Tensor Methods for Group Pattern Discovery of Pedestrian Trajectories

Abdullah Sawas*, Abdullah Abuolaim*, Mahmoud Afifi, and Manos Papagelis
EECS, Lassonde School of Engineering, York University, Canada
Email: {asawas, abuolaim, mafifi, and papaggel}@eecs.yorku.ca

***Abstract*—Mining large-scale trajectory data streams (of moving objects) has been of ever increasing research interest due to an abundance of modern tracking devices and its large number of critical applications. In this paper, we are interested in mining *group patterns* of moving objects. *Group pattern mining* describes a special type of trajectory mining task that requires to efficiently discover trajectories of objects that are found in close proximity to each other for a period of time. In particular, we focus on *trajectories of pedestrians* coming from motion video analysis and we are interested in interactive analysis and exploration of group dynamics, including various definitions of group *gathering* and *dispersion*. Towards this end, we present a suite of (three) *tensor-based methods* for efficient discovery of evolving groups of pedestrians. Traditional approaches to solve the problem heavily rely on well-defined clustering algorithms to discover groups of pedestrians at each time point, and then post-process these groups to discover groups that satisfy specific group pattern semantics, including time constraints. In contrast, our proposed methods are based on efficiently discovering *pairs of pedestrians* that *move together over time*, under varying conditions. *Pairs of pedestrians* are subsequently used as a building block for effectively discovering *groups of pedestrians*. The suite of proposed methods provides the ability to adapt to many different scenarios and application requirements. Furthermore, a *query-based search method* is provided that allows for *interactive exploration and analysis* of group dynamics over time and space. Through experiments on real data, we demonstrate the effectiveness of our methods on discovering group patterns of pedestrian trajectories against sensible baselines, for a varying range of conditions. In addition, a visual testing is performed on real motion video to assert the group dynamics discovered by each method.**

***Index Terms*—Trajectory mining, group pattern mining, pedestrian behavior**

I. Introduction

Advances in location acquisition and tracking devices have given rise to the generation of enormous trajectory data consisting of *spatial* and *temporal* information of moving objects, such as persons, vehicles or animals [1]. These trajectories can either be physically constrained (e.g., a pedestrian walking on a sidewalk) or unconstrained (e.g., a bird's flight). Mining trajectory data to find interesting patterns is of increased research interest due to a broad range of useful applications, including analysis of transportation systems, location-based social networks, and pedestrian behavior [2]–[4].

The primary focus of this research is on discovery of *pedestrian group patterns* through mining moving *pedestrian trajectories*. *Group pattern* mining describes a special type of trajectory mining task that seeks to efficiently discover moving objects that are found in close proximity to each other for a period of time. This is an important step towards understanding *pedestrian behavior*, including *group gathering* (people coming together) and *group dispersion* (people distributing over a wider area). To motivate our problem domain, Fig. 1 provides a simple visual example of some interesting group dynamics. It depicts a train station scene taken from a video surveillance camera, where four pedestrian trajectories are highlighted. By careful analysis of the trajectories one can gather that pedestrians *#23* (*yellow*) and *#24* (*green*), met each other in the station for a while, and then exited the station following alternate routes. Meanwhile, pedestrians *#53* and *#54* (*white*) walked and exited the station together (i.e., continuously stayed within close proximity to each other). Enabling this kind of analysis and understanding pedestrian grouping patterns can support a variety of useful applications ranging from monitoring physical areas [5], such as shopping malls, train stations, and airports to supporting pedestrian behavioral studies [6], [7]. In fact, crowded scenes would render the analysis more challenging, as individuals are often intermixed with the crowd. Motion video analysis raises additional challenges for the problem of interest. For example, the actual proximity of two pedestrians depends on the amount of video's perspective distortion (i.e., pedestrians far from the camera appear smaller than pedestrians closer to it). We address these problems in a data preprocessing phase

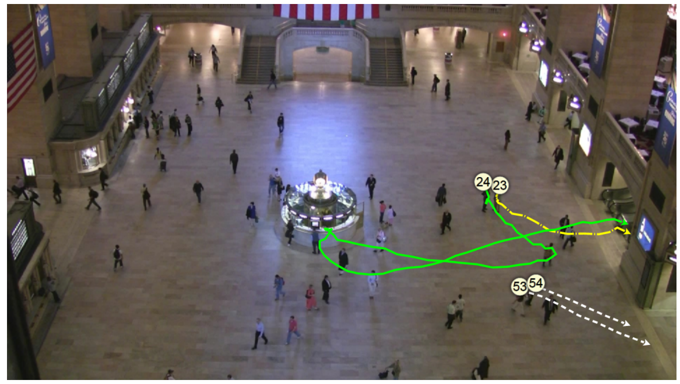


Fig. 1: Four pedestrian trajectories are highlighted in this scene of a train station. Pedestrians *#23* and *#24*, have met for some time and then exited the station following different routes. Pedestrians *#53* and *#54* walked and exited the station together.

*Both authors contributed equally to this work.

and discuss potential implications. Our methods are orthogonal and can be applied to any trajectory data (not necessarily coming from motion video).

There are many definitions of group patterns studied in the literature. The *flock pattern* [8] refers to groups of trajectories that stay and move together, as a cluster, under a predefined threshold of distance, over a certain time period. This pattern is not sufficient to deal with moving objects that divert in a wide area (potentially leaving a group and joining other groups over time). Another group pattern is the *moving cluster* [9], which defines the group as a sequence of spatial clusters that appear in consecutive snapshots of the object movements, such that two consecutive spatial clusters share a large number of common objects. However, this pattern does not require that the group's members are unique throughout the snapshots (i.e. it is not required to have the same members in the clusters). The *convoy pattern* [10] tries to merge the two concepts, such as the group consists of at least m objects moving together for at least k consecutive time instants. The *gathering pattern* [11] is another group pattern that focuses on adjacent clusters that move close to each other over time.

In the case of studying pedestrian groups, there is no single group pattern that can exactly describe the pedestrian group behavior, so more flexible definitions of groups are encouraged. In principle, pedestrians who *intentionally* walk together are considered a group. Accordingly, to model this kind of group pattern we should take into consideration the time dimension, and allow a group of pedestrians to be formed and dispersed over time. In addition pedestrian groups are typically small. For example, some studies observed that pedestrian groups usually consist of two to four members, while groups of size five or higher are considered rare cases [12]. Therefore emphasis should be given in methods that efficiently find small clusters. Recently, R. Lan *et al.* [13] presented a new group pattern, called *evolving group pattern* that defines an *evolving group* as a dense group of trajectories that share common behavior in most of the time and change gradually over time. Similarly, Q. Fan *et al.* [14] introduced a platform for mining co-movement trajectory patterns. In fact, it relaxes the "moving together" constraint, by allowing individual objects to join or leave a group at different times. This group pattern, while not exactly the same, is conceptually closer to the type of group patterns that we want to discover in this work. Briefly, our group pattern definition is based on what consists a *pair over time*. Then, pairs are used as building blocks to define larger groups (see Section II for details). Starting with pairs, allows to define more versatile strategies of group pattern discovery. It also allows to improve the time performance by considering only relevant pairs in the grouping phase.

