2019 IEEE Intelligent Vehicles Symposium (IV)
Paris, France. June 9-12, 2019

Pedestrian Group Detection in Shared Space

Hao Cheng[1], Yao Li[2], Monika Sester[1]

Abstract— **In shared space, pedestrians are often found walking in groups and behaving differently than individual pedestrians. However, automatically detecting pedestrian groups with high accuracy is not trivial given the dynamic environment and interactions in mixed traffic. Instead of tedious manual work and in order to cope with large scales of data, we propose a time–sequence Density–Based Spatial Clustering of Applications with Noise (DBSCAN) for pedestrian group detection. It is based on coexisting time and Euclidean distance between pedestrians. Our approach outputs reliable results with high IoU values. It can be easily adapted to other groups, e.g., cyclists and animals. In addition to individual behavior, the output data with differentiation of group behavior can be used in further studies in intent detection and motion prediction.**

I. Introduction

In crowded areas, pedestrians are often found walking in groups, with couples, families, or friends, and they usually respond to other traffic participants differently in comparison to individual pedestrians [1], [2]. In the classic Social Force Model, group members have the so-called attraction effect on each other [3]. For instance, pedestrians from one group tend to walk at the same speed and maintain a certain distance between each other [4]. They have to synchronize their speed and distance for communication and visibility between each other [5]. In shared space, where traffic regulations are weakened or even absent and direct interactions among mixed traffic are encouraged [6], pedestrian groups can be commonly observed to be more dominant in response to other encounters, e.g., single pedestrians, cyclists or cars [5].

Whereas, the definition for *group* varies in different domains and there are similar concepts, such as *flock*, *swarm*, *moving cluster*, *herd*, and *convoy*; they differ from the way how space and time constrain entities that stay close to each other [7], [8]. In the context of pedestrians (or road users) in shared space, the definition follows [2] that a pedestrian group normally moves in a linear formation or V-like pattern with close distance maintained by at least two members over a certain amount of time. Different from considering a complex socialogical collective behavior [9], here, the concept of group is purely focused on the similar locomotion shared by group members, i.e, moving close to each other in space and time.

Among individual road users, clustering pedestrian groups is essential for intent detection for autonomous driving and traffic planning. In motion modeling and behavioral prediction, many models treat group members as independent individuals [10], [11], rather than as one connected component. Close interaction between group members may erroneously be interpreted as a risk for collision. This will, consequently, degrade the performance and may even lead to failure in real–world traffic situations. Hence, differentiating group behavior from individual behavior is promising to improve the prediction for intent detection [4], [12].

However, how to automatically detect a pedestrian group with correct group members is not a trivial task in shared space. Firstly, the size of pedestrian groups differs depending on the given space and density. For example, on a university campus, students and colleagues are often seen walking in groups to class or to a cafeteria. Secondly, even a single pedestrian group can gain or lose members on the fly due to interactions with other single road users or groups [13], the so-called splitting and merging phenomenon. In addition, mixed traffic participants (pedestrians, cyclists, and vehicles) coexist with each other. The variety of transport modes makes interactions even more complicated and the distances between them change dynamically in time. Hence, a group detector has to not only correctly detect group members, but also exclude other road users even if they have close interactions.

Most of the early works in group detection were carried out by manual observations based on the maintained distance and the coexisting time between two or more members [1], [2], or describing specific movement patterns [14], [15]. It is expensive to scale up to large amounts of pedestrians.

Recently, hierarchical clustering approaches have been used to detect pedestrian groups automatically. Sandıkcı et al. employed a bottom-up hierarchical clustering to detect human groups based on positional, velocity, and directional similarities. The inter–cluster similarity between two groups is computed by maximum–linkage [16]. Ge et al. proposed a bottom-up hierarchical clustering approach based on pairwise Hausdorff distance to discover small groups in pedestrian crowds [17]. Similarly, Kjærgaard proposed a bottom-up hierarchical clustering based on a data fusion method with multiple sensors. It uses a centroid–linkage to compute the distance between two objects featured by different sensor data [18]. However, the aforementioned approaches require a cut-off for the hierarchical clustering structure defined by the linkage criteria and similarity metrics. They did not provide clues for how to optimize those criteria.

Another branch of studies use graph–based approaches

[1] Institute of Cartography and Geoinformatics, Leibniz University Hannover, Appelstrasse 9A, Hannover 30167, Germany hao.cheng@ikg.uni-hannover.de, monika.sester@ikg.uni-hannover.de

[2] Faculty of Civil Engineering and Geodesy, Leibniz University Hannover, Callinstrasse 34, Hannover 30167, Germany y.li@stud.uni-hannover.de

The authors cordially thank the funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 227198829/GRK1931.

978-1-7281-0560-4/19/$31.00 ©2019 IEEE

to group similar movement patterns, which is close to the method that we propose in this paper for pedestrian group detection. For example, Kuntzsch and Bohn implemented an algorithm to group moving point objects (MPOs) traveling together [7]. It uses a binary relation to cluster entities that are directly or indirectly connected to each other at a given time instance. The indirect connectivity allows transitivity among MPOs. Hence, a group is treated as a connected component. Meanwhile, a back–tracking is used to check whether a set of MPOs belong to a group over a certain amount of time. The same idea was employed by Buchin et al., a group is a connected component that the entities are connected directly or in directly within a spatial parameter and span over a certain length of duration [8].

