Tool support for detection and analysis of following and leadership behavior of pedestrians from mobile sensing data

Mikkel Baun Kjærgaard, Henrik Blunck*

Department of Computer Science, Aarhus University, Denmark

ARTICLE INFO

Article history:

Available online 6 November 2013

Keywords:

Pattern recognition
Crowd behavior sensing
Mobile sensing
Signal-strength-based methods

ABSTRACT

The vast availability of mobile phones with built-in movement and location sensors enables the collection of detailed information about human movement even indoors. As mobility is a key element of many processes and activities, an interesting class of information to extract is movement patterns that quantify how humans move, interact and group. In this paper we propose methods for detecting two common pedestrian movement patterns, namely individual following relations and group leadership. The proposed methods for identifying following patterns employ machine learning on features derived using similarity analysis on time-lagged sequences of WiFi measurements containing either raw signal strength values or derived locations. To detect leadership we combine the individual following relations into directed graphs and detect leadership within groups by graph link analysis. Methods for detecting these movement patterns open up new possibilities in – amongst others – computational social science, reality mining, marketing research and location-based gaming. We provide evaluation results that show error rates down to 7%, improving over state-of-the-art methods with up to eleven percentage points for following patterns and up to twenty percentage points for leadership patterns. Furthermore, we provide an analysis of the computational efficiency of the proposed methods and present visualizations for the analysis of detected patterns. Our methods are, contrary to state of the art, also applicable in challenging indoor environments, e.g., multi-story buildings. This implies that even quite small samples allow us to detect information such as how events and campaigns in multi-story shopping malls may trigger following in small groups, or which group members typically take the lead when triggered by e.g. commercials, or how rescue or police forces act during training exercises.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The vast availability of mobile phones with built-in movement and location sensors enables large-scale collection of sensor data about human movement behavior, as compared to what was previously possible with dedicated devices. However, such data only becomes valuable through methods that can process and aggregate the data to extract relevant information [1]. As mobility is a key element of many human processes and activities, an interesting class of information to extract is movement patterns that quantify how humans move, interact and group [2].

In this work we target the detection of two common pedestrian movement patterns: *following* relations and group *leadership* [3,2]. A *following* pattern involves two persons and describes a relation between them, where one person is following and the other is leading. The *following* pattern occurs when a leading person's movement continuously prescribes the movement of a following person. A group *leadership* pattern is a quasi global pattern for a group of persons: it is present

* Corresponding author. Tel.: +45 22422965; fax: +45 22422965.

E-mail addresses: mikkelbk@cs.au.dk (M.B. Kjærgaard), blunck@cs.au.dk (H. Blunck).

when a leader person's movement prescribes the movement of other persons but no other persons' movement prescribes the movement of the leader person. Both patterns are temporally and spatially constrained in so far as the persons have to be no further apart than δ m at the same point in time.

The detection of *following* and *leadership* patterns and the proposed visualization means enable new analysis methods within areas such as reality mining [1], computational social science [4], emergency research and marketing research [5], and provide new primitives for pervasive computing and location-based games. First, for computational social sciences and reality mining to extract patterns among moving co-workers including staff at hospitals, caretakers in large buildings and workers in warehouses. The detected patterns provide quantitative statistics about the relationship among co-workers. Furthermore, compared to mere proximity information the *following* and *leadership* patterns can reveal asymmetries among co-worker relationships [6]. For instance, one can deduce from such patterns whether people work independently or in teams and who leads the work. This information can then e.g. be compared with the intended working structure and help identify problems that when solved might provide e.g. gains in efficiency [7]. The clues extracted from movement and location data might also be correlated with clues extracted from audio or video data [8] or communication patterns among co-workers [1].

Second, there is a great and growing interest within marketing research to use movement data to better understand people's shopping behavior. However, methods for properly analyzing the data are lacking [5]. An important and so far less explored aspect is the impact of social interactions [5]. One example is the detection of co-shopping behaviors [9] for which the methods proposed in this paper would enable new forms of analysis, correlating *following* and *leadership* patterns with e.g. the shopping behavior of family members, product placements and store layouts.

Finally, the detection of *following* and *leadership* patterns also provides new primitives for pervasive computing and location-based games. We will later present an evaluation of our method in the context of a multi-person location-based evader game building on these primitives.

