$ and $\widetilde{T}_i$ will be used interchangeably and $dist_T$, $dist_S$ and $dist_O$ will therefore also refer to segmented trajectories.

### 3.1 Distance metric

In order to combine the advantages of all distance metrics mentioned in section 3, we propose two combined metrics $dist_{HMM}$ and $dist_{Com}$. $dist_{HMM}(i,j)$ uses HMM-based metrics only. Although each term of $dist_{HMM}$ uses trajectories which are based on the initial absolute position data, trajectories which are similarly shaped may be identified as belonging to the same cluster, even though their spatial displacement indicates their affiliation to different clusters. We therefore investigate $dist_{Com}(i,j)$ which combines the distance of motion trajectories with a spatial distance factor given as the euclidean distance of the trajectories starting and end points. The distance of starting and end points has the purpose to keep similar trajectories apart which run in opposite directions. For example, similar trajectories of a person walking in and out of a room. HMM-based representations tend to recognize these trajectories as belonging to one path. Although this may be in the users interest and represents one of the advantages of HMM-based representations, the assumption does not hold for all applications. For example, only persons entering a place or building may be of interest in some recognition tasks. The metrics are defined in the following way:

$$dist_{HMM}(i,j) = \frac{a * dist_T(i,j) + b * dist_S(i,j) + c * dist_O(i,j)}{a+b+c} \tag{9}$$

$$dist_{Com}(i,j) = \frac{a * dist_T(i,j) + b * dist_S(i,j) + c * dist_O(i,j) + d * (\|(S_i - S_j)\| + \|(E_i - E_j)\|)}{a+b+c+d} \tag{10}$$

with $S_i = (x_1, y_1)$ being the starting position and $E_i = (x_N, y_N)$ being the last position of trajectory $T_i$. Since there is no common normalization for HMM-based distances, each distance ($dist_{T|S|O}$) is normalized separately to allow for weighting and comparison in the combined metrics. Factors $a$, $b$, $c$ and $d$ were determined according to the behavior of their corresponding distance metrics. Speed and orientation of a trajectory are shape features appropriate for identifying similar shapes. Their influence for clustering paths is important, for example in situations like the one described in Figure 1. However, the influence of the original distance metric $dist_T(i,j)$ for the description and classification of trajectories is very important. Therefore, factors of $a = 6$, $b = 2$ and $c = 2$ were chosen and empirically proved to be suitable. Since the euclidean distance of start and end points is applied only to retain a certain amount of spatial correlation, it is weighted with factor $d = 2$. Choosing a larger value for $d$ soon leads to the clustering of trajectories according to their starting points, neglecting the pathways in-between.

### 3.2 Hierarchical clustering

A hierarchical complete linkage clustering is performed, merging the elements or clusters with the minimum distance in every step. The main obstacle in performing this clustering for the creation of ground truth data sets is the importance of a conservative clustering. If the result contains a larger number of clusters than necessary, this is less problematic than a result which shows trajectories belonging to different paths in one cluster. In the first case the user would have to add a few more labels for the additional clusters, however in the second case, he or she would have to manually identify and label every single trajectory in this cluster. Since the goal of this work is the simplification of the labeling task this is a very important factor.
The definition of a generic end criteria for the hierarchical clustering presents a major problem. A single specific criteria which provides a suitable set of clusters for any application does not exist. However, the merging of candidate clusters $c_u$ and $c_v$ is subject to the following conditions:

1. The mean intra-cluster distance of merged cluster $c_{\{uv\}}$ does not exceed the mean intra-cluster distance of all clusters before the merge operation is performed by more than a factor $f$. The factor is defined as a threshold function with scaling factor $\alpha$ which decreases as the number of iterations of the clustering increases.
$$f = \frac{\alpha}{(0.01 \cdot \#iteration + 1)}. \tag{11}$$
At the beginning of the clustering process, a larger difference in intra-cluster distance is more likely to be correct than after a series of iterations, therefore this condition aims to create clusters with a uniform perception of similarity. The exact shape of this threshold function is not to be determined beforehand. We propose to calculate the clustering for several values of $\alpha$ and let the user decide which one is suited best, similar to the block size $w$.