Towards this end, we present a suite of three tensor-based methods for efficiently discovering pairs and groups of moving objects that are intentionally traveling together in space and time. Our methods assume that the (x, y) coordinates of the motion trajectory for each pedestrian are given at each time point —we argue that this assumption is valid based on the rapid development of location-tracking devices (e.g., GPS) and vision-based pedestrian detection and tracking techniques (refer to representative works of vision-based detection and tracking technologies [5], [15]). The first method, called *locTgroups*, is a local *spatial-only* method that finds pairs at a certain time point. The second method, called *globTgroups*, is a global *spatio-temporal* method that finds pairs similar to the *flock pattern*. Based on these two methods, we derive a novel third method, called *timeWgroups* that utilizes a *time window* in order to find pairs. Once pairs are in place, all three methods utilize the same grouping method that finds groups.

There is a key idea that differentiates our approach to existing approaches. Existing methods operate in the following two phases: In the first phase, they utilize a spatial cluster algorithm (typically DBSCAN [16], any of its variants or other from the rich literature [1]) to discover groups (clusters) of objects at any specific time point. Then, in a second phase, they discover groups that follow the semantics of a specific group pattern by post-processing already discovered groups of phase 1. While this approach works, it inherits limitations of the clustering algorithms themselves, and it doesn't exploit the group pattern semantics up front. The latter is especially important when the time dimension is critical for a group pattern. In contrast, our approach is to first discover pairs of pedestrians that *spatially move together over a certain time interval*, then utilize the pairs to discover groups. That way, our methods provide more flexibility and can deal with different group pattern semantics, including *flock*, *evolving groups* and more, under the same framework.

Trajectory data sets are typically very sparse so representing the data using a simple tensor would contain a large number of zero values (this is because not all pedestrians move at the same time interval). To improve efficiency, we represent the data as a *sparse tensor*. A *sparse tensor* allows to keep large-data sets in memory and provides significant improvements in terms of time performance by utilizing optimized and scalable matrix operations, provided in many existing toolkits and software packages [17].

In summary, the major contributions of this work include:

- a novel framework for optimized tensor-based methods for group pattern discovery of pedestrian trajectories.
- a novel time window based method, *timeWgroups*, for efficiently and effectively finding groups of pedestrians with variant group pattern semantics.
- a thorough evaluation of group pattern discovery methods on large-scale real data, for a varying range of conditions. In addition, a visual testing is performed on real motion video to assert the groups discovered by each method.
- a novel tool that supports *interactive exploration of group dynamics over time* by end-users.
- making *source code*, *data*, *sample rendered videos* and *an online interactive demonstration* publicly available to encourage reproducibility of results. They can all be accessed at the following website: *https://sites.google.com/view/pedestrians-group-pattern/*.

TABLE I: Table of notations

Symbol	Meaning
G	Pedestrian group
(x, y)	Pedestrian's position in the 2D Cartesian coordinate systems
$\mathbf{p}_i$	Trajectory of pedestrian number i
$\mathbf{M}$	Sparse tensor contains all pedestrians' data over a certain period of time
N, n	Number of pedestrians
t	Given time point
V	Time period
w	Time window size
$\mathbf{D_l}$	Distance between two pedestrians
$\mathbf{D_g}$	Global distance between two pedestrians' trajectories
$\mathbf{D_w}$	Maximum distance over w time window
τ	Predefined threshold
ANG	Average number of identified groups (in the video) per time unit according to certain proximity distance threshold τ
AGS	Average group size (in the video) per time unit defined as number of pedestrians assigned to a group in the frame divided by the number of identified groups in that frame
ρ	Density of groups
$\mathbf{v}$	Feature vector extracted from a pair of pedestrians
m	Number of the random samples picked from a trajectory
$\mathbf{x}_t$, $\mathbf{y}_t$	x and y coordinates of all pedestrians' trajectories at time t
Φ	A mapping function of pedestrian indices

The remainder of this paper is organized as follows: Section II introduces notation and formally defines the problem. Our methods and overall framework are presented in Section III. Section IV provides details of our experimental evaluation. In Section V we present an interactive tool for exploration of group dynamics. After reviewing the related work in Section VI, we conclude in Section VII.

II. Problem Definition

In this section, we state the problem by giving the definitions of *pedestrian groups* and *pairs* that we aim to find. Table I lists the symbols we will use and their meanings. Given (x, y) coordinates of the motion trajectories for N pedestrians over V time points, the goal is to find pedestrian groups at any given time interval. The initial phase of the problem definition is the understanding of the pedestrian group's characteristics. The *evolving group* pattern is the most appropriate group pattern to describe the pedestrian group. However, other group patterns can be more appropriate for specific applications. In criminal investigation, for example, it can be more interesting to sieve through video archives and find activities of coherent pairs of pedestrian that have happened in the past. Accordingly, the *flock pattern* would be more suitable to describe this kind of pedestrian groups.

Definition 1: **(Coherent pair)** Given two pedestrian trajectories $(\mathbf{p}_i, \mathbf{p}_j)$, $i, j \in [1..N]$, where N is the total number of pedestrian over the given time period, (i, j) is considered a coherent pair, *iff* the average distance between them, over the entire given time, is below a fixed threshold. Intuitively,

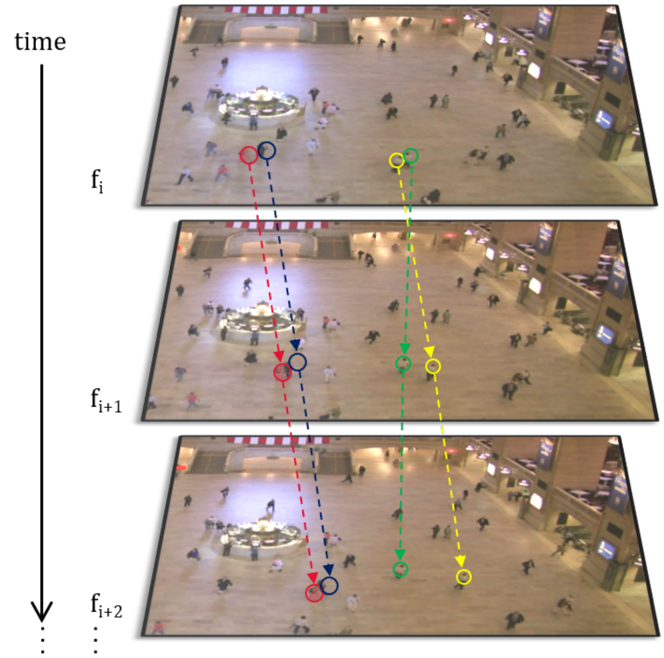


Fig. 2: *Coherent pair* follows similar path over time (red/blue arrows). Non-coherent pair (green/yellow arrows).

the coherent pair is a pair of pedestrians' trajectories that *intentionally* appear, travel, and disappear together over time.

Definition 2: **(Coherent group)** is a group of pedestrians; each one of them belongs to one or more *coherent pairs*, such that each pair shares a common pedestrian with at least one of the group pairs, i.e. $G = \{1, 2, ..., k\}$, $\forall i \in \{1, ..., k\} \exists j, m \in \{1, ..., k\} : (i, j)$ and (i, m) are coherent pairs.