The previous work on the detection of *following* and *leadership* patterns in a spatio-temporal sense from movement and location data is mainly limited to the work of Andersson et al. [3], which focuses on computational efficient detection of such patterns from planar location data. They evaluate their geometric algorithm on synthetic data mimicking outdoor animal movements. However, as most humans spend a lot of their day indoors it is for many use-cases crucial to provide methods that are also applicable in multi-story buildings.

The previous work on visual scene understanding has addressed some elements of the problem mainly focusing on *following* patterns [10]. However, visual approaches are hampered by their need for images with a good coverage of the scene which is particularly hard to satisfy in indoor areas. Furthermore, as the identity of the subjects is often unknown, vision-based methods are not applicable in several of the above scenarios. To capture non-spatio-temporal notions of *leadership* both audio and visual methods have been applied, e.g., by analyzing audio and visual clues during a conversation [8].

Focusing on indoor environments, there have been several works focusing on further movement patterns utilizing various sensor modalities, for example, *co-moving*, proximity, flocking, user behavior modeling, crowd density and group structure detection, but not *following* or *leadership* patterns. Chandrasekaran et al. [11] proposed methods for detecting *co-moving* devices from correlation features of WiFi signal strength measurements and later applied such methods for speed estimation [12]. Several works have utilized Bluetooth [6,1,13] for proximity detection. Eagle et al. [1] proposed methods for modeling users' behavior from such data and Do et al. [4] proposed methods for utilizing it for building probabilistic models of the latent group structures. Efstratiou et al. [13] used Bluetooth data to detect social interactions in a group and Adams et al. [6] used Bluetooth and GPS data to understand proximity and then model user behavior as rhythms of place visits and social interactions. Phung et al. [14] extended this work to also consider WiFi readings in an indoor setting to quantify user rhythms. Weppner et al. [15] used Bluetooth to estimate the density of a crowd. Krumm et al. [16] used WiFi measurements for detecting proximity for their NearMe system. WiFi measurements have also been utilized for flock detection [17,18]. However, no work has so far targeted the detection of *following* and *leadership* patterns. Furthermore, technologies, e.g., Bluetooth, that can only detect proximity among moving subjects are not applicable for the problem as they cannot detect who is following who—only that the two targets are in proximity.

In this paper, we propose methods identifying *following* patterns using machine learning algorithms on features derived by using similarity metrics on time-lagged sequences of mobile sensing measurements, e.g., raw WiFi signal strength and acceleration measurements. To detect *leadership* we first combine the detected pairwise *following* relations into directed graphs and then subsequently perform a graph link analysis. We provide evaluation results showing low error rates and a clear improvement over state of the art for datasets from both a scripted and an unscripted experiment, conducted in two different multi-story buildings. In this paper we extend our earlier work [19] by including more in-depth evaluation of both the parameterization of the methods and analyzing the runtime efficiency of the methods. Furthermore, we propose and analyze both temporally, person-focused and spatially centered visualizations of the detected patterns—with the aim to provide support for the analysis of detected follower and leadership patterns.

The proposed methods support a fine-grained real-time analysis, that can detect *following* and *leadership* relations, and evolution thereof, in indoor environments of various spatial layouts. Our results imply that, for instance, when a crowd coming from a transit area diffuses into an open space, and smaller groups are heading in different directions, then our methods will be superior in identifying and analyzing these groups w.r.t. *following* and *leadership* patterns. Thus, they allow us to e.g. measure the effect of a commercial or an announcement in the transit area. Also, they allow for quick identification of leading characters in groups, when arriving at an open space—e.g. they allow us to conclude that the teenage kids who

Fig. 1. Activity classification states for detecting *following* patterns.

Fig. 2. Overview of the method for *following* and *leadership* pattern detection.

saw a toy commercial started leading the family to the toy store. Another example is the training or evaluation of rescue teams' and police forces' actions in large multi-story public buildings, offices or industrial plants; here, our methods offer information about whether forces stuck to prescribed *leadership* patterns, or which events triggered new *following* and *leadership* patterns, or whether there was panic behavior, characterized by weak or highly volatile *following* relationships. With such an accurate, fine-grained, and prompt analysis, e.g. without cumbersome video analysis, the learning benefits from such training or evaluation will be much more substantial and immediate.