2. The merge operation must not increase intra-cluster distance too much. The merge operation is allowed if the increase in intra-cluster distance of cluster $c_{\{uv\}}$ is smaller than the mean increase caused by the last ten merging operations, weighted with factor $f$ as given in Equation (11). Similar to the first condition, this one incorporates the number of iterations and steadily decreases the allowed difference. The purpose of this condition is to restrict the growth of clusters as the time passes. If only Condition 1 is used, the growth of clusters will not actually be stopped, as long as all clusters exhibit a balanced intra-cluster distance.

Additional criteria, especially concerning the relation of inter-cluster distances were tested but found to be unsuitable in a generic approach. As discussed in the previous sections, the proposed technique incorporates two parameters, block size $w$ for segmentation of trajectories and percentage $\alpha$ for creating uniform clusters, which are not set to a specific value. However, the set of possible values for these parameters is not very large, since very large values are not reasonable for both parameters. Clusters are calculated for varying values of these parameters (e.g. $x = 5, 10, 20, 35$ and $w = 5, 10, 20, 40, 100$).

## 4. EVALUATION

In order to evaluate the proposed technique, a set of trajectories was extracted. The original video data was captured in a surveillance scenario at 15 frames per second. The detection and tracking of moving persons and objects was performed using an adapted background subtraction. The tracking was kept very simple, since the camera is static and the environment not subject to lighting changes. In our scenario, each time step of a trajectory $T_i(x, y)$ identifies the center of a moving object. Although the investigation of varying block sizes for the segmentation of trajectories is not part of this work, the evaluation revealed a block size of $w = 20$ to be suitable for this scenario. Parameter $\alpha$ was found to best suit our purposes with a value of 10. In order to provide a basis for comparison, the same values were used for all distance metrics.

Table 1 gives the results of the clustering capability in the surveillance scenario. A total of 241 trajectories was extracted. Manual labeling of the data identifies 17 paths, i.e. a perfect clustering returns these clusters. As already mentioned, our clustering approach is realized to prevent false positives rather than preventing false negatives, since the clustering results will be used for labeling training data. False positives belonging to a cluster must be identified manually by the user which is laborious. If not identified, these false positives weaken the classification results, since the classifier will be trained with wrong data labels.

In our experiments we compare four distance metrics, two simple metrics, $dist_{euclid}$ and $dist_T$, and the two combined metrics, $dist_{HMM}$ and $dist_{Com}$. We chose to compare the combined metrics with the euclidean distance and the HMM-based trajectory distance. Speed and orientation distances are not compared, because these metrics alone are not capable of fulfilling this task. The comparison of different test cases by Porikli[1] is the basis for this assumption. The two chosen simple metrics are subject to the several disadvantages, however, the comparison enables us to see the benefit of the combined metrics.

Table 1 gives an overview on the clustering results using the four different metrics. The euclidean distance was calculated on trajectories which were interpolated to provide trajectories of the same length. Although the metric is suitable for the creation of clusters with uniform direction, it fails if the paths are of varying length. Figure 2 shows an exemplary set of clusters. The paths shown in Figure 2 a) and f) differ significantly in their spatial distances of starting and end points. Figure 2 c), d) and e) display the major drawback of this distance metric. Two paths were identified as belonging to one cluster. Figure 3 displays an exemplary set of clusters created using metric $dist_T$. A major problem of this metric is the creation of clusters with non-uniform directions as indicated by the white circles which identify the starting points of trajectories. Figure 4 shows a subset of the clustering results obtained by the combined metric $dist_{HMM}$. Although the clusters still incorporate non-uniform directions, the separation of different paths works extremely well. The relatively high false positive rate is almost exclusively caused by non-uniform directions. Figure 5 shows an exemplary set of clusters created with the combined metric $dist_{com}$. Due to the additional factor containing the spatial distance between starting and end points, the clusters are uniform in regard to direction. In general the separation of different paths works well, however, Figure 5 f) shows two paths identified as one. Nevertheless, the false positive rate of this metric is significantly lower than the rate of the other compared metrics.