Definition 3: **(Pedestrian pair)** Given two pedestrian trajectories $(\mathbf{p}_i, \mathbf{p}_j)$ over a time window of size w, where $w \geq 1$, the pair (i, j) is considered a pedestrian pair *iff* the maximum distance between $\mathbf{p}_i$ and $\mathbf{p}_j$, during this time window, is below a certain threshold.

Definition 4: **(Pedestrian group)** is a group of pedestrians that belong to one or more *pedestrian pairs* over w time window. As w approaches to 1, the pattern becomes more susceptible to local changes over time. On the other hand, a larger w, that is close to the entire trajectory length, forms a *coherent group*. In other words, the pedestrian group is a dense group of pedestrians that *intentionally* walk together and can be gradually changed over time.

As we deal with a big amount of trajectory data, we use an efficient data structure to represent the pedestrians' trajectories over time. We store all trajectories in a big sparse tensor $\mathbf{M} \in \mathbb{R}^{N \times V \times 2}$), where N is the number of pedestrians' trajectories over time V. At any given time point t, $\mathbf{M}_{(i,t)} = (x_{i_t}, y_{i_t})$, where x_{i_t}, y_{i_t} are the (x, y) coordinates of pedestrian i at t. Our objective is to solve the following problems:

Problem 1: Given a set of pedestrian trajectories $(\mathbf{p}_i, \mathbf{p}_j)$, $i, j \in [1..N]$, find the *coherent pairs* and *coherent groups* of pedestrians in a certain time interval (*globTgroups*).

Problem 2: Given a set of pedestrian trajectories $(\mathbf{p}_i, \mathbf{p}_j)$,

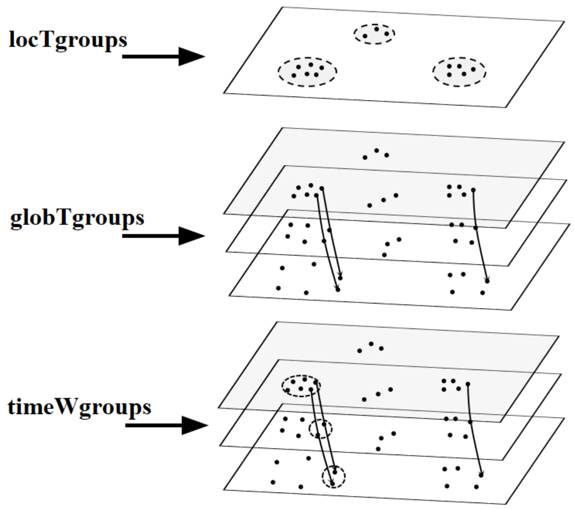


Fig. 3: A visual summary of the three proposed grouping methods: *locTgroups*, *globTgroups*, and *timeWgroups*.

$i, j \in [1..N]$, find the *pedestrian pairs* and *pedestrian groups* in a certain time interval (*locTgroups* and *timeWgroups*).

III. Methodology

In our problem domain, a large number of moving objects is expected to appear at the same time point (e.g., a train arrives). Since the performance is critical, dealing with such huge data requires that we adopt a straightforward approach to find groups with different patterns semantics. We use the Euclidean distance as a metric to measure the distance between trajectory pairs. Then a recursive algorithm is applied to extend pairs to groups. In this section, we present the details of the proposed methods to find pedestrian pairs and groups. As discussed earlier and later in Section VI, existing approaches rely on well-known clustering methods (e.g., DBSCAN) to first find groups (clusters) at any time point. In contrast, we consider the temporal nature of trajectories upfront by splitting the process of finding the groups into two steps: i) finding pairs over time, and ii) extending the discovered pairs to discover groups. This approach allows to define alternate strategies for group patterns (i.e. *coherent group* and *pedestrian group*) using the same framework.

A. Finding Pair Patterns

Finding *pairs* operates as a building block in our framework for finding both *coherent groups* and *pedestrian groups*. Given a sparse tensor ($\mathbf{M} \in \mathbb{R}^{N \times V \times 2}$) that contains N pedestrians' trajectories over time V, $\mathbf{M}$ captures the (x, y) coordinates of each trajectory per time point. Using this representation, we present the following three methods to find pair patterns:

1) Local Spatial Pairing of Trajectories (locTgroups): Given N_t pedestrians at a time point t, we first define the proximity measure between two pedestrians $i, j \in [1..N_t]$, in which we use the Euclidean distance $D_l(\mathbf{p}_i^t, \mathbf{p}_j^t)$ (Equ. 1) between two trajectories $\mathbf{p}_i$ and $\mathbf{p}_j$ in space such that:

$$D_l(\mathbf{p}_i^t, \mathbf{p}_j^t) = \sqrt{(x_i^t - x_j^t)^2 + (y_i^t - y_j^t)^2}. \tag{1}$$

TABLE II: Comparison between the proposed three methods: *locTgroups*, *globTgroups* and *timeWgroups*.

Comparison	***locTgroups***	***globTgroups***	***timeWgroups***
On-line	✓	✗	w time delay
Group gathering/dispersion	✓	✗	✓
Spatial proximity	✓	✓	✓
Temporal proximity	✗	✓	✓

Any two pedestrians that have distance below a predefined threshold τ will form a pair at the particular time point t. In some real application, τ should reflect the real proximity distance allowed. For example, in motion video a pedestrian's x, y coordinates are the pixel coordinates in each video frame (i.e., still image). For such motion video application, there should be a way to map dimensions of pixel units (image plane space) to meter units (real world space). Also the threshold τ can be dynamic based on factors such as density of trajectories and other application-specific factors. More analysis of the impact of τ is presented in Section IV.

2) Global Spatio-temporal Pairing of Trajectories (globTgroups): Given pedestrian $i \in N$, we are interested to find the *coherent pairs* of the pedestrian i over the entire time period V (see Fig. 2). However, since i appears for a certain period of time T_i, we limit the search space into the N_{T_i} pedestrian trajectories overlapping with the pedestrian i's trajectory. To find the spatio-temporal coherent pairs, we average the Euclidean distance over time as a proximity measure between two pedestrian trajectories using the following equation:

$$D_g(\mathbf{p}_i, \mathbf{p}_j) = \frac{\sum_{t=1}^{T_i} D_l(\mathbf{p}_i^t, \mathbf{p}_j^t)}{T_i}, \tag{2}$$

where $D_g(\mathbf{p}_i, \mathbf{p}_j)$ is the average distance between pedestrians i and j. After constructing D_g, any two trajectories can be considered as a pair if their distance is under a predefined threshold τ.

3) Time Window Based Pairing of Trajectories (timeWgroups): For the previous two methods, there are some strengths and weaknesses summarized in Table II. The local spatial method (*locTgroups*) can be more easily adapted to real-time applications, in which case, it can process sufficient amount of time points with less computational power. Furthermore, this method can be easily extended by implementing some incremental algorithms such as adopting speed, acceleration, and direction. It is also capable to capture the group gathering and group dispersion dynamics. However, this method computes only the spatial proximity and ignores the temporal one. On the other hand, the global spatio-temporal method (*globTgroups*) is efficient in finding *coherent pairs* in space and time. However, this method runs in an off-line manner, in which it requires the entire video to be available beforehand. It is also not able to capture the group gathering and group dispersion dynamics.