2. Overview of the proposed method

In the following, we present the methods we propose for detecting *following* and *leadership* patterns among pedestrians, and relying on mobile sensing data, primarily on WiFi signal strength measurements, from the pedestrians' mobile devices.

For detecting *following* and *leadership* patterns, we classify people's movement behavior according to a set of states. For two persons P1 and P2 potentially following each other, the different states in the set are shown in a classification hierarchy in Fig. 1: for *following* detection, the first step of the hierarchy is to classify each person as *moving* or *stationary*. If the two persons in question are both moving, they are further classified with regard to if they are *co-moving* or *moving independently*. If the persons are *co-moving*, it is then considered which of the following three situation applies: is P1 following P2, or P2 instead following P1, or are they *co-leading* each other. The *co-leading* state describes the situation where persons are walking side by side. We argue that the term *co-leading* more naturally characterizes this situation than *co-following*. Note also that the states *stationary* and *moving independently* could be further subclassified into activities such as meeting, not in proximity or passing, but this is beyond the scope of this work. For detecting the *leadership* pattern, we reuse the *following* pattern classifications by combining the pairwise *following* relations into directed graphs that are the basis for classifications of *leadership*, as will be detailed in Section 4.2. Fig. 2 shows the steps of the proposed method for detecting *following* and *leadership* patterns using mobile sensing measurements from peoples' mobile devices. We assume that a sensing service will be deployed on people's mobile devices through existing distribution channels, e.g., as part of a mobile application for a scientific study or a location-based game, or as a system service (A). When requested, the n mobile devices will each measure the acceleration of the device and the signal strength of nearby WiFi access points. The measurements are processed and a standard threshold-based classification on acceleration measurements is used to detect if a device is moving or stationary (B). The services on the devices $1, \dots, n$ will send the signal strength measurements to a remote service when detected to be moving as measurement vectors $m_1, \dots, m_n$. The remote service might be implemented as a web service running on a cloud platform (C). For each feature, the signal strength measurements are processed for pairs of devices to calculate time-lagged similarity features, based on raw signal strength measurements or derived location measurements using location fingerprinting [20] (D). For each feature space the feature values are gathered in feature vectors $v_1, \dots, v_k$ where k is the number of device pairs, where both devices are moving (E).

Fig. 3. Feature vector $v_{a,b}$ computed for devices a and b over a time window w within the minimum and maximum lag z .

The feature vectors are given to a classification algorithm that considers for each device pair if they are *co-moving* or not (F). For the *co-moving* devices their feature values are gathered in vectors $v'_1, \dots, v'_l$ where l is the number of *co-moving* device pairs (G). Each vector is processed to classify it with respect to *following* relationships. These classifications can both be output as final results or be used for *leadership* detection (H). For *leadership* detection the pairwise *following* relations are combined into directed graphs (I). *Leadership* is detected using graph link analysis (J). The final output consists of devices classified with respect to the *leadership* pattern (K). In the following we will discuss the steps in more detail.

3. Time-lagged similarity features

The proposed method for movement pattern detection requires an algorithm for binary classifying, for each pair of devices, whether they are *co-moving*, as sketched in the last section; see step (F) in Fig. 2. In this section we introduce time-lagged similarity features and propose these as the basis for such a classification algorithm. Common to these features is that they

- enable analysis of (snapshots of) signal measurements or derived location measurements to detect movement patterns, even from non-spatial data.
- enable *following* detection via the intuition that if a device b is following another device a , then observations of signal strength measurements or location coordinates for b are expected to appear “delayed” with a time lag, when comparing them to the measurements from a .
- achieve the capturing of similarity of multi-dimensional signal measurements or derived location measurements for various time shifts via (one of two variants of) the following general scheme:
 1. compute similarity, either (i) for each data dimension, using classic time series metrics (such as cross correlation) or pattern recognition algorithms (Dynamic Time Warping), or (ii) for each time step, using distance metrics (such as Euclidean distance) or similarity coefficients for sets (such as the Tanimoto Coefficient).
 2. combine similarity values computed for (i) individual dimensions or (ii) time steps by computing average values.