Considering our starting set of 241 trajectories, the number of 41 to 58 resulting clusters does not seem ideal. However, the data set contains approximately 20 trajectories which are deformed by tracking errors. These trajectories can be identified as belonging to a certain path by a human user but the clustering fails in all cases due to a large difference in position and shape.

Table 1. Experimental results of four discussed distance metrics. The simple metrics $dist_{euclid}$ and $dist_T$ exhibit major drawbacks. The combined metrics, however, provide a good separation of paths.

| | $dist_{euclid}$ | $dist_T$ | $dist_{HMM}$ | $dist_{Com}$ |
|---|---|---|---|---|
| # Clusters | 48 | 41 | 55 | 58 |
| False positives | 27 | 18 | 25 | 4 |

## 5. CONCLUSION

This work discussed the problem of creating ground truth data sets for video data. HMM-based distance metrics were proposed and evaluated on a surveillance scenario. The HMM-based metrics proved to be suitable for identifying similar paths. An evaluation found a combined metric of trajectory, orientation, speed and euclidean distance to best suit our general purpose. Using the proposed clustering technique greatly facilitates labeling of trajectories. However, to further facilitate the labeling task it will be necessary to add key frame visualizations to each cluster in order to provide the user with additional information. Although HMM-based representations are very well suited for the representation of trajectories, especially of varying length, they require comparatively large amounts of data, i.e. very short trajectories cannot be represented well. However, modern surveillance cameras provide frame rates which allow extraction of trajectories long enough for this technique. Since the application of trajectory clustering for creating ground truth data sets lies mainly in the surveillance recognition

tasks, this restriction does no longer present a severe obstacle.
Future work will focus on the extension of this technique in regard to adaptive parameters and user studies concerning the interaction aspect. A presentation of all clusters to the user will not be possible. Therefore, the development of graphical user interfaces which display clustering results for different parameter combinations requires the extraction of a discriminative subset of clusters first. It will also be necessary to identify the maximum number of parameter combinations that users are able to process. Additionally, user interaction techniques in regard to the identification of clustering mistakes will have to be investigated. The topic of video labeling has not yet been a major topic of research and further developments will be necessary to facilitate the creation of ground truth data sets for other applications than the identification of trajectories.

## ACKNOWLEDGMENTS

This work was developed within the Nexus project (collaborative research center/SFB 627), which is supported by the German Research Foundation (DFG).

## REFERENCES

[1] Porikli, F., "Trajectory distance metric using Hidden Markov Model based representation," in [*IEEE European Conference on Computer Vision, Pets Workshop*], (2004).

[2] Liao, T. W., "Clustering of time series data–a survey," *Pattern Recognition* **38**(11), 1857 – 1874 (2005).

[3] Hu, W., Xie, D., Fu, Z., Zeng, W., and Maybank, S., "Semantic-based surveillance video retrieval," *IEEE Transactions on Image Processing* **16**(4), 1168–1181 (2007).

[4] Makris, D. and Ellis, T., "Path detection in video surveillance," *Image and Vision Computing* **20**(12), 895 – 903 (2002).

[5] Morris, B. and Trivedi, M., "Learning, modeling, and classification of vehicle track patterns from live video," *IEEE Transactions on Intelligent Transportation Systems* **9**, 425–437 (Sept. 2008).

[6] Hervieu, A., Bouthemy, P., and Le Cadre, J.-P., "Video event classification and detection using 2d trajectories," in [*International Conference on Computer Vision Theory and Applications*], **2**, 158–166 (2008).

[7] Anjum, N. and Cavallaro, A., "Unsupervised fuzzy clustering for trajectory analysis," in [*IEEE International Conference on Image Processing*], **3**, III –213–III –216 (16 2007-Oct. 19 2007).

[8] Zhang, Z., Huang, K., and Tan, T., "Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes," in [*18th International Conference on Pattern Recognition*], **3**, 1135–1138 (2006).

