Geoinformatica
https://doi.org/10.1007/s10707-019-00353-2

A versatile computational framework for group pattern mining of pedestrian trajectories

Abdullah Sawas[1] · **Abdullah Abuolaim**[1] · **Mahmoud Afifi**[1] · **Manos Papagelis**[1]

Received: 16 October 2018 / Revised: 12 March 2019 / Accepted: 9 April 2019 /
Published online: 30 April 2019

Abstract
Mining patterns of large-scale trajectory data streams has been of increase research interest. 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*. Traditional approaches to solve the problem adhere to strict definition of group semantics. That restricts their application to specific problems and renders them inadequate for many real-world scenarios. To address this limitation, we propose a novel versatile method, *timeWgroups*, for efficient discovery of pedestrian groups that can adhere to different pattern semantics. First, the method efficiently discovers *pairs of pedestrians* that *move together over time*, under varying conditions of space and time. Subsequently, *pairs of pedestrians* are used as a building block for effectively discovering *groups of pedestrians* that can satisfy versatile group pattern semantics. As such, the proposed method can accommodate many different scenarios and application requirements. In addition, we introduce a new group pattern, *individual perspective grouping* that focuses on how individuals perceive groups. Based on the new group pattern we define the concept of *dominant groups*, a global metric for defining important groups that respects the *individual perspective group* pattern. 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. Furthermore, a *query-based search method* is provided that allows for *interactive exploration and analysis* of group dynamics over time and space. In addition, a visual testing is performed on real motion video to assert the group dynamics discovered by our methods.

Keywords Trajectory mining · Group pattern mining · Pedestrian behavior

✉ Abdullah Sawas
asawas@eecs.yorku.ca

[1] York University, 4700 Keele St, Toronto, ON M3J 1P3, Canada

1 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 [37]. 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 [13, 30, 38]. 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 group patterns can support a variety of useful applications ranging from monitoring physical areas [8], such as shopping malls, train stations, and airports to supporting pedestrian behavioral studies [7, 19]. In fact, crowded scenes would render the analysis more challenging, as individuals are often intermixed with the crowd. Motion video analysis

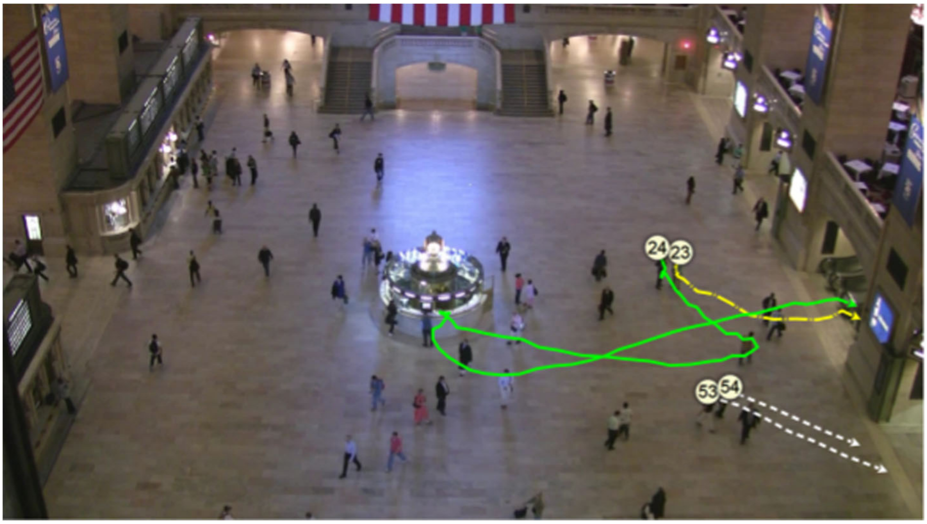


Fig. 1 Four pedestrian trajectories are highlighted in this scene of a train station. Pedestrians *#23* and *#24* met for some time and then exited the station following different routes. Pedestrians *#53* and *#54* walked and exited the station together

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 pre-processing phase 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* [5], for example, 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* [17], 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* [15] tries to merge the two concepts, such as the group consists of at least m same objects moving together for at least k consecutive time instants. The *gathering pattern* [41] is another group pattern that focuses on adjacent clusters that move close to each other over time. Recently, R. Lan et al. [18] 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 only gradually change over time. Similarly, Q. Fan et al. [10] introduced a platform for mining *co-movement patterns* in trajectories, which relaxes the "moving together" constraint, by allowing individual objects to join or leave a group at different times. This group pattern, is conceptually closer to the type of group patterns that we want to discover in this work. However, instead of designing a method that adheres to a strict definition of what consists a group, we propose a new method, *timeW-groups* that can accommodate variant group pattern semantics —the semantics still need to adhere to constraints of space and time.

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 [9], any of its variants or other from the rich literature [37]) 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 first discovers **pairs** of pedestrians that *spatially move together over a certain time interval*. Then, pairs are utilized as building blocks to define larger groups (see Section 2 for details). That way, our method provides more flexibility and allows to define more versatile strategies of group pattern discovery, including the proposed *individual perspective pattern* (see Section 4 for details).

It also allows to improve the time performance by considering only relevant pairs in the grouping phase. Towards this end, we present a *tensor-based* method for efficiently discovering pairs and groups of moving objects that are intentionally traveling together in space and time. Our method assumes 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 [3, 8]).

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 [22]. Therefore emphasis should be given in methods that efficiently find small clusters. Towards that end, we introduce a new group pattern, *individual perspective grouping* that focuses on how *individuals* perceive groups (i.e., own perspective). To motivate the new pattern, Fig. 2 shows four time snapshots of moving pedestrians (from top to bottom the snapshots are $S1$, $S2$, $S3$, and $S4$). Let us focus on two specific pedestrians, shown as *blue* and *yellow* colored dots. The *yellow* and *blue* pedestrians were members of the same group at only one time snapshot ($S3$). This is, in fact, the only time that the *yellow* pedestrian was ever grouped to other pedestrians, so this consists an important group for the *yellow* pedestrian. In the meantime, the *blue* pedestrian was consistently grouped with other pedestrians during the time snapshots ($S1$-$S3$), as depicted in Fig. 2a. If we were to apply the state-of-the-art or common methods for discovery of group patterns (e.g., *flock* or *evolving groups*), we would not be able to discover the group of the *yellow* pedestrian. In fact, the *yellow* pedestrian was member of only one group, as depicted in Fig. 2b. This individual perspective group pattern offers more insights and is meaningful for many applications, such as surveillance video analysis and anomaly detection. Based on the new group pattern we also define the concept of *dominant groups*. As will see later, this is referring to a global metric for discovery of important groups that at the same time respect the individual perspective group pattern.

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 [40].

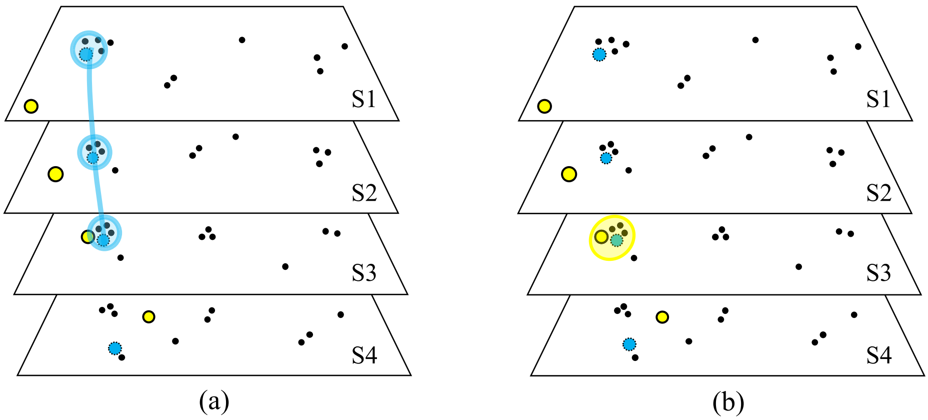


Fig. 2 Two examples of the *individual perspective group* pattern. **a** Groups discovered from the perspective of the *blue* pedestrian. **b** Groups discovered from the perspective of the *yellow* pedestrian

In summary, the major contributions of this work include:

- A novel time-window based versatile method, *timeWgroups*, for efficiently and effectively finding groups of pedestrians with variant group pattern semantics.
- A new *individual perspective group pattern* that considers how individuals perceive groups.
- Introducing the concept of *dominant groups*, for finding groups of pedestrians over a given time period.
- 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 online[1].

An earlier version of this work appeared in the proceedings of the IEEE International Conference on Mobile Data Management (IEEE MDM 2018) [27]. The current journal version provides a more complete coverage of the problem, introduces a novel method and extends the experimental evaluation to offer substantial, new insights of the problem's complexity and the versatility of our computational framework. In particular, this version introduces a new group pattern, *individual perspective grouping*, which focuses on how individuals perceive groups. A new metric is also proposed that identifies and characterizes what the *dominant groups* of an individual are. We experimentally evaluate the accuracy performance of the new method and demonstrate its effectiveness in finding dominant groups on a real data set *Student003* [21]. Moreover, a systematic method is presented for determining the optimal hyper-parameters of the new method. This version provides more examples throughout the manuscript to motivate the contributions and better illustrate dimensions of our work. The related work is also extended to provide a more complete coverage of the problem and its variations. To improve reproducibility of our methods, the source code of the proposed methods are publicly provided, as well as, all the datasets employed in our experimental evaluation.