We argue to use time-lagged similarity features instead of purely spatial but non-temporal features, since the latter are not able to capture temporal aspects of people’s movements: for instance, Andersson et al. [3] have proposed to define the *following* pattern, as occurring, whenever a (leading) device is observed in a two-dimensional wedge (of a given width and radius, e.g. in a half-circle) in front of another (i.e., a following) device. However, especially in restricted spaces such as indoor environments this characterization of the *following* pattern breaks down in many situations, such as when making frequent turns either when walking down a staircase, when walking through a crowded room, or when walking in front of one another but on different floors. In comparison, similarity features computed over a time window can operate on the similarity of movement and can thus capture *following* patterns also in the above situations. Furthermore, using such temporal similarity features allows us to utilize also other than purely spatial data for *following* detection, e.g. observed signal strengths.

Formally, for a device a we define m_{a_t} as the measurements for time step t . The methods presented here assume a measurement to be either a set of signal strength measurements for observable access points or a location coordinate triplet, derived from a set of signal strength measurements using location fingerprinting [20].

Time-lagged features are computed for a pair of devices a, b as a vector $v_{a,b}$, where each entry is associated with a certain time lag $i \in [-z, \dots, z]$ in seconds, where z defines the range of time lags we decided to check for. Hence, the length of $v_{a,b}$ is $2z + 1$, and for each time lag i , $v_{a,b}$ then holds a feature value, which indicates the similarity of the measurements from a and b , when shifting those of b by a time lag i .¹ Fig. 3 illustrates the concept, and that each feature value is computed over a time window of size w over m_a and m_b , and Fig. 4 depicts an example feature vector.

¹ The temporal granularity within $v_{a,b}$ is fixed to 1 s solely for the sake of ease of exposition.

Fig. 4. Feature vector $v_{a,b}$, time lag indices and mapping to *following* or *leading* classifications.

3.1. Similarity metrics

To unify the handling of the similarity features, when using them within our pattern detection method as sketched above, we define them all as distance metrics—so that a value of zero denotes perfect similarity and higher values denote higher dissimilarity. Some of the features have been applied by prior work for the detection of proximity, place or speed estimation. To evaluate them, with the aim to identify the similarity metric with the best performance for our pattern detection problem, we consider: (i) two distance metrics: Euclidean distance and Walking distance for location data, (ii) two pattern recognition algorithms: Dynamic Time Warping and Derivative Dynamic Time Warping, (iii) one time series metric: cross correlation coefficient and (iv) one statistical set coefficient: Tanimoto coefficient.

For a device pair a, b we compute the features for each time lag entry v_{a,b_i} for $i \in [-z, \dots, z]$ with respect to the current time t over a feature window of size w . We define two locations or signal vectors $a = m_{a_t}$ and $b = m_{b_{t+i}}$. The dimensionality d of these vectors stems from the spatial dimensions for location data and the number of access points observable by both devices for signal data, respectively.²

Euclidean Distance (ED) is a basic distance metric which has been extensively applied for, e.g., proximity detection [16, 14]. Given a function ed that computes the Euclidean distance per time step of two vectors, we compute the similarity of the two vectors a and b as follows:

$$v_{a,b_i} = \frac{1}{w} \sum_{0 \leq j < w} ed(a_j, b_j). \quad (1)$$

Walking Distance (WD) is a more refined distance metric for indoor spaces, modeling that pedestrians cannot move through walls and floor separations [18]. Given a function wd that computes the walking distance between two coordinates per time step we compute the similarity of the two vectors a and b using Eq. (1) with wd in place of ed .

Tanimoto Coefficient (TC) is a statistical coefficient for sets, which has previously been studied for proximity detection for signal strength measurements [21]. Given a function that computes the Tanimoto coefficient tc we compute the similarity of the two vectors a and b using Eq. (1) with $1 - tc$ in place of ed .

Dynamic Time Warping (DTW) is a classic dynamic programming algorithm which has been widely used for optimal alignment of two time series, but has so far not been used for movement pattern detection from signal strength or location measurements. As part of the alignment process it computes a similarity metric as a cost function equal to the minimum-distance to an optimal alignment. The algorithm can potentially compute better similarity metrics than some of the other features as it can realign data, e.g., to mask inconsistent sampling of measurements, variations in walking speed of targets or outlier measurements. This characteristic promises thus low sensitivity to (i) measurement noise that often exists in complex indoor environments, and (b) to smaller variations in distance and motion, as they frequently occur between two persons in a natural following relationship.