By comparing and discussing the shortcomings of the previous two methods, the task now is motivated towards compromise and overcome the shortcomings. Towards this end, we propose a novel method, *timeWgroups*, that finds pairs efficiently in space and time, and naturally captures the group

Algorithm 1 Grouping of Pedestrian Pairs

1: **procedure** GETGROUPS($\mathbf{D}$, τ) ▷ Group pedestrians using distance function $\mathbf{D}$ and proximity threshold τ
2: $\mathbf{P}(i,j) \leftarrow \mathbf{1}$, for all $i,j \in [1..N]$ where $\mathbf{D}(i,j) < \tau$
3: $V(i) \leftarrow 0$, for all $i \in [1..N]$
4: $\mathbf{Groups} \leftarrow []$ ▷ Identified pedestrian groups
5: $N_G \leftarrow 0$ ▷ Number of identified groups
6: **for all** Pedestrian $i \in [1..N]$ **do**
7: $G_i \leftarrow$ FindGroupOf(i);
8: **if** $G_i \neq \phi$ **then**
9: $N_G \leftarrow N_G + 1$
10: $\mathbf{Groups}(N_G) \leftarrow G_i$
11: **return** $\mathbf{Groups}$
12: **procedure** FINDGROUPOF(i) ▷ Recursively find the group of pedestrian i
13: **Global** V, $\mathbf{P}$
14: **if** $V(i)$ **then return** ϕ
15: $V(i) \leftarrow 1$;
16: $G_i \leftarrow \phi$
17: **for all** k where $\mathbf{P}(i,k) = 1$ **do**
18: **if** $V(k) \neq 1$ **then**
19: $G_k \leftarrow$ FindGroupOf(k)
20: $G_i \leftarrow G_i \cup G_k$
21: **if** $G_i \neq \phi$ **then** $G_i \leftarrow G_i \cup \{i\}$
22: **return** G_i

gathering/dispersion dynamics. We use a step time window of size w. The distance between $\mathbf{p}_i$ and $\mathbf{p}_j$ over w time window is calculated by the following equation:

$$D_w(\mathbf{p}_i, \mathbf{p}_j) = \max_{\{t \to t+w-1\}} \{D_l(\mathbf{p}_i^t, \mathbf{p}_j^t)\}. \tag{3}$$

Finally, a visual summary of the three proposed grouping methods *locTgroups*, *globTgroups*, and *timeWgroups* is illustrated in Fig. 3.

B. Recursive Grouping of Pairs

Given a trajectory p_i of pedestrian (i) and using the pairwise proximity distances found by any of the previous methods, we group together all the pedestrians (k) that are paired with the pedestrian (i). Formally, the group of pedestrian (i) is:

$$G_i = \{i\} \cup \{G_k : k \neq i, D(\mathbf{p}_i, \mathbf{p}_k) < \tau\}$$

where $D \in \{D_l, D_g, D_w\}$ is the distance metric defined according to *locTgroups*, *globTgroups* or *timeWgroups* methods, respectively. Until all the pairs are visited, we keep expanding the group (G) by adding all the pairs of the group members recursively as described in Algorithm 1.

The algorithm starts by initializing two global variables $\mathbf{P}$ and V in lines 2 and 3, respectively. $\mathbf{P}$ is a boolean matrix of size $N \times N$ that takes the value 1 when two pedestrians i, j are in one pair, otherwise it takes the value 0. V is a vector of size N that keeps a record of whether a pedestrian i has been visited or not. In line 4 an empty array $\mathbf{Groups}$, which will contain the identified groups, is initialized. The algorithm will call at line 7 the function *FindGroupOf(i)* which will initially, at line 14, verify that pedestrian i was not visited before. Then recursively iterates through the pairs of pedestrian i to find all their groups and return the union of all these groups. If pedestrian i has been visited, then the function returns an empty set, meaning that the pedestrian already belongs to a group (or does not have not been been paired).

TABLE III: Details of the dataset

Resolution (px)	$1,920 \times\ 1,080$
Annotated frame count	6000
Annotated pedestrian count	$12,684$
Average pedestrian number per frame	123
Max pedestrian number per frame	332
Number of data points	$1,266,502$

C. Tensor-based Optimization

Each method operates on different data size; the $locTgroups$ receives on-line trajectory information at a single time point. While, the $timeWgroups$ waits until it receives w time points. Eventually, the $globTgroups$ requires the entire trajectories data to be available. Pedestrians trajectories have a sparse representation, since their trajectories are not aligned in time. Based on that, we perform the following steps for performance optimization:

- At each time interval, we reduce the computations by applying the algorithms on the existing pedestrians at that interval using a map function

  $$\Phi : \Phi(I) \to I_{sub}, \Phi(I_{sub})^{-1} \to I,$$

  where I is the indices of N pedestrians in $\mathbf{M}$ and $I_{sub} \subset I : |I_{sub}| << |I|$ that contains the existing pedestrians.
- We use optimized matrix operations to calculate the distances in Eq. 1 2, and 3, as following: let $\mathbf{x}_t$ and $\mathbf{y}_t$ be column vectors represent the x and y coordinates of N_t pedestrians' locations at time t, respectively. We construct $\mathbf{X}_t, \mathbf{Y}_t \in \mathbb{R}^{N_t \times N_t}$ matrices, such that:

  $$\mathbf{X}_t = \begin{bmatrix} \mathbf{x}_t & \mathbf{x}_t & \dots & \mathbf{x}_t \end{bmatrix}, \tag{4}$$

  $$\mathbf{Y}_t = \begin{bmatrix} \mathbf{y}_t & \mathbf{y}_t & \dots & \mathbf{y}_t \end{bmatrix}. \tag{5}$$

  In that way, the distances between all pedestrians' at time t can be calculated by the following equation:

  $$\mathbf{D}_t = \sqrt{(\mathbf{X}_t - \mathbf{X}_t^T)^2 + (\mathbf{Y}_t - \mathbf{Y}_t^T)^2}, \tag{6}$$

  where $\mathbf{D}_t$ is the distances matrix between all pedestrians at time t.

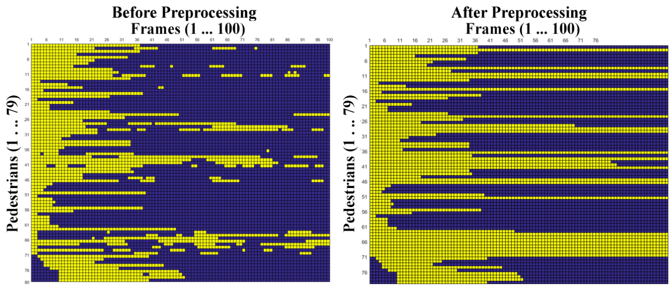


Fig. 4: Left and right maps show the trajectories before and after filling the unreported stationary positions respectively. Yellow cell indicates the reported (x, y) coordinates of moving pedestrian and the blue cell indicates the unreported position of stationary pedestrian.

Fig. 5: High perspective distortion. The pedestrian closer to camera is about two times larger than the one farther.