The remainder of this paper is organized as follows: Section 2 introduces notation and formally defines the problem. Our versatile grouping method and overall computational framework are presented in Section 3. In section 4, the *individual perspective group pattern* is explained. Section 5 provides details of our experimental evaluation. In Section 6 we present an interactive tool for exploration of group dynamics. After reviewing the related work in Section 7, we conclude in Section 8.

2 Problem definition

In this section, we state the problem by giving the definitions of *pedestrian groups* and *pairs* that we aim to find. Table 1 lists the symbols we will use and their meanings. Formally, given (x, y) coordinates of the motion trajectories for N pedestrians over a set of V time snapshots, we define the following.

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

[1] https://sites.google.com/view/pedestrians-group-pattern/

Table 1 Table of notations

Indices:			
t	Given time point	$\mathbf{x}_t, \mathbf{y}_t$	x and y coordinates of all pedestrians' trajectories at time t
i, j	Indices representing pedestrians	D_l	Distance between two pedestrians
Sets:			
G	Pedestrian group	D_g	Global distance between two pedestrians' trajectories
V	Time Period	D_w	Maximum distance over w time window
$\mathbf{G}_i$	Set of pedestrians grouped with i during the entire period pedestrian i appeared in	$\mathbf{p}_i$	Trajectory of pedestrian number i
Parameters:		$\mathbf{M}$	Sparse tensor contains all pedestrians' data over a certain period of time
τ	Proximity threshold	$\mathbf{R}_i$	The pedestrian i perspective grouping record
w	Time window size	$\mathbf{P}_i$	The pedestrian i perspective grouping probability record
κ	Selectivity parameter of the individual perspective group pattern	N, n	Number of pedestrians
Metrics:		c_i	Number of time snapshots the pedestrian i was grouped with other pedestrians
ANG	Average number of identified groups (in the video) per time unit according to certain proximity distance threshold τ	$c_{i,j}$	Number of time snapshots that both pedestrians i and j are members of the same group
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	**Functions:**	
ρ	Density of groups	Φ	Mapping function between pedestrian IDs and indices in the tensor
Variables:			
(x_i, y_i)	Pedestrian i position in the 2D Cartesian coordinate system		

a coherent pair, *iff* the average distance between them, over the entire given time, is below a fixed threshold. Intuitively, 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, where each member of the group belongs to one or more *coherent pairs* and each coherent pair shares at least one pedestrian with at least another coherent pair, i.e. let $G = \{1, 2, ..., k\}$ be the members of a coherent group, then $\forall i \in G, \exists j, m \in \{1, ..., k\} : (i, j)$ and (i, m) are both 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 the time window w. 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.

Definition 5 (Individual perspective group) given a pedestrian i, let c_i be the number of times the pedestrian i is grouped with other pedestrians. Let also $c_{i,j}$ be the number of time snapshots that both pedestrians i and j are members of the same group. Given a selectivity parameter κ, we define the individual perspective groups of i, with respect to κ, to be the pedestrians j, such as $c_{i,j} \geq \kappa$. Effectively, this group pattern represents the pedestrians that a specific pedestrian i has interacted the most with, during i's trajectory.

Definition 6 (Dominant groups) given a set of pedestrians $i_1, i_2, ...i_k$ over an observation time V, the concept of *dominant groups* describes the most prominent groups of all pedestrians, as identified using the individual perspective group pattern. In practice, it defines globally important groups by aggregating information of already discovered individual perspective groups. Section 4.2 provides details about the systematic approach followed to determine the dominant groups.

As we process a large amount of data, we employ an efficient data structure to store and represent the trajectories. 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 pedestrian trajectories over V time snapshots. At any given time 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 $\mathbf{p}_i \in \mathbf{P}, i \in [1...N]$, find the *coherent pairs* and *coherent groups* of pedestrians in a certain time interval.

Problem 2 Given a set of pedestrian trajectories $\mathbf{p}_i \in \mathbf{P}, i \in [1...N]$, find the *pedestrian pairs* and *pedestrian groups* in a certain time interval.

Problem 3 Given a set of pedestrian trajectories $\mathbf{p}_i \in \mathbf{P}, i \in [1...N]$, find the *individual perspective groups* of each pedestrian i in a certain time interval.

Problem 4 Given a set of pedestrian trajectories $\mathbf{p}_i \in \mathbf{P}, i \in [1...N]$, find the *dominant groups* of all pedestrians in a certain time interval.

Note that the main objective is to accommodate discovery of pedestrian groups that satisfy variant definitions of a group. While the *evolving group* pattern might be a good candidate group pattern to describe pedestrian groups, another group pattern might be more appropriate for specific domain applications. For example, in criminal investigation, it might be more interesting to sieve through video archives and find activities of coherent pairs of pedestrian that have occurred in the past. Accordingly, the *flock pattern* would be more suitable to describe this kind of pedestrian group activity. Although *flock pattern* is useful to report the coherent pairs or groups during a given time period, it can still fail to discover some groups or group members due to gathering/dispersion activity in groups. Our method, *timeWgroups* can support a more versatile analysis of group patterns within the same computational framework. In addition, our proposed *individual perspective pattern* and the concept of *dominant groups* provides the means to discover pedestrian groups that take into consideration an individual's perspective, offering more rich insights in the analysis.

3 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 7, 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 using the same computational framework.

3.1 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:

3.1.1 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 $D_l(\mathbf{p}_i^t, \mathbf{p}_j^t)$ (1) between two trajectories $\mathbf{p}_i$ and $\mathbf{p}_j$ in space such that:

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

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 5.

3.1.2 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. 3). 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 are considered as a pair if their distance is below a predefined threshold τ.

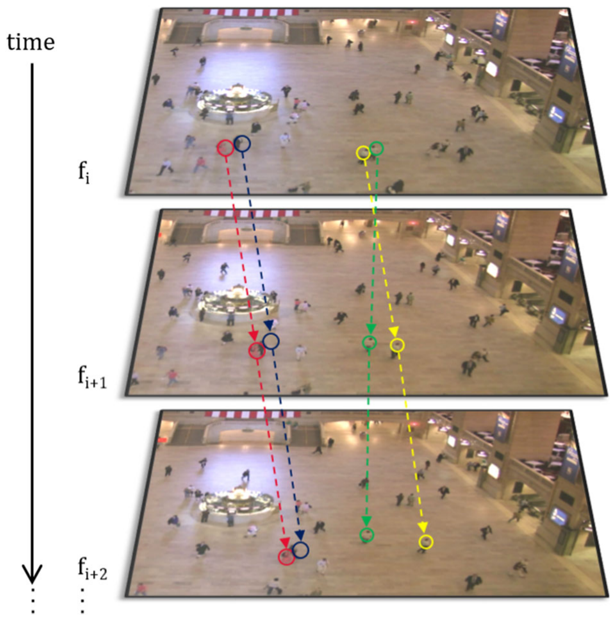


Fig. 3 *Coherent pair* follows similar path over time (red/blue arrows). Non-coherent pair (green/yellow arrows)

Table 2 Comparison between the three approaches: ***locTgroups***, ***globTgroups***, and ***timeWgroups***

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

3.1.3 Time window based pairing of trajectories (*timeWgroups*)

For the previous two approaches, there are some strengths and weaknesses summarized in Table 2. The local spatial approach (*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 approach (*globTgroups*) is efficient in finding *coherent pairs* in space and time. However, this method runs in as an offline batch process, as it requires the entire data 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 approaches, the task is now motivated by finding a way to overcome these limitations. Towards this end, we propose a novel method, *timeWgroups*. The method finds pairs efficiently that respect spatial and temporal constraints, and naturally captures a variant of group 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:

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

It is important to note that our *timeWgroups* method can be configured to perform exactly as the *locTgroups* and *globTgroups* methods. In particular, when $w = 1$, the *timeWgroups* instantiates to *locTgroups* method, and when $w = V$, (V is the entire video time snapshots), it instantiates to *globTgroups*.

3.2 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.

Algorithm 1 Grouping of pedestrian pairs.

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

The algorithm starts by initializing two global variables **P** and V in lines 2 and 3, respectively. **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 **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).

Our approach of first discovering pairs over time and then utilizing pairs as building blocks to define larger groups allows to define more versatile strategies of group pattern discovery. In addition, it can be tweaked to provide support for diverse group semantics that appear in the literature.

For instance, our global group pattern *globTgroups* can be employed to capture semantics of a flock group pattern. Recall that a flock is defined as a disk of radius r that contains at least m same objects for every point in a period of time (it is typical for flock to consider the whole observation time). It resembles a group of trajectories that stay and move together, as a cluster, under a predefined threshold of distance, over a certain time period. Once all groups are identified by *globTgroups*, a meta-analysis is required as there is no direct mapping between the threshold τ used in the method to define pair-wise groupings and the radius r of a circle that defines a flock. In order to establish this connection, we reduce the meta-analysis to the *smallest-circle problem* or *minimum covering circle problem*. This is the problem of computing the smallest circle that contains all of a given set of points in the Euclidean plane. In practice, since all pair-wise distances have already been computed, for each group with m objects, we first obtain the distance d of the two objects (out of m) that are farther away. By definition, these points lie on the circumference of the smallest circle that encloses all objects in the group. This circle represents a flock with disk radius $r = d/2$ and m objects. Performing the same analysis for each group will provide all the flocks that can be defined for a varying disk radius r. If we need to control the number of objects in a group, then this information needs to be taken into account when we recursively group pairs (see Algorithm 1). Similarly, our *timeWgroups* method can be adjusted to capture various local and global group pattern semantics by varying the proximity and duration parameters. A visual summary of some group patterns that can be instantiated using our versatile method is shown in Fig. 4.