Contributions: Inspired by the idea above, in this paper, we propose a density–based group detector that takes trajectories extracted using a deep learning tracker from videos recorded in a shared space. Different from [17], [19] that directly use extracted trajectories from videos, we follow the workflow in [20]. Before grouping, the extracted trajectories are firstly pre-processed by camera calibration and transformation from video pixel coordinates to real–world coordinates on a 2D plane with bird's–eye view, in order to achieve a good accuracy. The main contributions of this paper are as follows.

- It provides a time–sequence DBSCAN (Density–Based Spatial Clustering of Applications with Noise, [21]) cluster to automatically detect group members based on their trajectories extracted from a deep learning tracker.
- The group detector is not limited with respect to the size of input data. It can efficiently and effectively process large-scale datasets.
- IoU (Interaction over Union) is used to evaluate the performance of the proposed detector. It considers both the size and members of the detected groups against ground truth labeled by human experts.
- We also prove that distance and coexisting time ratio are two important parameters to determine if two users belong to one group. With hyper–parameter searching, we propose the optimal values based on our datasets.

II. Methodology

In this paper, we carry out a complete pipeline consisting of a static camera for data acquisition, followed by a deep learning trajectory tracker proposed by [22] using recorded video data. A projective transformation that transfers the extracted trajectories from the camera view to bird's–eye view into a 2D image coordinate system is leveraged for data pre-processing. The transferred 2D trajectories are used as input for a time–sequence DBSCAN cluster to detect pedestrian groups. In the end, the detected groups with corresponding group members are compared with the ground truth, that is, manually labeled by two human experts by observing the recorded videos. Fig. 1 depicts the pipeline.

A. Data Acquisition

The trajectory data were acquired by a static camera positioned with an elevation of approximately 20 meters above ground level in a German university area. Videos were recorded at $30\,\mathrm{fps}$. Both elevated position of camera and resolution ensured that no privacy issues arise. Before trajectory extraction, the videos were calibrated by distortion removal [23] using OpenCV[1].

B. Trajectory Extraction

To acquire large amount of real–world trajectories in shared space, a tracker pre-trained using labeled datasets (i.e., ILSVRC[2]) for arbitrary moving objects was applied. The tracker was created by Bertinetto et al. [22] using fully-convolutional Siamese networks. It was recently shown to be competitive with other trackers, providing reliable results at a real–world frame rate. With a slight modification, the tracker was used to track pedestrians, cyclists, and vehicles from the acquired videos. In order to avoid detection failure, the initial position of the target was marked manually. For more details about the tracker, see [22].

The initial videos were down–sampled to $2\,\mathrm{fps}$. In other words, every 15th frame was selected as input for trajectory extraction. This is because that the offset of every two consecutive frames for a road user is relatively small, especially for pedestrians who travel at an average speed of $1.41\,\mathrm{m/s}$ (see Fig. 2). The consecutive offset is normally less than 0.05 meters. Furthermore, the extracted trajectories were also corrected by two human experts manually when errors occurred, to achieve high accuracy.

C. Data Pre-processing

In order to get realistic trajectories on a 2D plane, we first applied a projective transformation and then a moving average filter to pre-process the acquired trajectories.

Due to the fact that the camera was not positioned directly on top of the shared space, trajectories were skewed, especially in the longitudinal direction. We applied a projective transformation [23] to reduce the distortion caused by the camera angle, which was not perpendicular to ground level. The transformation was also implemented by OpenCV.

The moving average filter uses a one–step window from both sides of each point for a given trajectory, except the first and last points. Each of those points are updated by the mean value of the previous, itself, and the next point to reduce fluctuations.

D. Group Detection

After the pre-processing, 2D trajectories (x_t, y_t) are used for group detection. t denotes the time step and each time step lasts 0.5 seconds. Similar to [18], the detector is mainly based on two factors: coexisting time and Euclidean distance between two pedestrians in the 2D plane.

Differently from [18], we propose to use DBSCAN [21] to cluster pedestrians into groups. Firstly, there is no need to decide the number of clusters as many other clustering algorithms have to do (e.g., K-means [24]). This is very important since in shared space the number of groups may

[1] https://opencv.org/

[2] http://www.image-net.org/challenges/LSVRC

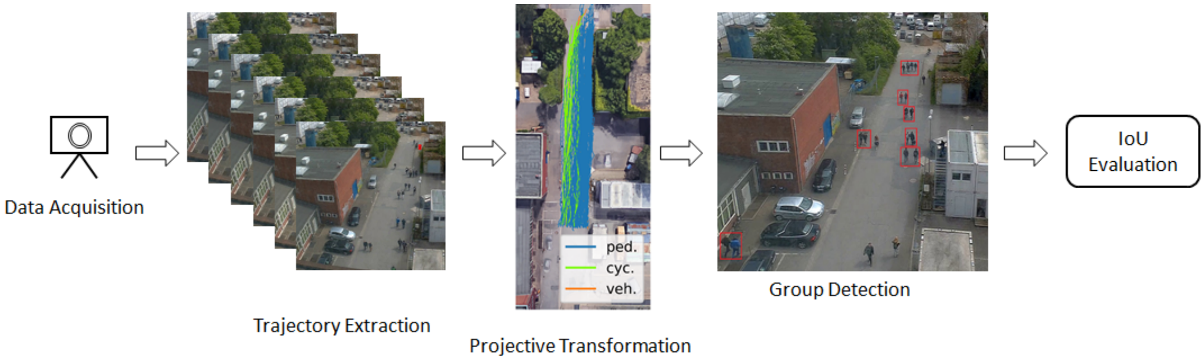


Fig. 1: Framework for group detection

change over time. Secondly, the maximum distance within a group is not measured from neighborhood points to one central point. Instead, it is measured from neighborhood points to core points, and the number of core points can be more than two. The transitivity allows a large distance between two members via indirect connection. It also helps to rule out noisy points that are not close to any core points.