IV. Experimental Evaluation

Experimental Setup: All methods were implemented in Matlab on an Intel® core™ i-7 6700 @ 3.40GHz machine with 16 GB RAM.

Data: We use the dataset proposed by Shuai Yi *et al.* [18] that contains real-scene crowd data. The trajectory dataset was extracted from one hour of video recorded by a surveillance camera that captures walking pedestrians at a train station. Data points were manually annotated every 2/3 second for 12,684 trajectories. It is worth mentioning that pedestrian trajectory is usually collected using GPS and/or some pedestrian tracking techniques in video streams. In this dataset on average there are around 123 pedestrians per video frame with the most crowded frame containing 332 pedestrians. Table III provides a summary of this dataset. Eventually, visualization videos of the following experiments are available in the following link: *https://goo.gl/qFF9t5*.

Preprocessing: In surveillance video, pedestrians move toward and away from the camera, and there is a noticeable amount of perspective distortion. This distortion results in objects' foreshortening, where the objects closer to camera appear larger than faraway objects with similar dimensions in real-world. This perspective distortion does not give the actual distance between pedestrians, which may affect the accuracy of the proximity measure. To overcome this distortion, thus, the x, y coordinates for each pedestrian are projected into the ground plane using an estimated Homography [19] as suggested by [20]. The proximity between pedestrians, thus, can be calculated after the projection, to give the actual distance. To roughly measure the amount of perspective distortion in the dataset we use, we pick two pedestrians, one is close to the camera whereas the other is faraway (Fig. 5). As shown in Fig. 5, the pedestrian closer to the camera is about two times larger than the pedestrian farther away. The impact of the projection distortion is a video specific problem that we had to address during the pre-processing phase, but it is out of the scope of this research. Our group pattern discovery methods are orthogonal and can be applied to any trajectory data (not necessarily coming from motion video).

Since the dataset we use reported only (x, y) coordinates of moving pedestrian. Another step was required to fill the discontinuity in pedestrian trajectory. This discontinuity happens when pedestrian stops moving (stationary pedestrian). The intuitive step to fill the discontinuity, between any two time points, is to replicate the last (x, y) coordinates for the unreported position of stationary pedestrian. Fig. 4 shows some pedestrian trajectories before and after filling the unreported stationary positions.

Evaluation Criteria

We have evaluated the proposed methods using the following criteria:

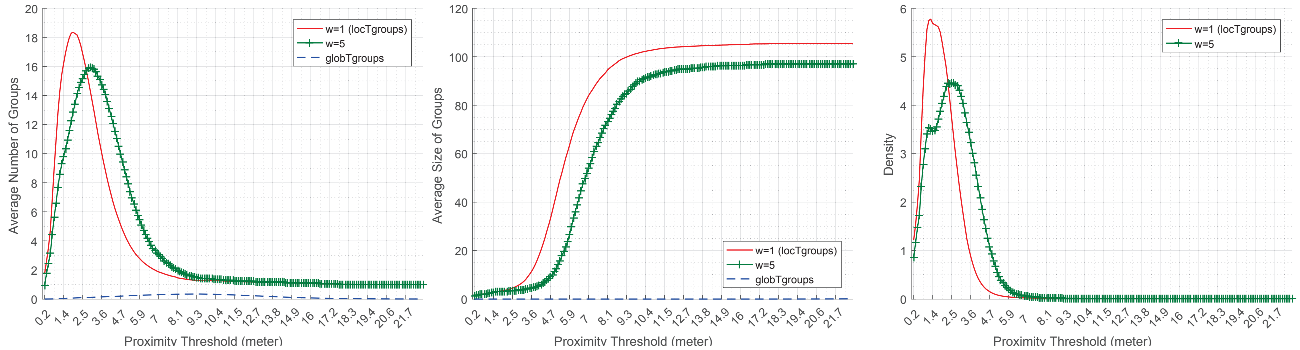
- ANG: Average Number of identified Groups per time unit (e.g. one time point when using the $locTgroups$, w time points with $timeWgroups$, and the entire time interval using the $globTgroups$) according to certain proximity distance threshold τ.
- AGS: Average Group size per time unit defined as the average of number of pedestrians assigned to a group in the frame divided by the number of identified groups in that time unit for all the frames.
- ρ: Density of groups which is defined as

$$\rho = \frac{AGS}{ANG}. \tag{7}$$

- Execution Time: The required time (in seconds) to process the given set of trajectories.

A. Determination of Threshold (τ)

Considering *only* the spatial information of the pedestrians, we can cluster them into groups based on the distance among them using the three proposed methods. However, deciding the minimum proximity (distance) τ is a critical factor that affects the pedestrian groups discovered by each method. Therefore, in order to find the suitable value of τ, we evaluated the three grouping algorithms ($locTgroups$, $globTgroups$, and $timeWgroups$) for different values of threshold τ (ranging from 0.2 to 22 meters). The results are shown in Fig. 6. Proximity threshold τ was represented in meters by estimating



(a) (b) (c)

Fig. 6: Analysis of proximity threshold (meters) impact for the three proposed methods on: (a) the average number of identified groups per time unit, (b) the average group size per time unit, (c) the density of the groups.

the pixel-to-meter ratio of an object in the scene that has a well known height range (e.g. doors, or average height of persons).

As shown in Fig. 6a, for small values of τ the average number of identified groups ANG of the pedestrians' trajectories is small since pedestrians that are not very closed to each other are not going to be grouped. ANG keeps increasing with increasing τ until it reaches a peak (at $\tau = 2, 3, 9.5$ meters for $locTgroups$, $timeWgroups[w = 5]$, and $globTgroups$ methods respectively) then it starts decreasing. This is because with large proximity distance threshold τ people relatively far from each other are now considered to be in the same group causing the size of the group to increase, while the number of groups to decrease. The threshold that maximizes the average number of groups for the *globTgroups* is quite higher than the *locTgroups* and *timeWgroups*; that is because the *globTgroups* has been designed to find *coherent* pairs and groups. As a consequence, the number of groups is expected to be smaller than the number of *evolving groups*. At $\tau = 9.5$ meters, the average number of groups is the max with the *globTgroups*, that means there are some coherent groups of pedestrians that follow the same path but are found far away from each other.

We can notice in Fig. 6b that the average size of the pedestrian groups AGS increases with the increase of proximity threshold, as expected. It can also be seen that at max the AGS for *globTgroups* method is very low (≈ 2.2) compared to the other methods. This is due to the fact that normal pedestrians movement is hardly ever coherent in larger groups.

Interestingly, the density of the groups (ρ) in Fig. 6c shows a peak ≈ 5 at proximity threshold $\tau = 1.5$ meter for the *locTgroups*, and a peak ≈ 4 at proximity threshold $\tau = 2.5$ meters for the *timeWgroups* with $w = 5$. This result of average groups' density obtained by the *timeWgroups*$[w = 5$ and $\tau = 2.5]$ meters, agree with the study presented by Mousaaïd *et al.* [12] that found that the average number of pedestrians in groups is usually less than or equal to four pedestrians. Finally, the method *globTgroups* was excluded from density calculation in Fig. 6c because of the very low number of the average group size; this can be explained by the fact that the *coherent groups*, by definition, have very hard constraints that are rare to happen in the real world —groups of pedestrians that intentionally walk together with a distance below a certain threshold over the entire time interval usually consists of two pedestrians, i.e. *coherent pairs*.