While our method is versatile enough to accommodate diverse group semantics, the mapping process is not always straightforward. This is because many of these patterns can be very restrictive/specific. The main claim of our work is that an analysis that begins with diverse definitions of what consists a pair, and obtains all these pairs efficiently, can provide the base for more rich analysis of diverse group pattern semantics that can accommodate different scenarios and application domains. This is in contrast to specialized methods that assume input parameters are known a priori and are provided to the system.

3.3 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

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

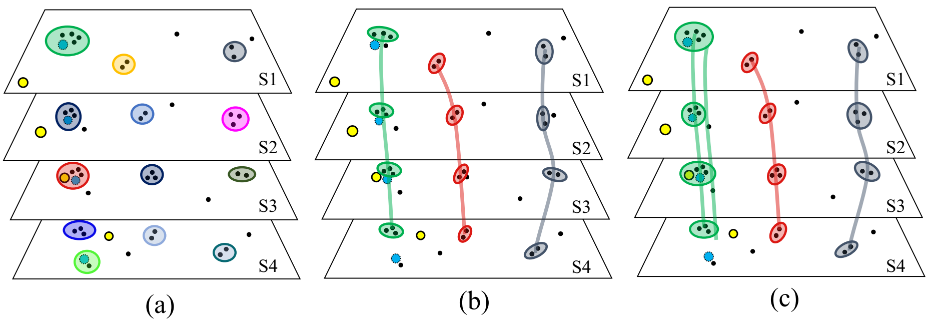


Fig. 4 Our proposed method can be configured to discover groups of variant group patterns: **a** local groups that are defined without taking into account the temporal information. **b** groups that follow the *flock* pattern, where group members move consistently together over time. **c** groups that follow the *evolving group* pattern that allow members to gather/disperse over time

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 Eqs. 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:

$$\mathbf{X}_t = \left[\, \mathbf{x}_t \;\; \mathbf{x}_t \;\; ... \;\; \mathbf{x}_t \,\right], \tag{4}$$

$$\mathbf{Y}_t = \left[\, \mathbf{y}_t \;\; \mathbf{y}_t \;\; ... \;\; \mathbf{y}_t \,\right]. \tag{5}$$

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.

4 Individual perspective group pattern

The proposed method described in the previous section can effectively find primitive groups of pedestrians over time which can be used to find more sophisticated patterns, such as *individual perspective groups* and *dominant groups*. To the best of our knowledge, the aforementioned patterns are novel, not studied in the literature before in the context of group pattern discovery. The following figures, Figs. 2 and 4 illustrate different patterns including the newly proposed patterns.

4.1 Individual perspective grouping

To this end, we need to first find a link between pedestrians that are members of the same primitive groups in successive time points. Through this linking process we are able to assign a score for each pair of pedestrians based on the number of time points they were primitively grouped together.

More specifically, given a pedestrian i, we assign pedestrian i with a list of other pedestrians $\mathbf{G}_i$ who were primitively grouped with i during the entire period pedestrian i appeared in. Formally, let $\mathbf{R}_i$ be the pedestrian i perspective grouping record where each element $\mathbf{R}_{i,j}$ counts the number of times the pedestrian pair (i and j) appeared together in the same primitive group at an instance of time. In addition, we keep a record c_i of how many times the pedestrian i was primitively grouped with others.

Assume that we are interested in finding the perspective groups of pedestrian i from his own point of view. We achieve that by computing the probability of other pedestrians being in the perspective groups of pedestrian i. Normalizing the elements of $\mathbf{R}_i$ by c_i, we get the perspective grouping probability record $\mathbf{P}_i$ of the pedestrian i. Hence, $\mathbf{P}_{i,j}$ represents how likely it is that a pedestrian j will be grouped with the pedestrian i from the perspective of i. In other words, $\mathbf{P}_{i,j}$ represents how importance the pedestrian j to i from the perspective of i.

Figure 5 illustrates an example of three pedestrians (*yellow*, *red*, and *blue*), where the *yellow* and *red* pedestrians spent with each other 10 minutes, while each of them met the *blue* pedestrian for only 30 seconds. Assuming that the time point is represented by 1 second, we can group those pedestrians by applying the grouping method proposed in Section 3 using a proper proximity distance τ and time window w. As a result the *yellow* and *red* pedestrians are grouped together for $10 \times 60 = 600$ times. The *blue* and each

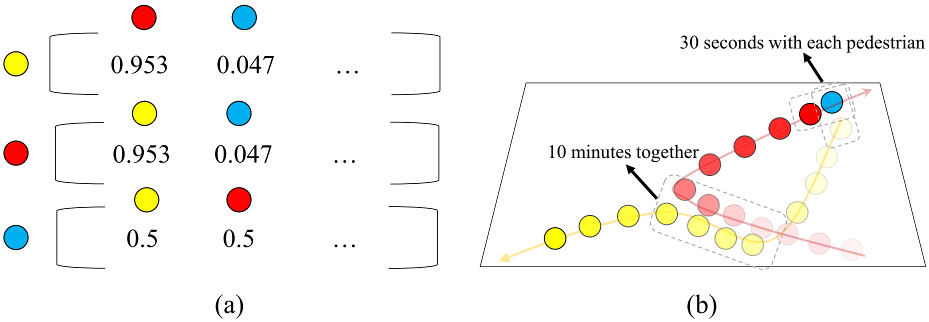


Fig. 5 An example of the individual perspective grouping probability record. **a** The individual perspective grouping probability records for each pedestrian reflects how important each other pedestrian (as a group member) is to the current pedestrian. **b** Shows the details of each pedestrian trajectory over the time

of *red* and *yellow* pedestrians are grouped together only for 30 times. By plugging these numbers to build the individual perspective grouping records $\mathbf{R}_{\text{red}}$, $\mathbf{R}_{\text{yellow}}$, and $\mathbf{R}_{\text{blue}}$, we have $\mathbf{R}_{\text{red}} = [600, 30]$, $\mathbf{R}_{\text{yellow}} = [600, 30]$, and $\mathbf{R}_{\text{blue}} = [30, 30]$ for the *red*, *yellow*, and *blue* pedestrians, respectively. Since $m_{\text{red}} = m_{\text{yellow}} = 630$ and $m_{\text{blue}} = 30$, the individual perspective grouping probability record of the *red*, *yellow*, *blue* pedestrians will be $\mathbf{P}_{\text{red}} = [0.953, 0.047]$, $\mathbf{P}_{\text{yellow}} = [0.953, 0.047]$, and $\mathbf{P}_{\text{blue}} = [0.5, 0.5]$, respectively.

Intuitively, both *red* and *yellow* pedestrians are equally important from the *blue* pedestrian's point of view. However, the *blue* pedestrian is less important from both the *yellow* and *red* pedestrians point of views, as she only spent very short amount of time as a group member with each of them. Figure 5a shows that the individual perspective grouping probability record of each pedestrian reflects this relation between each pedestrian by giving high probability to the most important pedestrians from each pedestrian point of view.

4.2 Dominant groups pattern

So far, we have discovered the individual perspective groups of a certain pedestrian i. To find the *dominant groups pattern* we will extend this concept to a set of pedestrians as follows: first we compute the pairwise averages of each individual perspective group probabilities as follows:

$$\overline{\mathbf{P}}_{i,j} = \frac{\mathbf{P}_i(j) + \mathbf{P}_j(i)}{2}. \tag{7}$$

Subsequently, we combine the pairwise probabilities in a single vector Γ that will be used to compute the threshold value that achieves the selectivity value κ using the probability quantile. In other words, we compute the threshold value that discards the low $\kappa\%$ of the entries in Γ. The computed threshold is used later to select the dominant groups from the pairwise perspective grouping probability records $\overline{\mathbf{P}}$. The essence of constructing the vector Γ is to take into consideration the correlation between the groups in a certain set of pedestrian trajectories.

5 Experimental evaluation

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

Table 3 Details of the two datasets used for evaluation

Dataset	*Pedestrian walking route* [36]	*Student003* [21]
Resolution (px)	$1,920 \times 1,080$	720×576
Annotated frame count	6,000	900
Annotated pedestrian count	12,684	434
Average pedestrian number per frame	123	38.93
Max pedestrian number per frame	332	51
Number of data points	1,266,502	33,792

Data To evaluate and validate our methods, two datasets are employed, namely *Pedestrian Walking Route* dataset introduced by Shuai Yi et al. [36] and *Student003* dataset introduced by Lerner et al. [21]. Both datasets contain a real-scene crowd data. In the case of the *Pedestrian Walking Route* dataset, the trajectories were 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. On average there were around 123 pedestrians per video frame with the most crowded frame containing 332 pedestrians. In the case of the *Student003* dataset, trajectories of pedestrians were extracted from 220 seconds of video recorded by a surveillance camera that captures walking students at a University walkway. The data points of moving pedestrians and groups ground truth were manually annotated at each single frame. It is worth mentioning that pedestrian trajectory is usually collected using GPS and/or some pedestrian tracking techniques in video streams. Table 3 provides a summary of these two datasets. Note that a number of visualization videos of the following experiments are also available online[2] (Fig. 6).