Moreover, in terms of computational runtime, DBSCAN has an average runtime complexity of $O(n \log n)$ [21], which is much faster than K-means (runtime complexity $O(n^3)$, [25]). For pedestrian group detection in a shared space, n is relatively small. At each frame, there are less than 100 road users coexisting in the shared space depicted by Fig. 1.

However, a simple DBSCAN can only detect pedestrian groups for a given time step and one time step is not sufficient to decide whether multiple pedestrians belong to one group. They may be close to each other by coincidence. To solve this problem, we propose a time–sequence DBSCAN. By definition, we apply DBSCAN at each time step for a given pedestrian, which we call *ego* pedestrian. We store all the pedestrians who stay in the same cluster with the ego pedestrian for a certain length of time. Only if their coexisting time ratio r exceeds a predefined threshold, they are assigned to the same group. r is the fraction of T' and T, where T' is the overlapping time steps that the non-ego and ego users stay in the same cluster and T is the union of the distinct time steps covered by them. Equations (1) and (2) describe the detection process. ϵ is the maximum Euclidean distance from the neighborhood point q to the core point set P. MinPts is the minimum number of points in the point set P, including the non-core point. The minimum size (MinPts) is set to two since the smallest pedestrian group is two.

$$N_\epsilon(P) := \{q \in D | \text{dist.}(P, q) \leq \epsilon\}, \text{MinPts} = 2 \tag{1}$$

$$r = \frac{T'}{T},\ T' = \text{Count}[\text{Cluster(ego, non-ego)}_t] \tag{2}$$

During detection, the detector loops over all pedestrians as ego pedestrian and output all the group candidates. Only group members symmetrically connected are clustered together. Here, symmetry means group members are connected no matter which member is the ego pedestrian. For example, during the loop over ego user a, b, and c, set (a, b), (b, a), and (c, b) are respective group candidates in each loop. Only a and b are clustered into a group since they are connected in both directions. However, the candidate (c, b) is discarded for their asymmetric connection.

There are two hyper–parameters in the time–sequence DBSCAN: r and ϵ. We keep one parameter fixed and search for the other to determine the optimal values based on the real–world data. More details can be seen in Section IV.

E. Evaluation Metrics and Baselines

To evaluate the performance for our proposed approach, we use the IoU value to measure how close the predicted groups are to the true groups labeled by two human experts. The human experts use the observation of the duration of close interaction and even explicit communication (talking) between multiple pedestrians to distinguish pedestrian groups from individuals. The shape of a pedestrian group often changes when interaction occurs.

IoU is the intersection divided by union, which is commonly used for evaluating object detection in computer vision. In terms of pedestrian groups, the intersection consists of the correctly predicted members for a group. The union is the combination of all the members either in the predicted or in the true group. Both false positive and false negative members will lead to a penalty for the IoU value.

$$\text{IoU} = \frac{G_{\text{pred}} \cap G_{\text{true}}}{G_{\text{pred}} \cup G_{\text{true}}} \tag{3}$$

We compute the IoU value using Equation 3 for each road user and report the mean and standard deviation across all the users in the datasets. We also report the fraction of single users that are correctly predicted as single users.

Three basic approaches for group detection based on coexisting time and/or distance are used as baselines for comparison with the time–sequence DBSCAN.

- *Model-1* is merely based on the overlapping coexisting time, without consideration of distance threshold. If two road users are existing at the same time steps, they will be clustered into the same group, no matter how far they are from each other.

- *Model-2* merely considers the distance between two pedestrians' trajectories as long as they have coexisted in at least one time step. Because there is no consideration of the coexisting time ratio, we cannot calculate adequate pairwise Euclidean distances between them. Instead, we calculate the longest Euclidean distance from one user's trajectory to the other's trajectory, which is called Hausdorff distance [26].
- *Model-3* considers both the coexisting time ratio and the distance that is centralized by the ego user. It is a hierarchical combination of Model-1 and Model-2; Model-3 clusters pedestrians using Hausdorff distance threshold based on the output of Model-1, which is similar to the approach for detecting small groups in crowds in [17].
- *Model-4* is the proposed model time–sequence DBSCAN.

III. Datasets

We collected three datasets (D1, D2, and D3) from a German university area at different daytimes, especially during class breaks with high traffic flow density. They last 2256, 998, and 402 time steps, respectively. The area is a shared street (width: $12.5\,\mathrm{m}$, length: $74.5\,\mathrm{m}$, see Fig. 1) for mixed road users. Most of them are pedestrians and the rest are several cyclists and a few vehicles. Pedestrians walk alone or in groups containing members from two to nine. Table I gives the details. We use the letter G and the corresponding number to denote groups and their size. Single users are denoted as S.

TABLE I: Datasets with group allocation of ground truth

No.	ped	cyc	veh	S	G2	G3	G4	G5	G6	G9
D1	220	25	9	86	31	7	5	2	2	1
D2	129	15	4	47	27[1]	7	2	-	-	-
D3	35	2	0	12	4	5	-	-	-	-

[1] One group formed by a pedestrian and a walking cyclist with his bicycle.