[9] Jiang, F., Wu, Y., and Katsaggelos, A., "A dynamic hierarchical clustering method for trajectory-based unusual video event detection," *IEEE Transactions on Image Processing* **18**(4), 907–913 (2009).

[10] Juang, B. and Rabiner, L., "A probabilistic distance measure for Hidden Markov modeling," *AT&T Tech. J.* **64**(2), 391–408 (1985).

[11] Bashir, F., Khokhar, A., and Schonfeld, D., "View-invariant motion trajectory-based activity classification and recognition," *Multimedia Systems* **12**(1), 45–54 (2006).

[12] Xiang, T. and Gong, S., "Video behaviour profiling and abnormality detection without manual labelling," in [*Tenth IEEE International Conference on Computer Vision*], **2**, 1238–1245 Vol. 2 (2005).

[13] Zhong, H., Shi, J., and Visontai, M., "Detecting unusual activity in video," *IEEE Conference on Computer Vision and Pattern Recognition* **2**, 819–826 (2004).

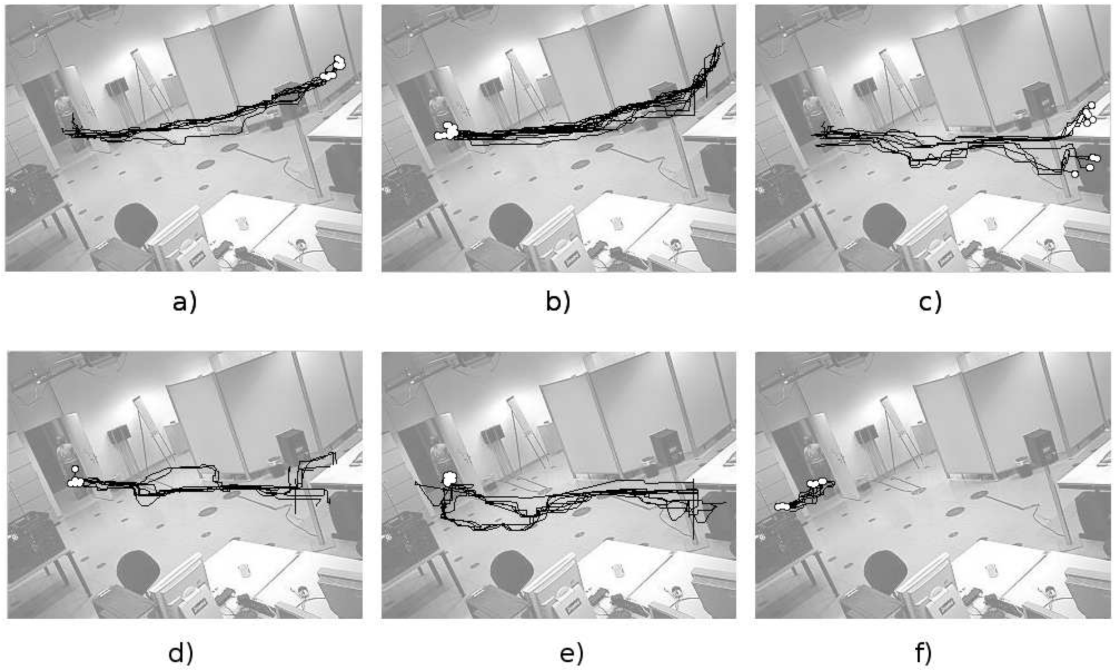

Figure 2. Exemplary set of clusters created using the euclidean distance $dist_{euclid}$. a) and b) show perfect clusters with trajectories of uniform direction in each cluster. c), d) and e) show clusters where two different paths were clustered by mistake. f) shows a cluster of trajectories, where the separation of trajectories according to direction fails, due to the relatively short distance between start and end point of the trajectories. The white circles indicate the starting point of each trajectory.