B. Determination of Window Size (w)

As aforementioned, when $w = 1$, the $timeWgroups$ performs exactly the same as the $locTgroups$, and when $w = V$, (V is the entire video frames), the $timeWgroups$ finds the *coherent groups* similar to the *globTgroups*. Consequently, the window size (w) has an obvious impact in the *timeWgroups* method in order to find the desirable group patterns.

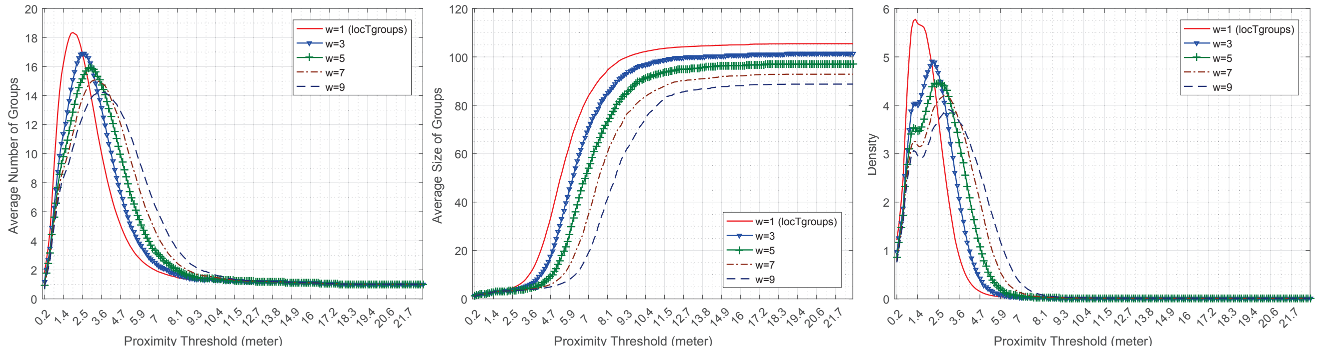
Considering the time window size w in grouping pedestrian trajectories adds a temporal aspect to the grouping method, besides the spacial aspect. This allows to filter pedestrians activities based on both time and space. For example, pedestrians crossing each other (walking in opposite direction) will not be grouped together if the time they spent during crossing was below the average pedestrian crossing time. If however the pedestrians have a longer interaction compared to a normal crossing, then they will be grouped together.

The results shown in Fig. 7 for the impact of varying the window size w on the average number of groups ANG, average group size AGS, and groups' density ρ at different values of τ affirms that conclusion. Increasing the window size w effectively controls the time that pedestrians need to spend close to each other in order for them to be considered as a group. Hence, ANG and AGS decreases with increasing w. Finally, our results for peak value of the group density (ρ) matches the finding by Mousaaïd *et al.* [12]. That indicates the effectiveness of the *timeWgroups* in effectively finding *evolving groups*.

C. Time Performance

Based on the tensor-based optimization presented in Section III-C, the proposed methods can be ran efficiently in a reasonable execution time. Fig. 8 shows the total execution time for the proposed methods to find the pedestrian groups in the *entire* video (1 hour) at different proximity distance threshold τ.

As the value of τ increases, the required time for the grouping method increases, because high τ value means more



(a) (b) (c)

Fig. 7: Analysis of window size and threshold impact for the *timeWgroups* proposed method on: (a) the average number of identified groups per time unit, (b) the average group size per time unit, (c) the density of the groups.

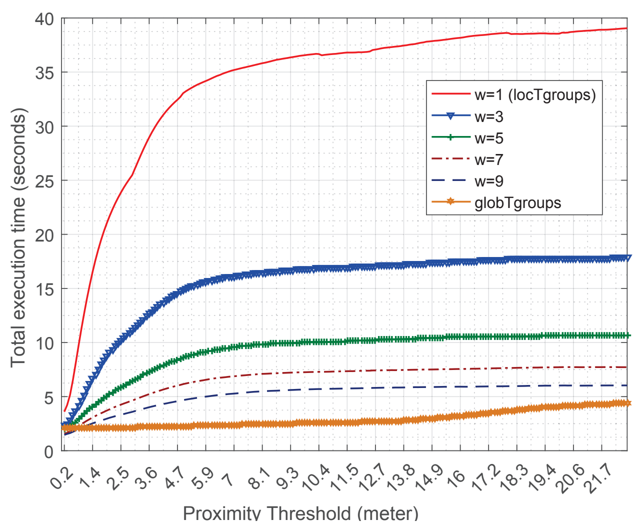


Fig. 8: Execution time of the proposed methods for finding the pedestrian groups in the **entire** video ($\approx$ 1 hour) for varying proximity threshold τ.

pedestrians are considered as pairs, resulting in longer processing time required by the grouping algorithm 1. However, after a certain value of τ, the processing time is converging. This is because we are getting the same groups, since no more pairs are generated. This threshold corresponds to the maximum distance between the pedestrians in the scene. It can be seen that the *timeWgroups* method runs faster for larger window size w because the grouping method needs to be executed fewer times compared to smaller window size values.

V. Interactive Exploration of Group Dynamics

Interactive exploration of group dynamics of pedestrians in motion videos is an important application. In this Section, we describe the main features of an online tool that we have developed to enable ad hoc search and retrieval of pedestrian groups [21]. A live online demonstration of this tool can be accessed at: *https://goo.gl/qFF9t5*.

The tool allows to extract information about pedestrians given only their trajectories. A snapshot of the tool's User Interface (UI) is shown in Fig. 9. The interface comprises of four panels: (A) a video frame panel showing the pedestrian IDs and trajectories in the scene; (B) a frame slider at the top to navigate video frames (i.e., a timeline slider) and provides a summary of the number of pedestrians at each video frame; (C) a video panel at the top left showing aggregated statistics and insights about the current frame, and (D) a group information panel, at the bottom left, that shows analysis of pedestrian groups for different proximity threshold settings.

Using this service can reduce the time spent in searching and analyzing videos and can also help researchers in this field to validate the results of their algorithms. The tool visualizes the results and helps answering several important questions:

- Showing the route of each pedestrian projected on the scene.
- Identifying the entry/exit gates each pedestrian has used to enter/exit the scene.
- Visualizing the location where pedestrians spent most of their time in the scene.
- Reporting the length of a pedestrian stay in the scene.
- Querying about pedestrians who stayed in the station more than the average time.
- Querying about where, when and who are other people that a certain pedestrian has been moving close to.

As this tool deals with thousands of frames, several implementation optimizations had to be considered to enhance its performance in terms of fast data loading and online updating of the visualization. Moreover, some statistics are computed on the browser to reduce data transfer requirements.

VI. Related Work

Discovery of pedestrian groups is a special type of data mining task that can facilitate pedestrian behavior analysis. Our work is related to topics of *trajectory-based pedestrian group mining* and to *vision-based pedestrian group detection*.