Fig. 2 shows the velocity distributions for different user types. Pedestrians walk at a relatively constant speed (M: $1.41\,\mathrm{m/s}$, std.: $0.44\,\mathrm{m/s}$). Cyclists travel at a much faster speed (M: $3.89\,\mathrm{m/s}$, std.: $1.25\,\mathrm{m/s}$) than pedestrians. Vehicles travel at a slightly faster speed than pedestrians (M: $1.89\,\mathrm{m/s}$, std.: $2.03\,\mathrm{m/s}$), but with large speed variation.

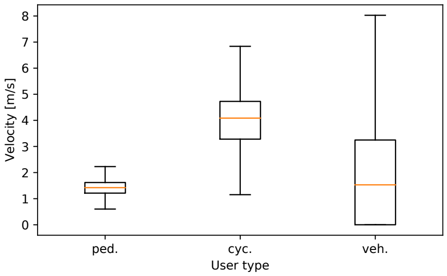


Fig. 2: The distribution of velocity regarding user type

In the datasets, the maximum distance maintained by a group at each time step can also be observed. With the consideration of transitive connection, the maximum distance is retrieved using a Minimum Spanning Tree (MST) algorithm for each group at each time step. The longest edge connecting group members in the MST at one time step is defined as the corresponding maximum distance.

One interesting observation of the manually labeled group data is that the maximum distance maintained by group members is influenced by the size of groups; in general, larger groups (from G2 to G4) tend to have larger distance between group members. Fig. 3 shows the intra distances for different group sizes. Pairwise Mann–Whitney U tests demonstrate significant differences ($p << 0.01$) of maximum distance across different sizes. However, the value of the maximum distance jumps to $3.5\,\mathrm{m}$ in G5, drops in G6, but increases in G9 again. This pattern may not be justified by the small numbers of groups G5 to G9 statistically.

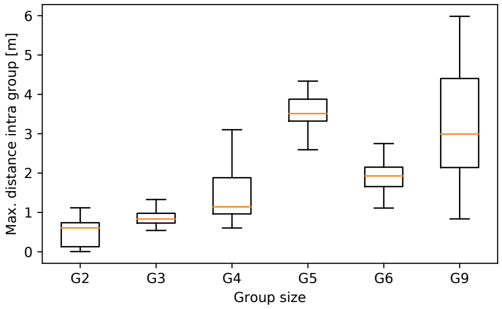


Fig. 3: Max. intra group distances for different group sizes

Additionally, these datasets were extended with estimated categorical demographic attributes based on observation, including gender, age range, cellphone usage, talking, and extra luggage. For privacy protection, all the video data were digitized into trajectory data so that the identity of the road users cannot be recovered. We believe that they can assist to investigate more about road users' behavior. For the current study, however, this information has not been exploited yet. The data is available at `https://doi.org/10.25835/0094187`.

IV. Experiments and Results

A. Hyper–Parameter Searching

To find the optimal values for the maximum distance ϵ and the minimum coexisting time ratio r between two members for the DBSCAN detector, we keep one parameter fixed and search for the other based on the IoU values for group detection performance. Heatmaps from Fig. 4 show the performances for each model configured by different ϵ and r. A darker blue color is associated with a higher IoU value.

Model-1 is only based on coexisting time ratio. Hence, Model-1 only demonstrates the performance difference regarding r. The higher r is, the better the performance is. However, when the coexisting time ratio is too high ($r > 0.9$), its performance decreases. This is because that even members from one group may enter or leave the camera view at different time steps. In general, it has a good performance when r is 0.9. This might due to the special layout of the shared space—both sides are blocked by buildings and constructions. There are many homogeneous longitudinal walking but few cross walking trajectories.