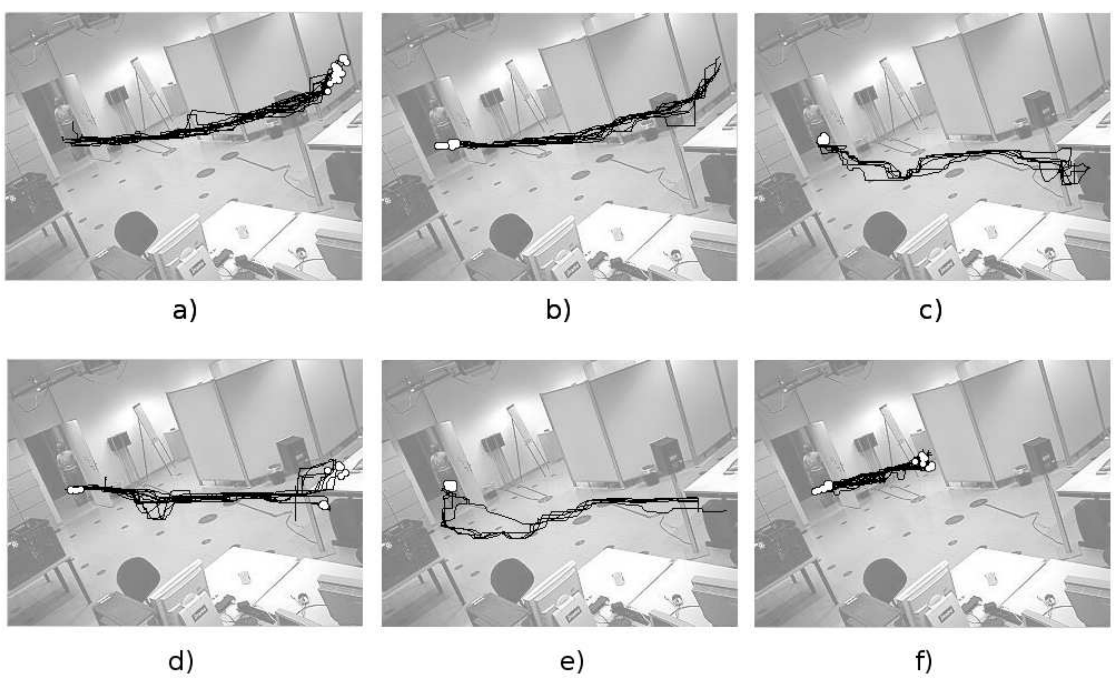

Figure 3. Exemplary set of clusters created using $dist_T$. a), b) and c) show clusters of trajectories which were perfectly separated according to their shape. d) and f), however, show clusters of non-uniform direction. e) shows a cluster where the separation of trajectories according to the shape fails. This single metric alone is not suitable for distinguishing paths according to shape.