Trajectory-based Pedestrian Group Mining: The works more related to our research has already been reviewed in Section I, so here we expand to other related work of this broad topic. Clustering the trajectories was utilized by Gaffney and Smyth *et al.* [22] using a mixture regression model. However,

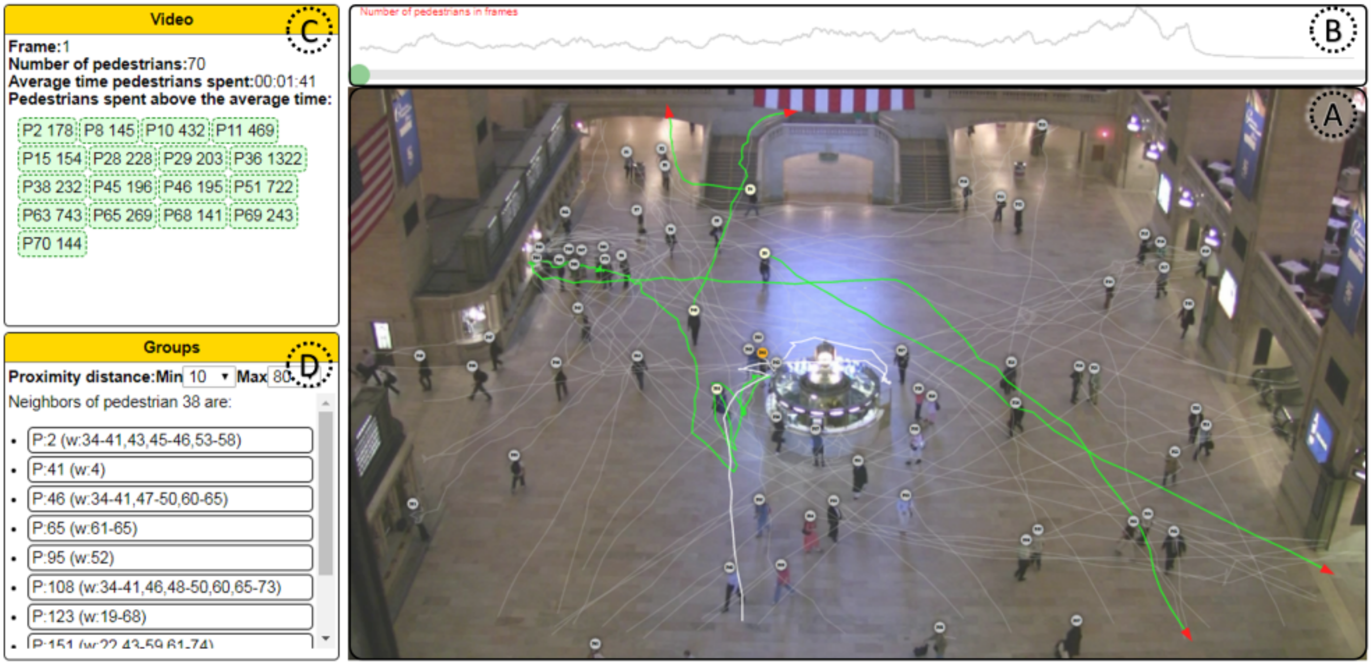


Fig. 9: Snapshot of the tool that allows for an interactive exploration of pedestrian group dynamics.

their method is applied to the entire trajectories in order to find the groups. C^2P, a two-phase clustering algorithm, was used in order to gradually cluster closest pairs of trajectories [23]. In the first phase, a set of sub-clusters were found. In the second phase, the sub-clusters were merged iteratively to construct finer final clusters. However, the C^2P algorithm does not deal with the temporal information that may be considered in the data points. Authors in [24] proposed a method for matching pedestrian trajectories on maps using a dynamic time warping algorithm, however they did not explore the issue of matching pedestrian trajectories together to extract groups. Lee *et al.* [25] presented a partition-and-group framework that is based on clustering sub-trajectories instead of the entire trajectories; the focus is on local characteristics of the trajectories. A linear interpolation method was adopted by Jeung *et al.* [26] to complete missing spatial data over time in order to find *convoy patterns* (i.e. coherent moving groups) by applying a density clustering algorithm followed by a post-processing to find the coherent moving groups. Pelekis *et al.* [27] presented a ReTraTree indexing structure to maintain (sub-)trajectories' information over time. Lan *et al.* [13] proposed an algorithm to find *evolving groups* by finding the candidate clusters at each time point using DBSCAN [16]. Then, the Hausdorff distance was utilized for each time interval specified by a sliding window. Although the temporal information is considered to capture the changes of groups over time, the method is hard to be adapted with different group patterns, as being limited, in the first stage, by a *spatial* based clustering algorithm performed at each time point.

Vision-based Pedestrian Group Detection: Many vision-based methods have been proposed to detect and track the movement of pedestrians in a video [5], [28]–[30]. With the advent of location-tracking technology, many methods have been proposed to analyze pedestrian movements and recognize specific behaviors (e.g., lying pose recognition [31], anomaly detection [20], and escape behavior [32]). Bastani *et al.* [33] utilized Kalman filter to estimate the trajectory pattern flow of each pedestrian. A symmetrized version of Kullback-Leibler (KL) divergence was used as a metric to build up a similarity graph that is used lately to find pedestrian groups after clustering is performed using the spectral clustering algorithm. This clustering algorithm was used by Rupasinghe *et al.* [34] to extract a set of nodes, where each node represents a particular motion pattern. From another perspective, the study proposed by Zanlungo [35] showed that the direction vectors of interacting pedestrians are perpendicular to each other. Accordingly, the angles between the movement directions of each pair of pedestrians were used in [36] to calculate the probability of being interacting pedestrians. Then, Bayes' theorem was adopted to estimate the pedestrian groups. However, this method can not deal with groups of more than three people.

VII. CONCLUSIONS

We considered the problem of discovering groups of pedestrians when their trajectories are provided. This is an interesting but challenging problem, with a broad range of applications. In particular, we proposed *timeWgroup*, an efficient time window based method that effectively discovers groups of pedestrians of varying group pattern semantics. The novelty provided by our method is based on the idea of first efficiently discovering the pairs of moving objects over time and then, discovering evolving groups by expanding pairs to groups. Moreover, the flexibility provided by our method is important, as pedestrian movement (and probably trajectory data of moving objects in other application domains) does not necessarily adhere to well defined group semantics.

To improve efficiency, we represented trajectory data as a *sparse tensor*. That way, we were able to devise optimized tensor-based operations that could scale to large-scale analysis. For example, we were able to perform group pattern analysis of approximately 1h of motion video, including more than 12k pedestrians and more than 1M trajectory data points, in a matter of *seconds*. To appreciate the efficiency of the methods, one needs to consider that for n moving objects the number of candidate pairs that need to be evaluated are in the order of $O(n^2)$. An even more interesting characteristic of the method is that it can enable interactive exploration and analysis of the group patterns by an end-user.

Overall, the methods we described are *simple* to understand and implement, *accurate*, *fast*, and *general*, so they can be easily adopted in a variety of strategies for group pattern discovery. As such, we expect our methods to be beneficial in diverse settings and disciplines.

Reproducibility: The *source code*, *data*, *sample rendered videos* and *an online interactive demonstration* are publicly available to encourage reproducibility of results. They can be accessed at the following website: *https://sites.google.com/view/pedestrians-group-pattern/*.