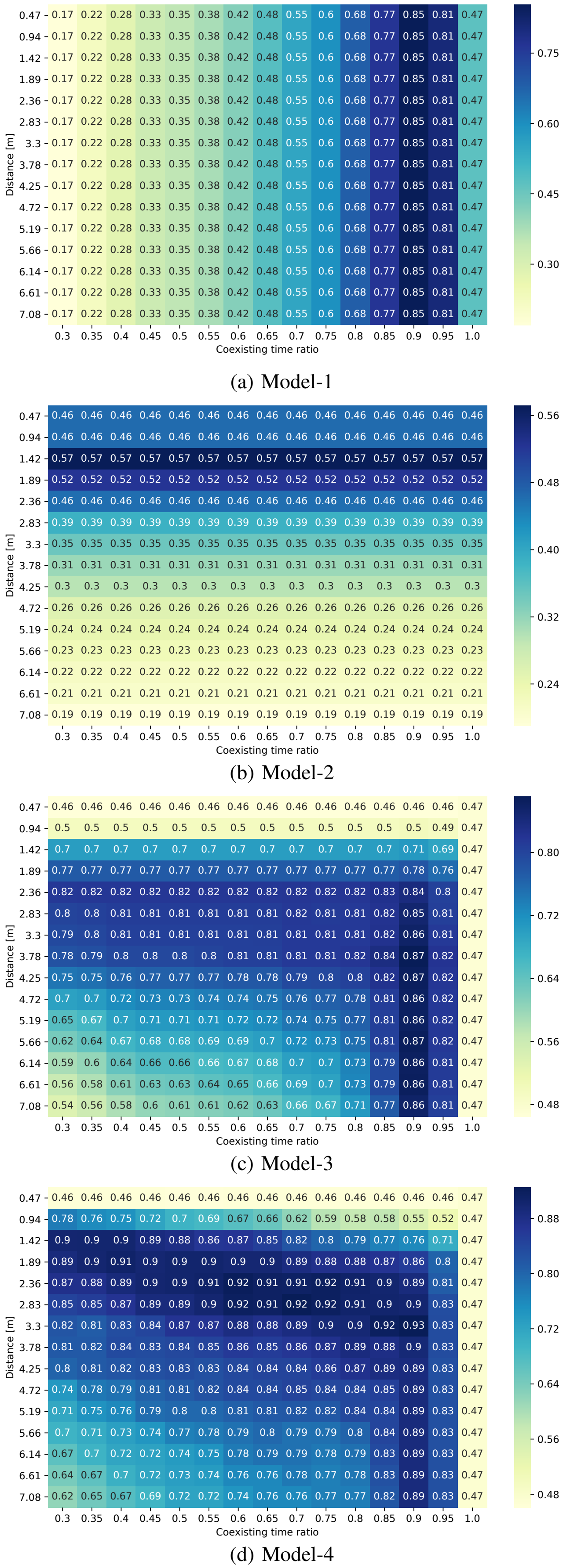
On the other hand, Model-2 only considers the Hausdorff distance maintained by two road users, regardless of their coexisting time ratio as long as they have coexisted in at least one time step. It happens very often that two non-group members walk closely in a short time and in similar routes. They are wrongly clustered as a group.

As Model-3 combines Model-1 and Model-2, the performance now depends on both r and ϵ. There is a weak correlation coefficient (Pearson's $R = 0.16$) between r and ϵ. When ϵ is fixed at $2.36\,\mathrm{m}$, the performance increases as r increases up to 0.9, but decreases with too saturated r values. This indicates that a minimum coexisting time ratio is necessary to determine if two users are in the same group. In addition, when the distance increases with r fixed at 0.9, the performance also increases. However, if ϵ is too large, then many neighbourhood users with large distances will be wrongly clustered into a group.

Model-4 is the time–sequence DBSCAN. In comparison with Model-3, the correlation coefficient (Pearson's $R = 0.38$) shows a stronger positive dependence between r and ϵ for maintaining good performance. This means that when r is small, ϵ should also be small to rule out neighbourhood users that are far away from being clustered as the group members. On the other hand, when ϵ is large, a high coexisting time ratio is required to guarantee that two users with a relatively large distance are in the same group. One interesting observation from Model-4 is that, even a small pair of values for coexisting time ratio ($r = 0.4$) and distance ($\epsilon = 1.89\,\mathrm{m}$) gives good performance ($\mathrm{IoU} = 0.91$) compared with Model-3. This attribute of DBSCAN gives Model-4 the possibility to correctly predict a group even if the members shortly split (small r value) and gather together again. In addition, DBSCAN measures the distance from neighborhood points to the set of core points, which allows indirect connection (transitivity). Instead, Model-3 uses direct connection from neighborhood points to one central point (the ego user). Therefore, in order to achieve a relatively close performance, Model-3 requires a larger distance. On the other hand, a larger distance also introduces errors such as clustering far neighborhood points which are not members of the group.

B. Results for Clustering Models

Based on the hyper–parameter searching aforementioned, we report the performance of each model on each dataset. Table II shows the results. All the values are averaged over all pedestrians. The time–sequence DBSCAN outputs very reliable results for group detection. Its IoU values and accuracy for single pedestrians are above 0.9 and 0.95 respectively for all datasets. Compared with other models, it outperforms all the others on D1 and D2 and correctly predicts all the groups in the relatively simple D3. Meanwhile, model-1 and model-3 also correctly predict all the groups in D3.



(a) Model-1

(b) Model-2

(c) Model-3

(d) Model-4

Fig. 4: Heatmap for hyper–parameters regarding IoU value

TABLE II: Model performance summary

Data	D1			D2			D3		
Measure	IoU	Std.	Accu. for ind.	IoU	Std.	Accu. for ind.	IoU	Std.	Accu. for ind.
Model-1	0.85	0.30	0.88	0.89	0.29	**0.99**	**1.00**	**0**	**1.00**
Model-2	0.57	0.43	0.81	0.54	0.45	0.25	0.78	0.36	0.64
Model-3	0.87	0.28	0.94	0.88	0.31	0.91	**1.00**	**0**	**1.00**
Model-4	**0.93**	**0.23**	**0.97**	**0.92**	**0.23**	**0.99**	**1.00**	**0**	**1.00**

IoU is the value for intersection of the union, *Std.* is the standard deviation for IoU. *Accu. for ind.* is the accuracy for correctly predicted single pedestrians who walk alone. Best values are in boldface

C. Test on Other Datasets

We run the time–sequence DBSCAN cluster on the BIWI benchmark datasets [27] to further test its utility. Table III denotes the statistics of single pedestrians and groups in the datasets. The ETH dataset has more groups with different sizes than the HOTEL dataset. Compared with the ground truth for groups provided by the data, the prediction performance for the cluster is very close to the performance on our datasets, see Fig. 5. Even on the relatively complicated ETH dataset, the IoU value reaches 0.85 and the accuracy for single pedestrians maintains at 0.90.