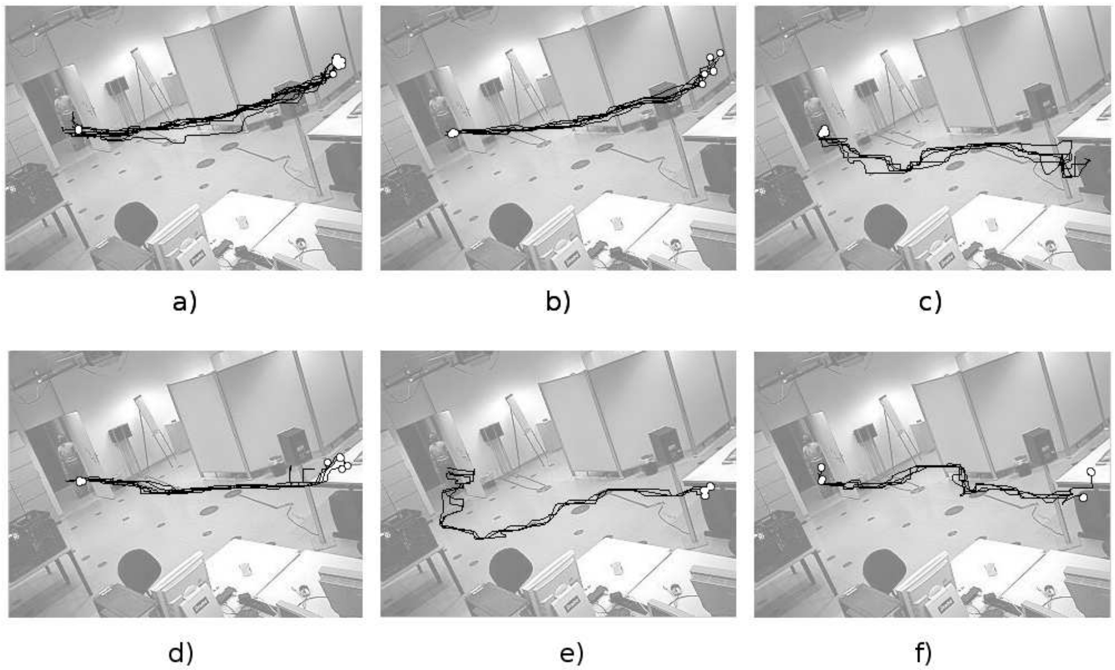


Figure 4. Exemplary set of clusters created using the combined metric $dist_{HMM}$. a), c) and e) show clusters with trajectories belonging to the same path but with non-uniform directions, indicated by the starting points depicted as white circles. b), d) and f) show perfect clusters which clearly extract different shapes.

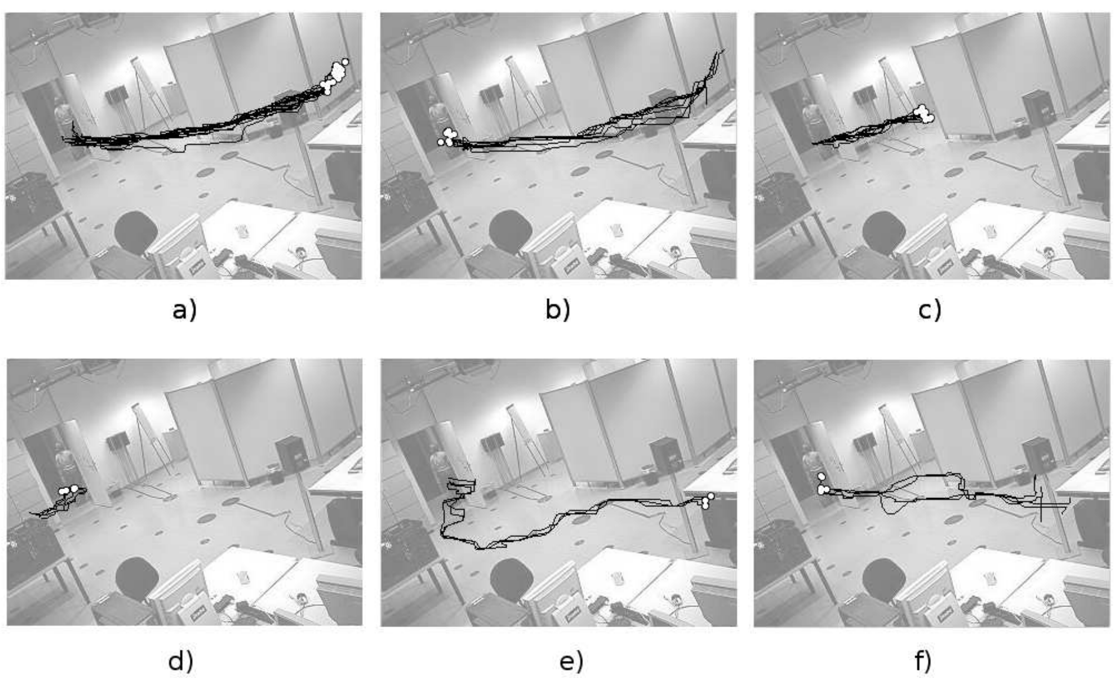


Figure 5. Exemplary set of clusters created using the combined metric $dist_{Com}$. a), b), c), d) and e) show perfect clusters with uniform shape and directions. f), however, shows trajectories of different shapes which were clustered by mistake.