Pre-processing 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 [11] as suggested by [1]. 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. 7). As shown in Fig. 7, 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

[2] https://sites.google.com/view/pedestrians-group-pattern/

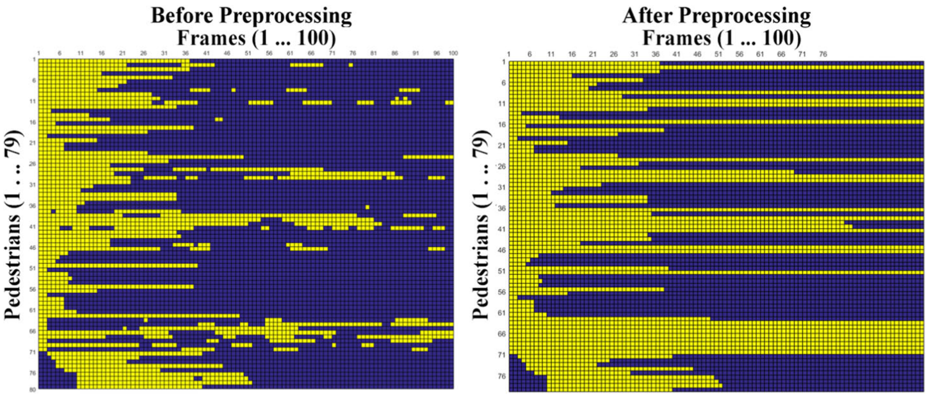


Fig. 6 Left and right maps show the trajectories before and after filling the unreported stationary positions respectively. Yellow cell indicates the reported (x, y) coordinates of moving pedestrian and the blue cell indicates the unreported position of stationary pedestrian

unreported position of stationary pedestrian. Figure 6 shows some pedestrian trajectories before and after filling the unreported stationary positions.

Recall that the w parameter of our versatile method *timeWgroups* can be configured to take any value in the range $1 \leq w \leq V$. As discussed in Sections 3 and 4, it can be instantiated to discover local groups (i.e., $w = 1$) as in *locTgroups*, to discover *coherent groups* for the whole observation time V (i.e., $w = V$) as in *globTgroups*, or anything in between. Lastly, the terms *perBgroups* and *perDgroups* refer to the post-processing steps to find *individual perspective groups* and *dominant groups*, respectively.

Evaluation Criteria We have evaluated the proposed method 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.

Fig. 7 High perspective distortion in the *Pedestrian Walking Route* dataset. The pedestrian closer to camera is about two times larger than the one farther

– 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.

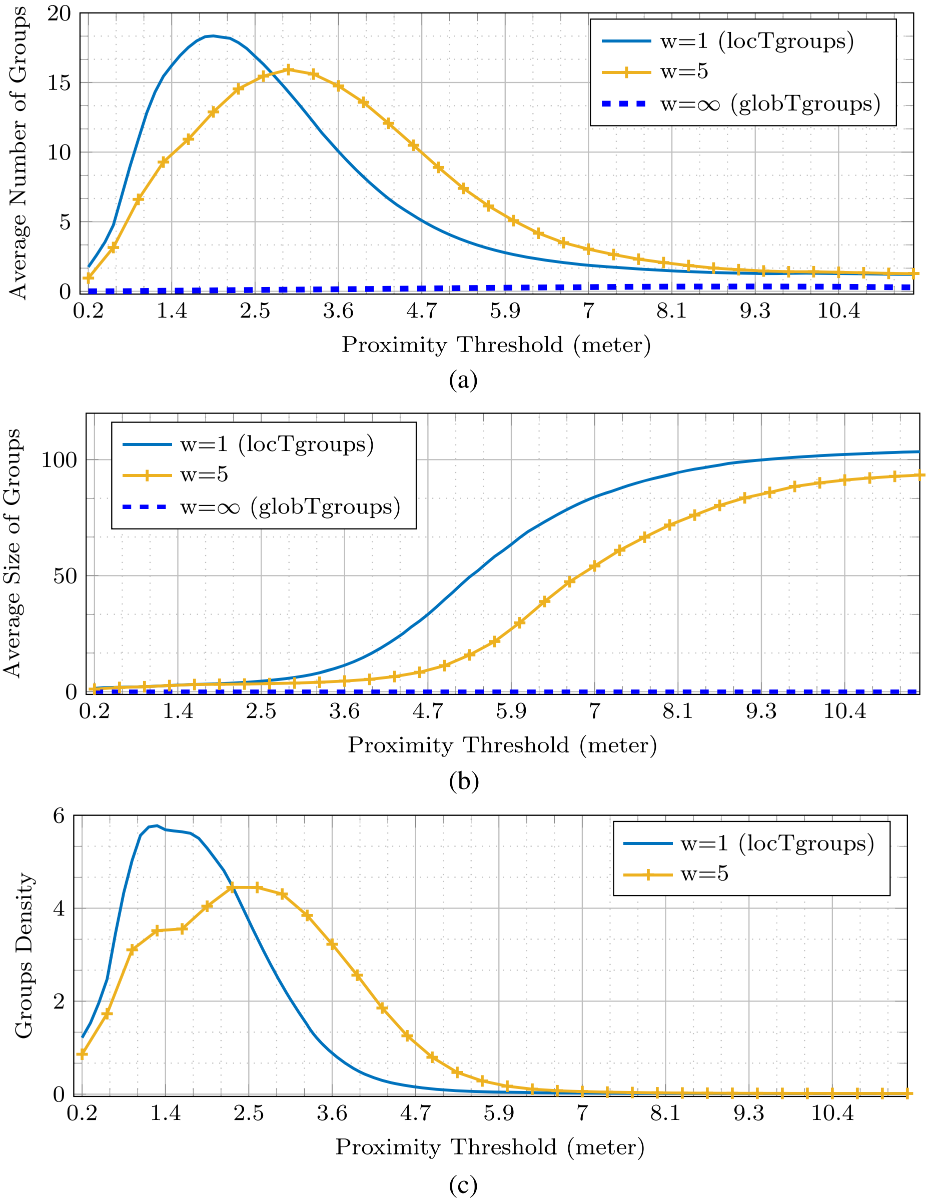


Fig. 8 Analysis of the proximity threshold (meters) impact for the three proposed approaches on the *Pedestrian Walking Route* dataset: **a** the average number of identified groups per time unit, **b** the average group size per time unit, **c** the density of the groups

– ρ: Density of groups which is defined as

$$\rho = \frac{AGS}{ANG}. \tag{8}$$

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

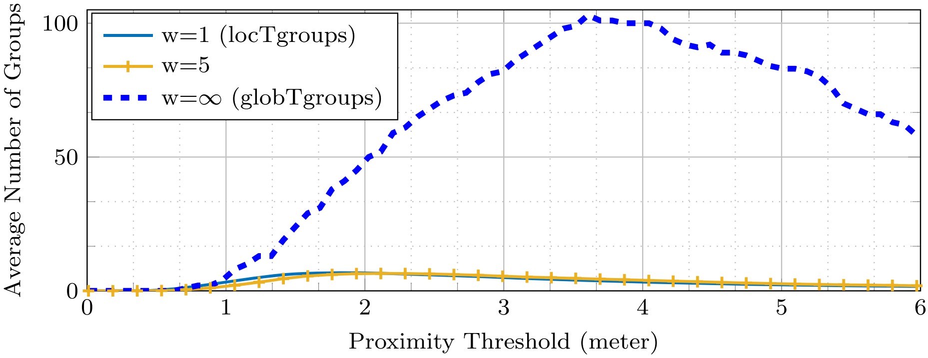
5.1 Determination of threshold (τ)

Considering *only* the spatial information of the pedestrians, we can cluster them into groups based on the distance among them using three approaches ($locTgroups$, $globTgroups$, and $timeWgroups$). However, deciding the minimum proximity (distance) τ is a critical factor that affects the pedestrian groups discovered by each approach. Therefore, in order to find the suitable value of τ, we evaluated the three grouping approaches for different values of threshold τ (ranging from 0.2 to 22 meters). The effect of changing τ on the proposed approaches is shown in Figs. 8 and 9 for the *Pedestrian Walking Route* and the *Student003* datasets, respectively. The proximity threshold τ was represented in meters by estimating 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).