TABLE III: BIWI dataset

Dataset	ped	S	G2	G3	G4	G5	G6
ETH	360	192	38	10	7	3	3
HOTEL	390	305	38	3	-	-	-

In addition, we use the time–sequence DBSCAN cluster to group road users on two unlabeled datasets with different density—Dalian University of Technology (DUT) dataset [28] and Hamburg Bergedorf Station dataset (HBS) [29]. Due to the lack of ground truth, the IoU values cannot be calculated. Whereas, from the snapshots (see Fig. 6 and 7), road users moving close to each other in time and space are clustered into groups in both dense and sparse spaces.

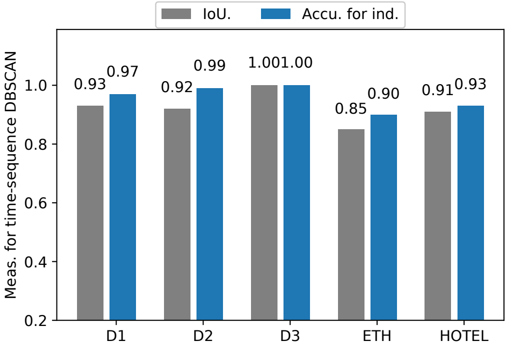


Fig. 5: The Performance of the time–sequence DBSCAN on different datasets

Table IV lists the hyper–parameters and runtime on a laptop with an Intel Core i7 8750H Processor (2.2 GHz) and 16 GB DDR 4 for the time–sequence DBSCAN on the datasets with different density. It can be seen that the model can be executed very fast. Even on the very dense dataset DUT, processing one user can be done within half a second.

TABLE IV: hyper–parameters and runtime for the time–sequence DBSCAN on different datasets

Parameters	ETH	HOTEL	DUT	HBS
ϵ [m]	1.5	1.0	1.5	3.0
r	0.85	0.90	0.85	0.85
runtime [ms]	10	6	396	15
density [# users / frame]	6.2	5.6	115.0	12.0

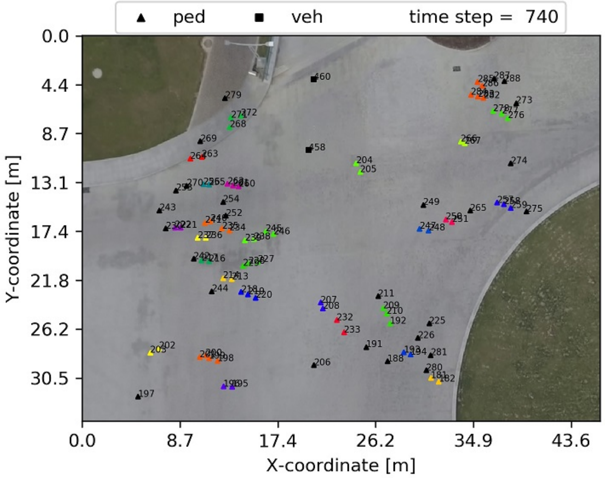


Fig. 6: A shared space in Dalian University of Technology. Groups are color coded and single road users are in black

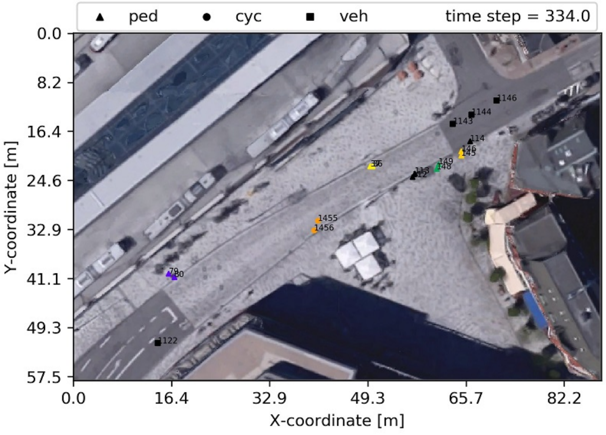


Fig. 7: A shared space at Hamburg Bergedorf Station. Groups are color coded and single road users are in black (Background image: Imagery ©2017 Google, Map data ©2017 GeoBasis–DE/BKG(©2009), Google)

D. Incorporating Grouping for Trajectory Prediction

We use an ablation study to investigate the contribution of the time–sequence DBSCAN to an LSTM-based model (i.e., LSTM-PDM) for trajectory prediction in shared space. More details of the implementation for LSTM-PDM can be found in [30]. The model was trained 100 epochs using D1 and D2, and tested on D3. With the consideration of grouping, during training, we found that the model converges much faster than without grouping. The predicted trajectories are compared with ground truth trajectories by Euclidean and Hausdorff distances, heading error, and speed deviation. Table V lists the measurements in prediction for six time steps. A lower value denotes a better result. Overall, the performance for the model improves by 15 % in terms of Euclidean and Hausdorff distances, 29 % in terms of heading error, and 11 % in terms of speed deviation. It proves that the grouping mechanism can be used to boost the performance for trajectory detection.

TABLE V: Trajectory prediction using LSTM-PDM [30] with and without time–sequence DBSCAN grouping.

Metrics	Euclidean dist. [m]	Hausdorff dist. [m]	Heading error	speed dev. [m/s]
with grouping	**0.95**	**1.06**	**34.3**	**0.32**
without grouping	1.13	1.27	48.4	0.36

V. Discussion

In this section, we discuss some interesting parts for optimizing the hyper–parameters ϵ and r and potential aspects for improvement.