Acknowledgments. This research has been partially supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (#RGPIN-2017-05680).

REFERENCES

[1] G. Yuan, P. Sun, J. Zhao, D. Li, and C. Wang, “A review of moving object trajectory clustering algorithms,” *Artificial Intelligence Review*, vol. 47, no. 1, pp. 123–144, 2017.

[2] D. Guo, S. Liu, and H. Jin, “A graph-based approach to vehicle trajectory analysis,” *Journal of Location Based Services*, vol. 4, no. 3-4, pp. 183–199, 2010.

[3] K. Siła-Nowicka, J. Vandrol, T. Oshan, J. A. Long, U. Demšar, and A. S. Fotheringham, “Analysis of human mobility patterns from gps trajectories and contextual information,” *International Journal of Geographical Information Science*, vol. 30, no. 5, pp. 881–906, 2016.

[4] F. Zanlungo, T. Ikeda, and T. Kanda, “Potential for the dynamics of pedestrians in a socially interacting group,” *Physical Review E*, vol. 89, no. 1, p. 012811, 2014.

[5] P. Dollar, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: An evaluation of the state of the art,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[6] D. Cartwritht and A. Zander, *Group dynamics*. New York, NY, USA: Harper and Row, 1960.

[7] G. Le Bon, *The Crowd: A study of the popular mind*. New York, NY, USA: Macmillan New York, 1921.

[8] M. Benkert, J. Gudmundsson, F. Hübner, and T. Wolle, “Reporting flock patterns,” *Computational Geometry*, vol. 41, no. 3, pp. 111–125, 2008.

[9] P. Kalnis, N. Mamoulis, and S. Bakiras, “On discovering moving clusters in spatio-temporal data,” in *Spatial and Temporal Databases (SSTD)*. Springer, 2005, pp. 364–381.

[10] H. Jeung, H. T. Shen, and X. Zhou, “Convoy queries in spatio-temporal databases,” in *International Conference on Data Engineering (ICDE)*. IEEE, 2008, pp. 1457–1459.

[11] K. Zheng, Y. Zheng, N. J. Yuan, and S. Shang, “On discovery of gathering patterns from trajectories,” in *International Conference on Data Engineering (ICDE)*. IEEE, 2013, pp. 242–253.

[12] 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.

[13] R. Lan, Y. Yu, L. Cao, P. Song, and Y. Wang, “Discovering evolving moving object groups from massive-scale trajectory streams,” in *Mobile Data Management (MDM)*. IEEE, 2017, pp. 256–265.

[14] Q. Fan, D. Zhang, H. Wu, and K.-L. Tan, “A general and parallel platform for mining co-movement patterns over large-scale trajectories,” *Proceedings of the VLDB Endowment*, vol. 10, no. 4, pp. 313–324, 2016.

[15] S.-H. Bae and K.-J. Yoon, “Robust online multiobject tracking with data association and track management,” *IEEE transactions on image processing*, vol. 23, no. 7, pp. 2820–2833, 2014.

[16] 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.

[17] T. Zhang, S. H. Soh, X. Fu, K. K. Lee, L. Wong, S. Ma, G. Xiao, and C. K. Kwoh, “Hpcgen a fast generator of contact networks of large urban cities for epidemiological studies,” in *Computational Intelligence, Modelling and Simulation (CSSim)*. IEEE, 2009, pp. 198–203.

[18] S. Yi, H. Li, and X. Wang, “Understanding pedestrian behaviors from stationary crowd groups,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 3488–3496.

[19] O. Faugeras, *Three-dimensional computer vision: a geometric viewpoint*. MIT press, 1993.

[20] A. A. Abuolaim, W. K. Leow, J. Varadarajan, and N. Ahuja, “On the essence of unsupervised detection of anomalous motion in surveillance videos,” in *International Conference on Computer Analysis of Images and Patterns*. Springer, 2017, pp. 160–171.

[21] A. Sawas, A. Abuolaim, M. Afifi, and M. Papagelis, “Trajectolizer: Interactive analysis and exploration of trajectory group dynamics,” in *Mobile Data Management (MDM)*. IEEE, 2018.

[22] S. Gaffney and P. Smyth, “Trajectory clustering with mixtures of regression models,” in *Knowledge discovery and data mining (SIGKDD)*. ACM, 1999, pp. 63–72.

[23] A. Nanopoulos, Y. Theodoridis, and Y. Manolopoulos, “C2p: clustering based on closest pairs,” in *VLDB*, 2001, pp. 331–340.

[24] Y. Wakuda, S. Asano, N. Koshizuka, and K. Sakamura, “An adaptive map-matching based on dynamic time warping for pedestrian positioning using network map,” in *Proceedings of the 2012 IEEE/ION Position, Location and Navigation Symposium*, 2012, pp. 590–597.

[25] J.-G. Lee, J. Han, and K.-Y. Whang, “Trajectory clustering: a partition-and-group framework,” in *Proceedings of the international conference on Management of data (SIGMOD)*. ACM, 2007, pp. 593–604.

[26] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, “Discovery of convoys in trajectory databases,” *Proceedings of the VLDB Endowment*, vol. 1, no. 1, pp. 1068–1080, 2008.

[27] N. Pelekis, P. Tampakis, M. Vodas, C. Doulkeridis, and Y. Theodoridis, “On temporal-constrained sub-trajectory cluster analysis,” *Data Mining and Knowledge Discovery*, pp. 1–37, 2017.

[28] M. Andriluka, S. Roth, and B. Schiele, “People-tracking-by-detection and people-detection-by-tracking,” in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.

[29] G. Shu, A. Dehghan, O. Oreifej, E. Hand, and M. Shah, “Part-based multiple-person tracking with partial occlusion handling,” in *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2012, pp. 1815–1821.

[30] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, “Multispectral pedestrian detection: Benchmark dataset and baseline,” in *Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1037–1045.

[31] M. Volkhardt, F. Schneemann, and H.-M. Gross, “Fallen person detection for mobile robots using 3d depth data,” in *International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2013, pp. 3573–3578.

[32] S. Wu, H.-S. Wong, and Z. Yu, “A bayesian model for crowd escape behavior detection,” *IEEE transactions on circuits and systems for video technology*, vol. 24, no. 1, pp. 85–98, 2014.

[33] V. Bastani, D. Campo, L. Marcenaro, and C. Regazzoni, “Online pedestrian group walking event detection using spectral analysis of motion similarity graph,” in *International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2015, pp. 1–5.

[34] R. Rupasinghe, S. Senanayake, D. Padmasiri, M. Ekanayake, G. Godaliyadda, and J. Wijayakulasooriya, “Modes of clustering for motion pattern analysis in video surveillance,” in *Information and Automation for Sustainability (ICIAfS)*. IEEE, 2016, pp. 1–6.

[35] F. Zanlungo and T. Kanda, “Do walking pedestrians stabily interact inside a large group? analysis of group and sub-group spatial structure.” in *CogSci*, 2013.

[36] D. Brščić, F. Zanlungo, and T. Kanda, “Modelling of pedestrian groups and application to group recognition,” in *MIPRO*, 2017, pp. 564–569.