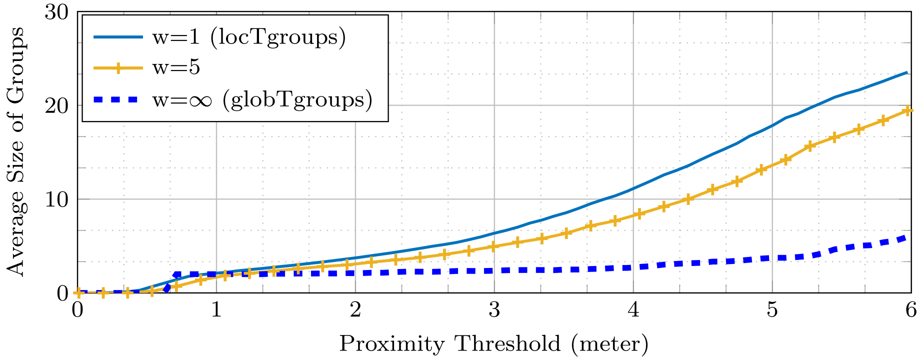
As shown in Figs. 8a and 9a, for small values of τ the average number of identified groups ANG of the pedestrian trajectories is small since pedestrians that are not very close to each other are not going to be grouped. ANG keeps increasing with increasing τ until it reaches a peak (at $\tau = 2, 3, 9.5$ meters for $locTgroups$, $timeWgroups[w = 5]$, and $globTgroups$ approaches, 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 Figs. 8a and 9b 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* approach is very low ($\approx$ 2.2) compared to the other approaches. This is due to the fact that normal pedestrians movement is hardly ever coherent in larger groups.

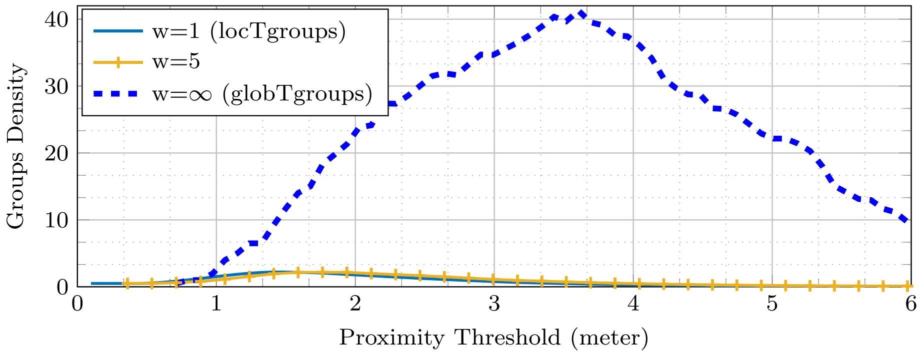
Interestingly, the density of the groups (ρ) in both Figs. 8c and 9c shows a peak at proximity threshold $\tau = 1.5$ meter for the *locTgroups*. It worth mentioning that the curve corresponding to the *globTgroups* mehtod is not shown in the density figure because the values of ANG were very closed to zero. Only in Fig. 8c, it shows 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. [22] that found that the average number of pedestrians in groups is usually less than or equal to four pedestrians. Finally, the *coherent groups*, captured by the *globTgroups*, were excluded from density calculation in Fig. 8c 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; this refers to pedestrians that intentionally walk together with a distance below a certain threshold over the entire time interval and usually consists of just two pedestrians, i.e. *coherent pairs*.



(a)



(b)



(c)

Fig. 9 Analysis of the proximity threshold (meters) impact for the three proposed approaches on the *Student003* dataset: **a** the average number of identified groups per time unit, **b** the average group size per time unit, **c** the density of the groups

5.2 Determination of window size (*w*)

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

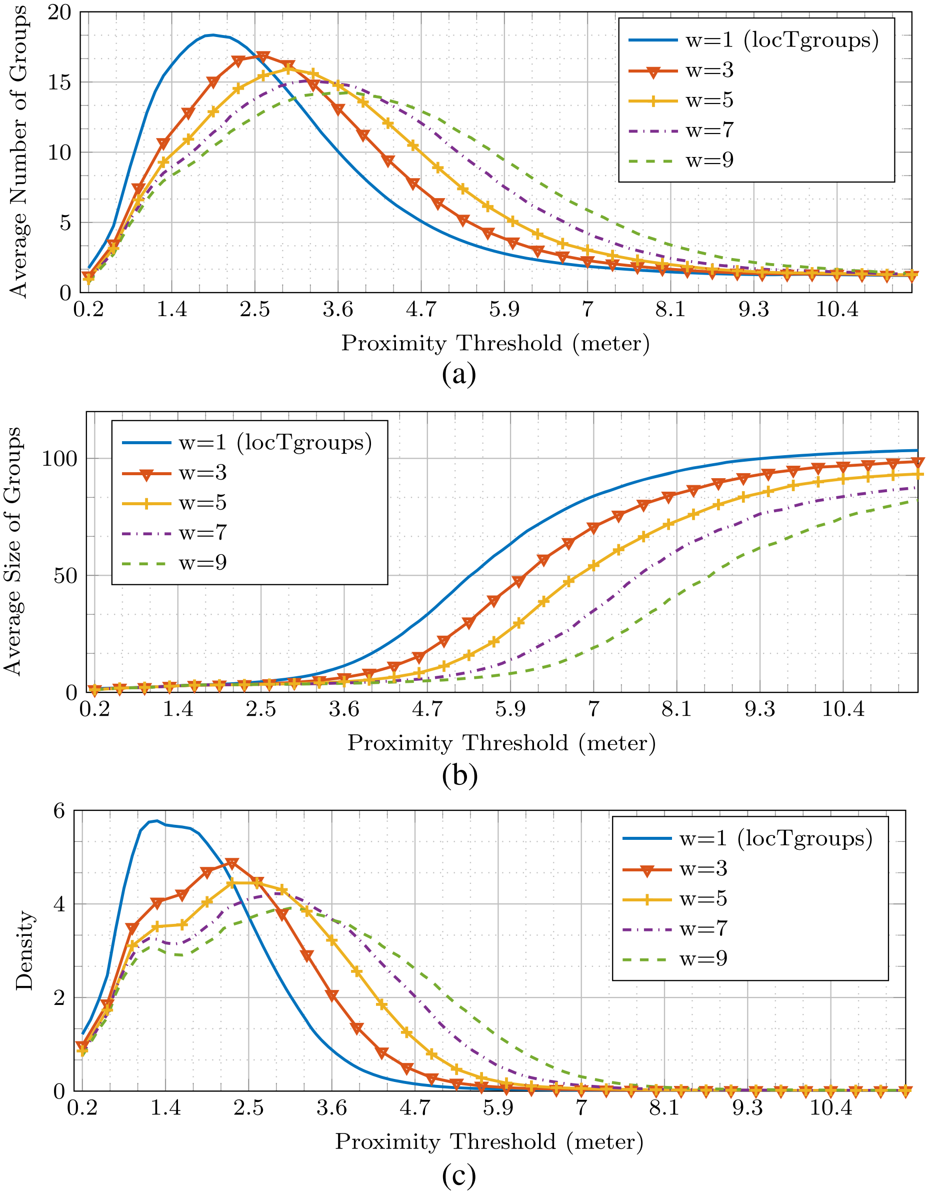


Fig. 10 Analysis of window size and threshold impact for the *timeWgroups* proposed method on: **a** the average number of identified groups per time unit, **b** the average group size per time unit, **c** the density of the groups

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. 10 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. [22]. That indicates the effectiveness of the *timeWgroups* in effectively finding *evolving groups*.

5.3 Time performance

Based on the tensor-based optimization presented in Section 3.3, the proposed method can be ran efficiently in a reasonable execution time. Figure 11 shows the total execution time for the proposed approaches 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 pedestrians are considered as pairs, resulting in longer processing

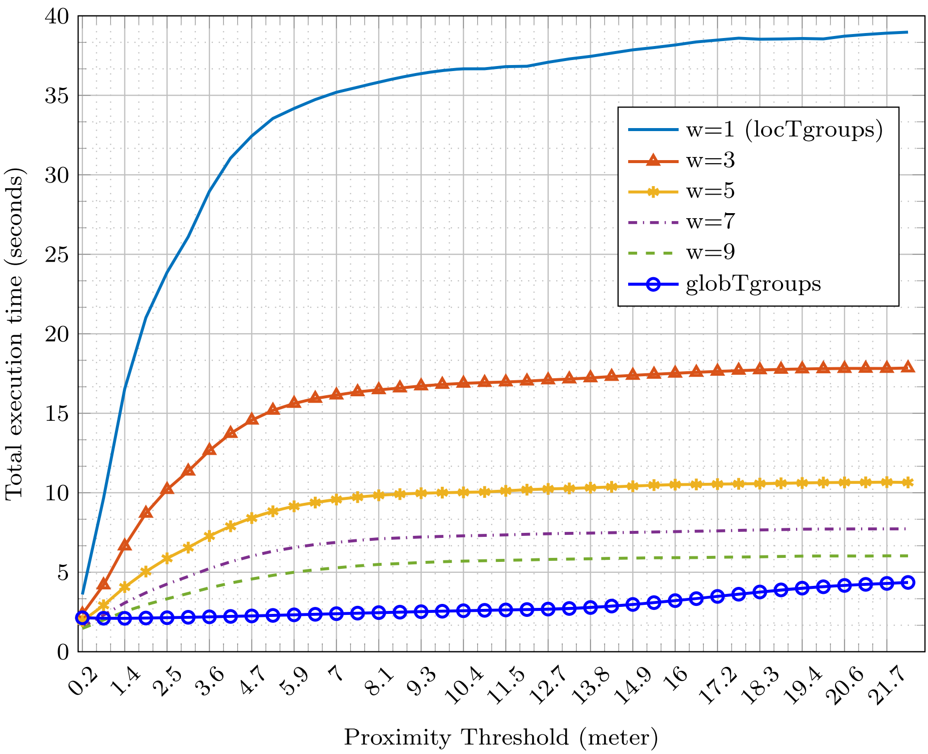


Fig. 11 Execution time of the proposed approaches for finding the pedestrian groups in the **entire** video (≈ 1 hour) for varying proximity threshold τ

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.