According to the experiments and results, we have shown that coexisting time ratio and distance are two important indicators for group detection. In general, if pedestrians are members from the same group, they are likely to be walking together with a distance not much larger than 3 meters and 85 % of the time such a distance threshold is maintained (see Table IV). However, the optimal values for r and ϵ are highly depended on the dynamics the data contain. For example, if many groups have to split during interactions, r and ϵ may need to be set to smaller values so that temporary separation between group members can be tolerated by the detector. Other complicated situations, for example, waiting for group members or meeting a friend in the middle of a road, etc., will also lead to difficulties for the time–sequence DBSCAN detector with such proposed hyper–parameters.

Fig. 8 demonstrates a waiting scenario. A member (user-id 255) was first waited for by other five members for up to 18 seconds, and then the user merged into that group at $t = 819.5$ [s]. But the human experts still label him as part of the group, depicted by Fig. 8. Nevertheless, the coexisting time ratio is too low for the detector to cluster that user into the group. It can only correctly predict that the other five users are members in the same group.

There are also ambiguous situations even for human experts to decide whether two members are belonging to one group. For example, in Fig. 9, user 116 and user 237 met in the middle of the street at $t = 983.0$ [s]. The detector still clusters them as individual road users even though they stayed close to each other for up to 10.5 seconds. In this case, they may be friends in real life, but they should not be clustered into the same group due to their very different trajectories in opposite directions.

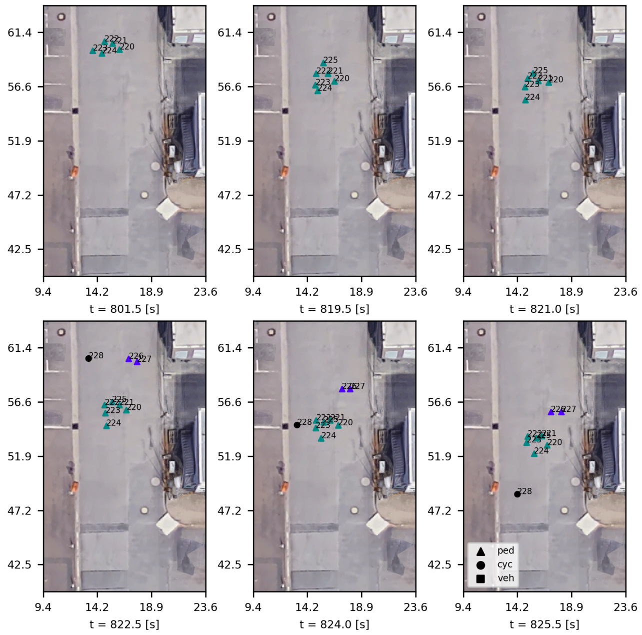


Fig. 8: Group forming in a long waiting time

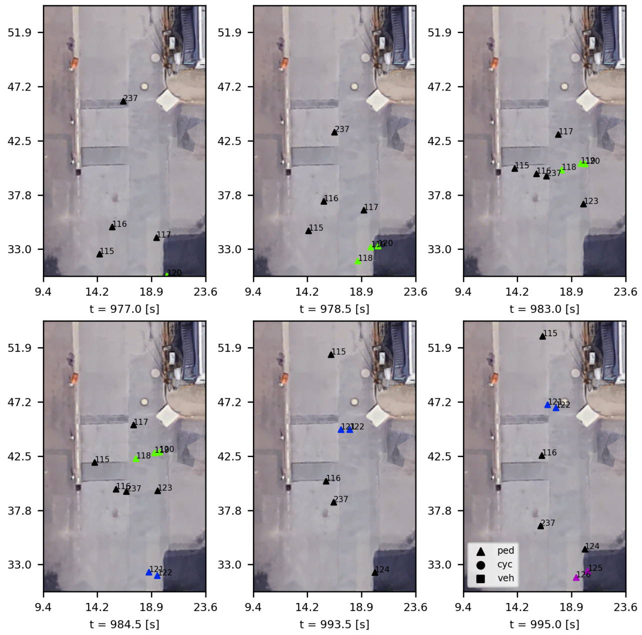


Fig. 9: Two single users meet with each other

From both Fig. 8 and 9, it is difficult to decide whether two users are group members, especially, when they have social ties and stay close to each other intentionally. The social relation may impact their behavior [2], [31] in shared space. However, in this study, the detection of group members is more focused on group behavior, whose movements demonstrate similarity. They can be treated as an integrated agent (even without social tie in real life), which can be used to enhance trajectory prediction (see Section IV-D).

One drawback of the time–sequence DBSCAN detector is that, at this point, it does not adapt to the maximum distance ϵ to group size, which also shows significant impact (see Section III). It might be difficult for this detector to cluster large groups with relatively loose connection between some group members.

VI. Conclusions and Future Work

In conclusion, in this paper, we propose to use a time–sequence DBSCAN cluster for group detection on the data collected from a shared space with mixed traffic. It is based on coexisting time and Euclidean distance to cluster pedestrians into groups with reliable performance. It can automatically label group members demonstrating similar behavior. With low runtime complexity, the group detector can be utilized to process large-scale datasets. We prove that, with the consideration of grouping, the performance for trajectory prediction can be improved. We also provide an overall framework for data acquisition, pre-processing, and group detection. It can be easily adapted to other types of groups, such as cyclists or animals.