Given a function dtw that computes the cost of the minimum-distance of the optimal alignment using DTW, we compute the similarity of the two vectors a and b as follows:

$$v_{a,b_i} = \frac{1}{n} \sum_{0 \leq k < d} dtw(\dim_k(a, b)). \quad (2)$$

Derivative Dynamic Time Warping (DDTW) is an extension of DTW that computes an optimal alignment of the derivatives of two time series. Existing work has considered using DDTW to estimate speed from GSM readings [12] by analyzing the warp path produced as an output of the DDTW algorithm. Here, we apply DDTW for a different task and instead of analyzing the warp path we use the cost which is the minimum-distance to an optimal alignment. The previous work that applied DDTW argued for the superiority of DDTW because it does not compare absolute values but instead compares the rate of change of the measurements. However, e.g. in indoor environments signal strength values can change rapidly and chaotically

² The signal strength measurements are preprocessed, prior to the feature calculation, by excluding measurements for weak and unstable access points and by smoothing measurements – to reduce the effect of outliers – with a running average over five seconds of measurements.

Fig. 5. Example graph of detected *following* relations.

over a few meters due to obstructions by building elements, thus constituting a potential error source for DDTW. Given a function ddtw that computes this cost we compute the similarity of the two vectors a and b using Eq. (2) with ddtw in place of dtw .

Cross Correlation (CC) is a common technique for detecting *co-moving* used in the previous work, both for signal strength measurements [11] and location measurements [22]. Given a function cc that computes the cross correlation we compute the similarity of the two vectors a and b using Eq. (2) with $1 - \text{cc}$ in place of dtw .

4. Detecting following and leadership

4.1. Following detection

As a first step of following detection we filter out non-*co-moving* device pairs: to detect *co-moving* devices, we employ a standard machine learning classifier, based on one of the presented time-lagged similarity features. The classifier is trained with labeled data of *co-moving* devices and of devices moving individually. The data labeled individually moving includes devices far away from each other, vertically displaced and passing each other. The resulting classifier outputs for a given pair a, b of devices its classification into either *co-moving* or moving independently. We choose a support vector machine classifier, since these showed the best performance on our datasets compared to, e.g., Naive Bayes or tree-based classifiers.

For *co-moving* device pairs we utilize again the time-lagged similarity features to detect *following* relations. We analyze the feature vectors $v_{a,b}$, see Fig. 4, to find the lag i_{est} , for which the highest similarity is achieved (for the measurements from the two devices), and which thus represents their most likely alignment in time. If the lag is negative, a device is classified as *following*, if positive as *leading*, and if close to zero ($+/-0.1$) as *co-leading*. To extract the lag i_{est} from the vector, we calculate the inverse weighted average using the similarity values as weights, where s denotes the normalization factor:

$$i_{\text{est}} = \frac{1}{s} \sum_{-z \leq i \leq z} \frac{1}{v_{a,b}(i)} \cdot i, \quad \text{with } s = \sum_{-z \leq i \leq z} 1/(v_{a,b}(i)).$$

In the example in Fig. 4, the lag is estimated to -0.23 and therefore device b is classified as *following* a . Note that we also evaluated to extract i_{est} as the minimal value from $v_{a,b}$, but the weighted solution provides a 1% point higher detection accuracy in our experiments.

4.2. Leadership detection

As we illustrated using Fig. 2, we base both the definition as well as the detection of the *leadership* pattern on analyzing the pairwise *following* relations among persons. These relations, interpreted as directed edges, constitute a graph; an example is given in Fig. 5. Thus, *leadership* can then be detected via analyzing the said graph. An apt definition of *leadership* is given by Andersson et al. [3], who define a leader for a given timestamp among a set of tracked persons as a person who (i) has a certain minimal amount of followers, but (ii) is not following any tracked person. The above definition though does not take into account some phenomena, which we aim to address in our *leadership* detection and which are discussed below.

First, we propose to take into account the transitivity of *following* relations: if a person a is following a person b who is following a person c , we say that a is *transitively following* c . We furthermore propose to score potential leaders by counting the transitive followers for each potential leader c so that in the example c receives a higher score than (a) , but to do so in a weighted fashion, where the weight of a transitive follower diminishes w.r.t. to a potential follower c diminishes with the link distance to it. The resulting scoring scheme bears similarity to node scoring graph algorithms such as PageRank [23].