5.4 Dominant group identification

Dominant groups are identified based on the individual perspective group pattern. In this section, we discuss the effect of the selectivity parameter on the number and size of the dominant groups identified by our method. In addition, we evaluate the performance of our proposed method, *perDgroups*, on identifying dominant groups in *Student003*, a real dataset that provides ground truth labeling. We compare the performance of our method to state-of-the-art alternatives from the literature. Before we present the details of the analysis, we provide an illustrative example that simplifies the individual perspective group patter.

Illustrative example The example is based on careful selection of a set of eight video frames coming from the *Student003* real dataset as shown in Fig. 12 (i.e., frames #300, #388, #400, #416, #428, #452, #598, and #634). In these frames, there are two groups annotated, the *red* group and the *white* group. In addition, an individual person is annotated with a *yellow oval dashed outline*. As shown in the sequence of the video frames, the *yellow* pedestrian only spent a limited amount of time being in the proximity of the pedestrians that are members of the *red* and *white* groups. As such, when we consider our individual perspective grouping method from the perspective of any of the members of the *red* or *white* groups, the *yellow* individual will not be part of the defined groups. On the other hand, when we consider a grouping from the perspective of the *yellow* pedestrian, it will appear as if she was member of the *white* group (rather than the *red* group), since she spent substantial time of her trajectory being close to that group. This illustrative example was obtained by running our method described in Section 4.

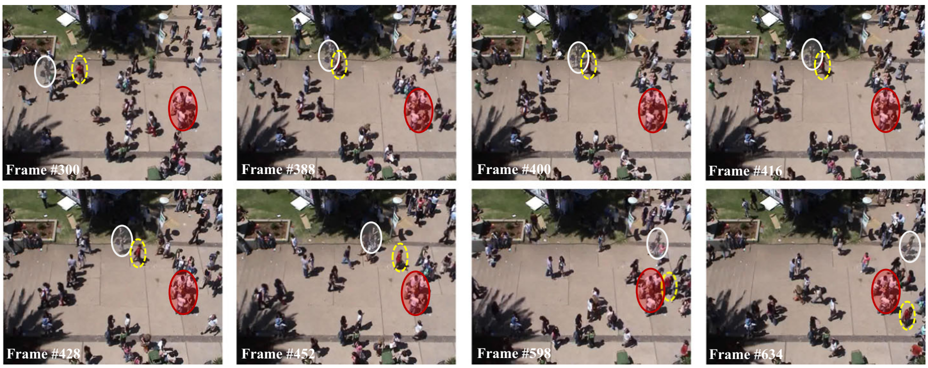


Fig. 12 Illustrative example of the individual perspective group pattern coming from the *Student003* dataset. Depending on which pedestrian's perspective is considered, different groups are identified. For example, from the perspective of the *yellow* pedestrian she will be grouped with the *white* group due to her proximity to its members in some frames. On the other hand, from the perspective of any pedestrian in the *white* or *red* groups, the *yellow* pedestrian won't be part of their dominant groups, as they didn't spend as much time being close to each other, as with other pedestrians

5.4.1 Analysis of the selectivity parameter

In order to study the impact of the selectivity parameter κ, we consider values in the range of 0 to 1. The selectivity parameter acts as a threshold for filtering pairs of pedestrians based on the frequency with which each pair is found in a group. Any of the grouping methods presented (*locTgroups* and *timeWgroups*) can be used to identify groups. In principle, the smaller the selectivity parameter value is, the less frequent a pair of pedestrians need to be in order to define a group. On the other hand, values of κ near 1 will constrain a pair of pedestrians to be considered a group, if only they are moving together for a longer time.

In Fig. 13a, we show how the number of dominant groups changes with varying values of the selectivity parameter. By looking at the blue (triangles) curve (at $\tau = 0.8$), the maximum number of dominant groups can be achieved at lower value of the selectivity parameter ($\kappa \approx 0.4$). In other words, when imposing a very small proximity threshold ($\tau = 0.8$) between group members, a smaller selectivity value is needed to maximize the number of dominant groups. That said, we need to allow less travel-together time within the proximity distance w.r.t the total travel time; that will provide flexibility to the group members to travel beyond the proximity distance. On the other hand, the green (circles) curve (at $\tau = 3.43$) can achieve the maximum number of dominant groups at a higher value of the selectivity parameter ($\kappa \approx 0.96$). Higher values of the selectivity parameter means that the group members should

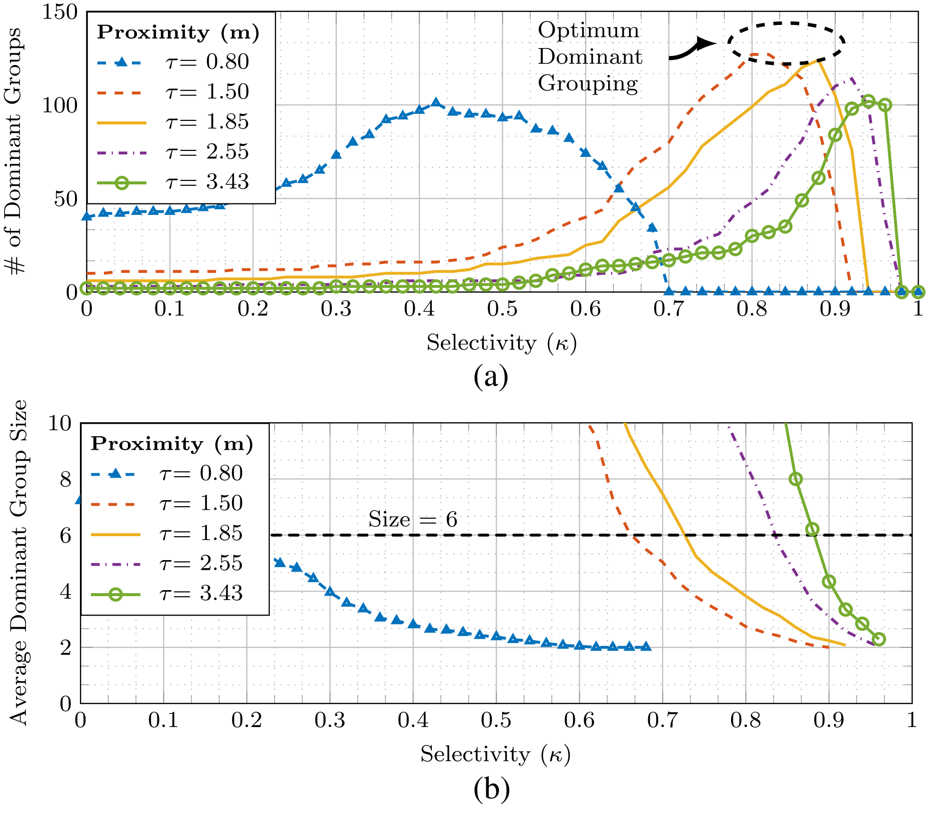


Fig. 13 *Student003* dataset: **a** number of identified groups for different values of the selectivity parameter, **b** average dominant group size for different values of the selectivity parameter

travel most of the time within the proximity distance and leaving the proximity circle will be costly.

As far as the dominant group size concerns, Fig. 13b demonstrates that the average dominant group size is decreasing as the value of the selectivity parameter increases, which is intuitive. Besides, the lower the proximity threshold the earlier that decrease occurs (see for example the blue (triangles) curve at $\tau = 0.8$ as it crosses the black dashed line at a selectivity parameter value smaller than 0.2). Therefore, imposing higher value for the selectivity parameter will result in many small size groups. The figure shows a horizontal line at size of six (6), this is to compare with other studies such as [22], which estimates the average size of pedestrian groups very rarely includes more than six (6) pedestrians. Using this as a guideline, the search space in Fig. 13 can be limited to groups of size below that threshold.

5.4.2 Parameter selection

As mentioned earlier, there are two parameters, namely proximity threshold (τ) and selectivity parameter (κ) that control *perDgroups* performance in finding dominant groups. To show the direct effect of these two parameters, we plot in Fig. 14 the precision heat map by varying κ ($x-$axis) from min to max ($0 - 1$) and τ ($y-$axis) within a reasonable range of proximity ($0 - 8$ meters). The heat map nicely demonstrates the effect of *perDgroups*'s two parameter values to the precision. The black dashed ellipse indicates the area of the highest precision. This is represented by a proximity value in the range of $1 - 2$ meters and by selectivity values in the range of $0.80 - 0.95$. This is in accordance to our analysis in Fig. 13a which shows a peak at ($\tau = 1.5$ and $\kappa = 0.82$) of number of dominant groups.