In future work, we will apply the time–sequence DBSCAN detector to label group behavior for large datasets regarding shared space, and on other types of users. Those data will be leveraged to train and generalize deep learning models for trajectory prediction in shared space [12], [30] with the consideration of group behavior. The influence of additional parameters (e.g. gender, age) on the group behaviour will also be studied. The data is open source so that they can be further utilized for other research purposes.

References

[1] A. F. Aveni, “The not-so-lonely crowd: Friendship groups in collective behavior,” *Sociometry*, pp. 96–99, 1977.

[2] M. Moussaïd, N. Perozo, S. Garnier, D. Helbing, and G. Theraulaz, “The walking behaviour of pedestrian social groups and its impact on crowd dynamics,” *PloS one*, vol. 5, no. 4, p. e10047, 2010.

[3] D. Helbing and P. Molnar, “Social force model for pedestrian dynamics,” *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[4] K. Yamaguchi, A. C. Berg, L. E. Ortiz, and T. L. Berg, “Who are you with and where are you going?” in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1345–1352.

[5] N. Rinke, C. Schiermeyer, F. Pascucci, V. Berkhahn, and B. Friedrich, “A multi-layer social force approach to model interactions in shared spaces using collision prediction,” *Transportation Research Procedia*, vol. 25, pp. 1249–1267, 2017.

[6] S. Reid, *DfT shared space project stage 1: appraisal of shared space*. MVA Consultancy, 2009.

[7] C. Kuntzsch and A. Bohn, “A framework for on-line detection of custom group movement patterns,” in *Progress in Location-Based Services*. Springer, 2013, pp. 91–107.

[8] K. Buchin, M. Buchin, M. van Kreveld, B. Speckmann, and F. Staals, “Trajectory grouping structure,” in *Workshop on Algorithms and Data Structures*. Springer, 2013, pp. 219–230.

[9] D. T. Campbell, “Common fate, similarity, and other indices of the status of aggregates of persons as social entities,” *Behavioral science*, vol. 3, no. 1, pp. 14–25, 1958.

[10] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, “Social lstm: Human trajectory prediction in crowded spaces,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[11] A. Vemula, K. Muelling, and J. Oh, “Social attention: Modeling attention in human crowds,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[12] H. Cheng and M. Sester, “Modeling mixed traffic in shared space using lstm with probability density mapping,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 3898–3904.

[13] J. S. Coleman and J. James, “The equilibrium size distribution of freely-forming groups,” *Sociometry*, vol. 24, no. 1, pp. 36–45, 1961.

[14] P. Laube, S. Imfeld, and R. Weibel, “Discovering relative motion patterns in groups of moving point objects,” *International Journal of Geographical Information Science*, vol. 19, no. 6, pp. 639–668, 2005.

[15] M. Sester, U. Feuerhake, C. Kuntzsch, and L. Zhang, “Revealing underlying structure and behaviour from movement data,” *KI-Künstliche Intelligenz*, vol. 26, no. 3, pp. 223–231, 2012.

[16] S. Sandıkcı, S. Zinger *et al.*, “Detection of human groups in videos,” in *International Conference on Advanced Concepts for Intelligent Vision Systems*. Springer, 2011, pp. 507–518.

[17] W. Ge, R. T. Collins, and R. B. Ruback, “Vision-based analysis of small groups in pedestrian crowds,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 5, pp. 1003–1016, 2012.

[18] M. B. Kjærgaard, M. Wirz, D. Roggen, and G. Tröster, “Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones,” in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 240–249.

[19] L. Bazzani, M. Cristani, and V. Murino, “Decentralized particle filter for joint individual-group tracking,” in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 1886–1893.

[20] M. H. Zaki and T. Sayed, “Automated analysis of pedestrian group behavior in urban settings,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1880–1889, 2018.

[21] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, “A density-based algorithm for discovering clusters in large spatial databases with noise.” in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[22] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, “Fully-convolutional siamese networks for object tracking,” in *European conference on computer vision*. Springer, 2016, pp. 850–865.

[23] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[24] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, no. 14. Oakland, CA, USA, 1967, pp. 281–297.

[25] R. F. Ling, “On the theory and construction of k-clusters,” *The computer journal*, vol. 15, no. 4, pp. 326–332, 1972.

[26] J. R. Munkres and J. R. Munkres, *Topology: a first course*. Prentice-Hall Englewood Cliffs, NJ, 1975, vol. 23.

[27] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, “You’ll never walk alone: Modeling social behavior for multi-target tracking,” in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 261–268.

[28] D. Yang, L. Li, K. Redmill, and Ü. Özgüner, “Top-view trajectories: A pedestrian dataset of vehicle-crowd interaction from controlled experiments and crowded campus,” *arXiv preprint arXiv:1902.00487*, 2019.

[29] F. Pascucci, N. Rinke, C. Schiermeyer, V. Berkhahn, and B. Friedrich, “A discrete choice model for solving conflict situations between pedestrians and vehicles in shared space,” *arXiv preprint arXiv:1709.09412*, 2017.

[30] H. Cheng and M. Sester, “Mixed traffic trajectory prediction using lstm–based models in shared space,” in *The Annual International Conference on Geographic Information Science*. Springer, 2018, pp. 309–325.

[31] A. P. Hare, *Handbook of small group research*. Free Press, 1976.