5.4.3 Clustering evaluation criteria

Given a desired ground truth pedestrian groups set $\hat{\mathbf{C}}$ and a set of grouping result $\mathbf{C}_i$, the problem of evaluating the dissimilarity between the two clusterings can be formulated using

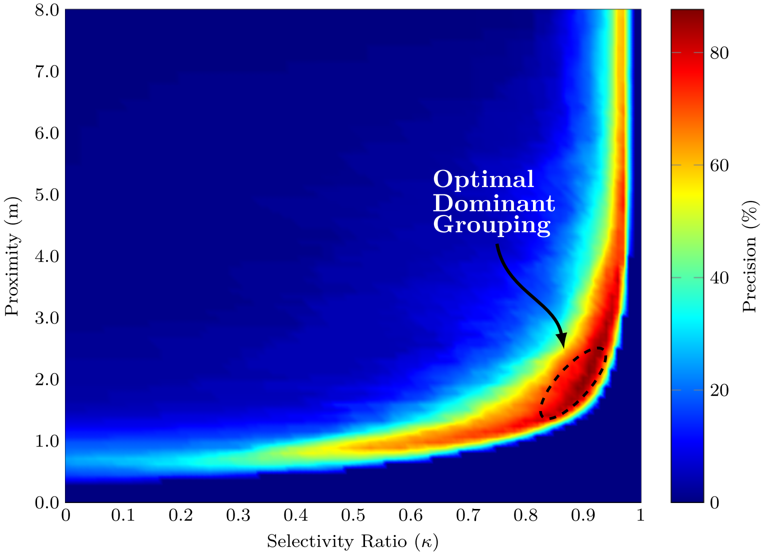


Fig. 14 *Student003* dataset: Precision heat–map of the extracted dominant groups found at varying *proximity* and *relative selectivity* parameters. Area in red color indicates high accuracy compared to area in blue color

Table 4 Accuracy performance of our proposed dominant groups method against the state-of-the-art method presented in [31]

Method	Dominant groups	Solera [31]
Precision	**91.2**	81.7
Recall	**92.7**	82.5

Results are based on the real dataset *Student003* [21] using *G*-MITRE loss Δ_{GM} introduced in [31]

a loss function. A common choice of loss function is *the pair-wise loss*, which is a generalization of the *Rand coefficient* [32]. This loss function is defined as the ratio between the number of pairs on which $\hat{\mathbf{C}}$ and $\mathbf{C}_i$ disagree on their cluster element membership and the number of all possible pairs of elements in the set. However, due to the quadratic number of pairs that exist among pedestrian groups, this metric tends to be imprecise when dealing with large number of groups. This issue is also highlighted in [31] where the authors proposed a different loss function, called *GROUP-MITRE* loss function, denoted for the rest of the manuscript as (*G*-MITRE) $\Delta_{GM}(\hat{\mathbf{C}}, \mathbf{C})$. We adopt the same metric for evaluating the quality of our group method, so results are directly comparable to those reported by a the state-of-the-art described in [31]. In short, in the *G*-MITRE loss function, groups are represented as spanning trees, while a group partition resembles a spanning forest. The loss score is computed by counting the number of links that need to be added or removed to transform the spanning forest representing a proposed clustering partition into the spanning tree corresponding to the ground truth. Besides considering the links between the members in a group, the *G*-MITRE loss function takes into account, as well, the classification of singletons (elements that are not grouped; have no links in the spanning forest). Hence, it will reward the partitioning when those singletons are clustered correctly.

5.4.4 Experimental results

We evaluate the accuracy performance of *perDgroups* against the ground truth provided by the *Student003* real dataset using the *G*-MITRE loss function. Based on the *G*-MITRE loss function, *precision* and *recall* values for each clustering method are computed to evaluate its accuracy performance. We run our proposed dominant groups method using the following parameters ($\tau = 1.5$, $w = 1$, $\kappa = 0.82$) obtained as described in Section 5.4.2.

Table 4 shows the results, where it can be seen that our proposed method outperforms the state-of-the-art presented in [31]. By focusing on individual's perspective, our method is able to identify groups that globally defined group patterns are ignoring as non-important.

6 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 [28]. A live online demonstration of this tool can be accessed online[3].

[3] https://sites.google.com/view/pedestrians-group-pattern/

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. 15. 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.

7 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*.

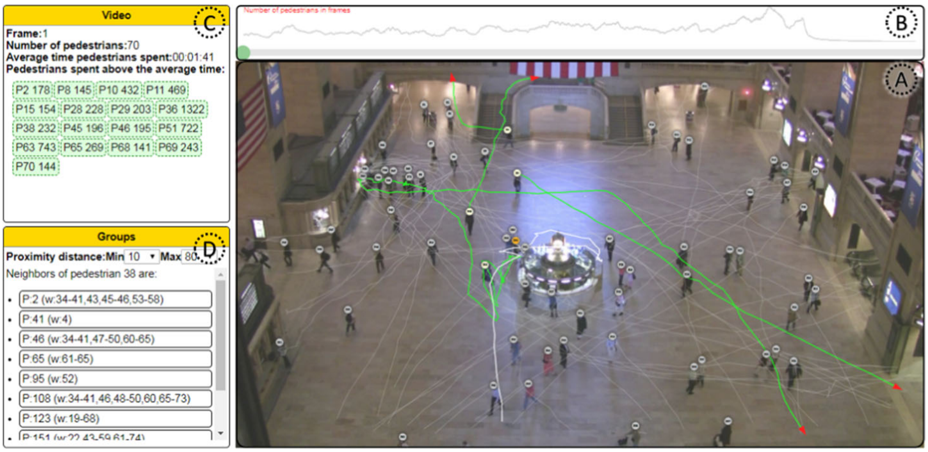


Fig. 15 Snapshot of the tool that allows for an interactive exploration of pedestrian group dynamics

Trajectory-based Group Mining The works more related to our research has already been reviewed in Section 1, so here we expand to other related work of this broad topic. Sakr et al. [26] proposed a generic query method for matching group spatiotemporal patterns by studying the group pattern matching problem from the perspective of database query. They also discussed the issues of integrating their method with moving object database (MOD) systems. Their method mainly utilizes history results of queries and merges them with incoming data. However, there is a lack of analysis and study of different query types related to moving objects that find similarities and common processings over time. Clustering the trajectories was utilized by Gaffney and Smyth et al. [12] using a mixture regression model. However, 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 [34] 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. [20] 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. [16] 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. [24] presented a ReTraTree indexing structure to maintain (sub-)trajectories' information over time. Lan et al. [18] proposed an algorithm to find *evolving groups* by finding the candidate clusters at each time point using DBSCAN [9]. 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 [2, 8, 14, 29]. 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 [33], anomaly detection [1], and escape behavior [35]). Solera et al. [31] presented an algorithm for detecting social groups of pedestrians in crowd by clustering trajectories using a novel affinity model based on sociological concepts and Structural Support Vector Machine (Structural SVM). They devised and investigated new features and their method achieved good results compared to state-of-the-art methods. However, obtaining trajectories in denser crowds is challenging and the problem becomes harder due to the fact that the clusters – used to identify each group – tend to present subtle differences. Bastani et al. [4] 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. [25] to extract a set of nodes, where each node represents a particular motion pattern. From another perspective, the study proposed by Zanlungo [39] 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 [6] 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.

8 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. In addition, we presented a novel pattern, *individual perspective pattern* that emphasizes on how individuals perceive groups. Based on this pattern, we introduced the concept of *dominant groups* and presented a systematic method for identifying the best candidate values for the hyperparameters, and for a given ground truth dataset. Our analysis reveals that these values will be restricted around the peaks of the proximity curves (as depicted by the black dashed ellipse of Fig. 13a). By limiting the search space, we can effectively find the most prominent values for setting the selectivity parameter and guide the group pattern analysis.

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

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

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 online[4].

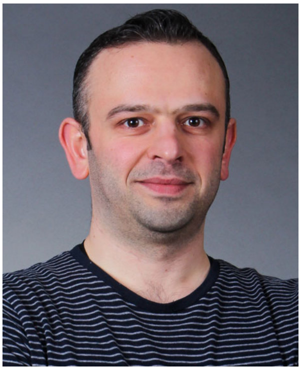
References

1. Abuolaim A, Leow WK, Varadarajan J, Ahuja N (2017) On the essence of unsupervised detection of anomalous motion in surveillance videos. In: International Conference on Computer Analysis of Images and Patterns, pp 160–171
2. Andriluka M, Roth S, Schiele B (2008) People-tracking-by-detection and people-detection-by-tracking. In: Computer vision and Pattern Recognition (CVPR), pp 1–8

[4] https://sites.google.com/view/pedestrians-group-pattern/

3. Bae SH, Yoon KJ (2014) Robust online multiobject tracking with data association and track management. IEEE Trans Image Process 23(7):2820–2833
4. Bastani V, Campo D, Marcenaro L, Regazzoni C (2015) Online pedestrian group walking event detection using spectral analysis of motion similarity graph. In: International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp 1–5
5. Benkert M, Gudmundsson J, Hübner F, Wolle T (2008) Reporting flock patterns. Comput Geom 41(3):111–125
6. Brščić D, Zanlungo F, Kanda T (2017) Modelling of pedestrian groups and application to group recognition. In: International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp 564–569
7. Cartwritht D, Zander A (1960) Group dynamics. Harper and Row, New York
8. Dollar P, Wojek C, Schiele B, Perona P (2012) Pedestrian detection: an evaluation of the state of the art. IEEE Trans Pattern Anal Mach Intell 34(4):743–761
9. Ester M, Kriegel HP, Sander J, Xu X et al (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDD, vol 96, pp 226–231
10. Fan Q, Zhang D, Wu H, Tan KL (2016) A general and parallel platform for mining co-movement patterns over large-scale trajectories. Proceed VLDB Endowment 10(4):313–324
11. Faugeras O (1993) Three-dimensional computer vision: a geometric viewpoint. MIT Press, Cambridge
12. Gaffney S, Smyth P (1999) Trajectory clustering with mixtures of regression models. In: Knowledge Discovery and Data Mining (SIGKDD), pp 63–72
13. Guo D, Liu S, Jin H (2010) A graph-based approach to vehicle trajectory analysis. J Locat Based Serv 4(3-4):183–199
14. Hwang S, Park J, Kim N, Choi Y, So Kweon I (2015) Multispectral pedestrian detection: Benchmark dataset and baseline. In: Computer Vision and Pattern Recognition (CVPR), pp 1037–1045
15. Jeung H, Shen HT, Zhou X (2008) Convoy queries in spatio-temporal databases. In: International Conference on Data Engineering (ICDE), pp 1457–1459
16. Jeung H, Yiu ML, Zhou X, Jensen CS, Shen HT (2008) Discovery of convoys in trajectory databases. Proc VLDB Endowment 1(1):1068–1080
17. Kalnis P, Mamoulis N, Bakiras S (2005) On discovering moving clusters in spatio-temporal data. In: Spatial and Temporal Databases (SSTD), pp 364–381
18. Lan R, Yu Y, Cao L, Song P, Wang Y (2017) Discovering evolving moving object groups from massive-scale trajectory streams. In: Mobile Data Management (MDM), pp 256–265
19. Le Bon G (1921) The Crowd: A study of the popular mind. Macmillan New York, New York
20. Lee JG, Han J, Whang KY (2007) Trajectory clustering: a partition-and-group framework. In: Proceedings of the international conference on Management of data (SIGMOD). ACM, pp 593–604
21. Lerner A, Chrysanthou Y, Lischinski D (2007) Crowds by example. In: Computer Graphics Forum, vol 26, pp 655–664
22. Moussaïd M, Perozo N, Garnier S, Helbing D, Theraulaz G (2010) The walking behaviour of pedestrian social groups and its impact on crowd dynamics. PloS one 5(4):e10047
23. Nanopoulos A, Theodoridis Y, Manolopoulos Y (2001) C2p: clustering based on closest pairs. In: VLDB, pp 331–340
24. Pelekis N, Tampakis P, Vodas M, Doulkeridis C, Theodoridis Y (2017) On temporal-constrained sub-trajectory cluster analysis. Data Min Knowl Disc 31:1294–1330
25. Rupasinghe R, Senanayake S, Padmasiri D, Ekanayake M, Godaliyadda G, Wijayakulasooriya J (2016) Modes of clustering for motion pattern analysis in video surveillance. In: Information and Automation for Sustainability (ICIAfs), pp 1–6
26. Sakr MA, Güting RH (2014) Group spatiotemporal pattern queries. GeoInformatica 18(4):699–746
27. Sawas A, Abuolaim A, Afifi M, Papagelis M (2018) Tensor methods for group pattern discovery of pedestrian trajectories. In: 2018 19Th IEEE International Conference on Mobile Data Management (MDM), pp 76–85. IEEE
28. Sawas A, Abuolaim A, Afifi M, Papagelis M (2018) Trajectolizer: Interactive analysis and exploration of trajectory group dynamics. In: Mobile Data Management (MDM), pp 286–287
29. Shu G, Dehghan A, Oreifej O, Hand E, Shah M (2012) Part-based multiple-person tracking with partial occlusion handling. In: Computer Vision and Pattern Recognition (CVPR), pp 1815–1821
30. Siła-Nowicka K, Vandrol J, Oshan T, Long JA, Demšar U, Fotheringham AS (2016) Analysis of human mobility patterns from gps trajectories and contextual information. Int J Geogr Inf Sci 30(5):881–906
31. Solera F, Calderara S, Cucchiara R (2016) Socially constrained structural learning for groups detection in crowd. IEEE Trans Pattern Anal Mach Intell 38(5):995–1008
32. Vilain M, Burger J, Aberdeen J, Connolly D, Hirschman L (1995) A model-theoretic coreference scoring scheme. In: Proceedings of the 6th Conference on Message Understanding. Association for Computational Linguistics, pp 45–52

33. Volkhardt M, Schneemann F, Gross HM (2013) Fallen person detection for mobile robots using 3d depth data. In: International Conference on Systems, Man, and Cybernetics (SMC), pp 3573–3578
34. Wakuda Y, Asano S, Koshizuka N, Sakamura K (2012) 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, pp 590–597
35. Wu S, Wong HS, Yu Z (2014) A bayesian model for crowd escape behavior detection. IEEE Trans Circ Syst Video Technol 24(1):85–98
36. Yi S, Li H, Wang X (2015) Understanding pedestrian behaviors from stationary crowd groups. In: Computer Vision and Pattern Recognition (CVPR), pp 3488–3496
37. Yuan G, Sun P, Zhao J, Li D, Wang C (2017) A review of moving object trajectory clustering algorithms. Artif Intell Rev 47(1):123–144
38. Zanlungo F, Ikeda T, Kanda T (2014) Potential for the dynamics of pedestrians in a socially interacting group. Phys Rev E 89(1):012811
39. Zanlungo F, Kanda T (2013) Do walking pedestrians stabily interact inside a large group? analysis of group and sub-group spatial structure. In: Cogsci
40. Zhang T, Soh SH, Fu X, Lee KK, Wong L, Ma S, Xiao G, Kwoh CK (2009) Hpcgen a fast generator of contact networks of large urban cities for epidemiological studies. In: Computational Intelligence, Modelling and Simulation (CSSim), pp 198–203
41. Zheng K, Zheng Y, Yuan NJ, Shang S (2013) On discovery of gathering patterns from trajectories. In: International Conference on Data Engineering (ICDE), pp 242–253



Abdullah Sawas is a Ph.D. candidate in Electrical Engineering and Computer Science (EECS) at York University, Toronto, Canada. His research interests include big data, IoT, cyber security, machine learning, and smart grids. He holds a M.Sc. in Computing and Information Science from Masdar Institute, Abu Dhabi, United Arab Emirates, and a B.Sc. in Informatics Engineering from the Aleppo University, Aleppo, Syria. Before joining York University, he was a project engineer in the department of Electrical Engineering and Power System at Saudi Aramco, Dhahran, KSA. He received the best paper award at the 19th IEEE International Conference on Mobile Data Management (IEEE MDM 2018).

Abdullah Abuolaim is a Ph.D. candidate in Electrical Engineering and Computer Science (EECS) at York University, Toronto, Canada. He received his Bachelor of Science in Computer Engineering (BSCE) from Jordan University of Science and Technology (JUST), June, 2015. Then he obtained his Masters of Science in Computer Science (M.Sc.) from the National University of Singapore (NUS), Singapore, July, 2017. His research interests include Computer Vision, image processing, computational photography, deep learning, and trajectory mining.

Mahmoud Afifi received his B.S. and M.S. degrees in information technology from the Faculty of Computers and Information, Assiut University, Egypt, in 2009 and 2015, respectively. He was with Assiut University from 2011 to 2016. He served as the vice chair of the ACM Assiut Student Chapter, Egypt. He is currently with the Department of Electrical Engineering and Computer Science, Lassonde School of Engineering, York University, Canada. Since 2017, he is an editorial board member of the Electronics Science Technology and Application journal. He is reviewer of several international journals and conferences, such as IEEE Transactions on Intelligent Transportation Systems, Journal of Electronic Imaging, British Machine Vision Conference, and IEEE Winter Conf. on Applications of Computer Vision. His research interests include image processing, computer vision, and computational photography.

Manos Papagelis is an Assistant Professor of Electrical Engineering and Computer Science (EECS) at York University, Toronto, Canada. His research interests include data mining, graph mining, machine learning, big data analytics, knowledge discovery and city science. He holds a Ph.D. in Computer Science from the University of Toronto, Canada, and a M.Sc. and a B.Sc. in Computer Science from the University of Crete, Greece. Prior to joining York University, he was a postdoctoral research scholar at the University of California, Berkeley. In the past, he has worked twice as a research intern at Yahoo! Labs, Barcelona and as a research fellow at the Institute of Computer Science, FORTH, Greece. His research has appeared in the ACM Trans. on Knowledge Discovery from Data (ACM TKDD) and the IEEE Trans. on Knowledge and Data Engineering (IEEE TKDE). He received the best paper award at the 19th IEEE International Conference on Mobile Data Management (IEEE MDM 2018), and the outstanding reviewer award at the 26th ACM International Conference on Information and Knowledge Management (ACM CIKM 2017). He has taught at York University, the University of California, Berkeley and the University of Toronto.