Second, our *leadership* detection also takes into account that the graph which it operates on may contain noise, i.e. the set of detected *following* relations may deviate from the actual relations – due to sensor noise or inaccurate pattern extraction. An example is given in Fig. 5, where it has been erroneously detected, that d is following e – which leads to a cycle of transitive *following* relations in the example graph. While no leader exists in the said graph component, according to Andersson's definition, we argue that a robust *leadership* detection should not categorically rule out identifying leaders in such cases, as depicted here with d . Note also that such detection has no certainty about which of the edges of the cycle are erroneous, and thus cannot just eliminate cycles by removing an arbitrary edge of each cycle.

Furthermore a robust detection should also allow us to relax Andersson's definition and not rule out to identify more than one leader (e.g. g and h in Fig. 5) within a (seemingly) connected component of the graph in cases where a potentially

Table 1
Scenes of the experiment.

Name	Description
<i>Scripted - Horiz. A</i>	Subjects moving on a single floor (4:08 min).
<i>Scripted - Vertical A</i>	Subjects moving on both floors (4:27 min).
<i>Scripted - Vertical B</i>	Subjects moving on both floors (10:25 min).
<i>Scripted - Vertical C</i>	Subjects moving on both floors (7:23 min).
<i>Unscripted - Horiz. A</i>	Subjects moving on a single floor (8:40 min).
<i>Unscripted - Horiz. B</i>	Subjects moving on a single floor (6:56 min).

Fig. 6. Building layout for two buildings with icons for staircases.

erroneous edge connects two otherwise unconnected components of the graph (as does edge (i,j) in Fig. 5). Note that incorporating transitive *following* relations as described above may also help addressing poor *following* detection accuracy, specifically in cases where some direct *following* relations were not detected.

Lastly, the noise-resilience of *leadership* detection can be increased if in the input graph each edge is assigned a weight, which resembles the confidence with which the associated *following* relation was detected. Such weights can then be taken into consideration by graph node scoring algorithms, such as by PageRank. In the context of *leadership* detection, it allows us to pay less attention or even disregard edges, which are more likely to be erroneous.

5. Evaluation

We evaluated for the proposed *following* and *leadership* detection methods their detection accuracy, considering the accuracy of the individual features and comparing with state-of-the-art algorithms.

5.1. Experiments and datasets

We utilize two datasets in our evaluation. The first was collected during a scripted experiment using data from fourteen participants and the second during an unscripted experiment using a location-based game as an evaluation setting with ten participants. Table 1 lists the properties of the individual scenes of the two experiments. As shown in Fig. 6, the scripted experiment was conducted in an office building and the unscripted experiment in a lecturing building with large open spaces. Both buildings had a deployment of WiFi access points and we observed on average ten and eleven access points per WiFi measurement for the scripted and the unscripted experiments, respectively.

In both experiments we utilized Android phones of various models: Google Nexus, HTC Desire and Samsung Galaxy S+. The phones ran dedicated sensor logging software to record WiFi signal strength and acceleration at the platform's default sampling rate. For the WiFi positioning we collected fingerprints throughout the buildings in the scripted experiment in a 1.5 m grid on two floors and in the unscripted experiment in a 3.0 m grid on a single floor.

The scripted experiment was conducted according to a predefined script and monitored with video cameras and manual annotation in order to obtain a ground truth against which the method's performance can be evaluated. During each scene, *following* relations and the subjects showing *leadership* behavior are changing several times. For obtaining ground truth we analyzed the video to label the participants' behavior accordingly. In order to obtain recordings of natural behavior, we did not instruct the subjects on how to move or behave and only instructed them to keep their mobile phone in alternatively their left or right front trouser pocket and stay close together in their designated group (where "close" was not further specified but was observed to be around 1.5 m).

The unscripted experiment was conducted using a location-based game as an evaluation setting. The game was designed as a "follower-evader" game with each subject taking the role as either a follower or an evader. Followers would score points by following an evader in close distance ("close" being specified to participants as up to 7 m) and evaders would score points when nobody followed them. Using the method presented in this paper we would be able to implement this game without manual observation or intervention on the participant side. However, to obtain ground truth labels while collecting the dataset, we designed a smartphone game application in which the participants labeled the data as part of the game.

Fig. 7. Game experiment including the game application interface for followers and evaders.

Fig. 8. Error rates for *co-moving* detection for the scripted experiment.

Fig. 7 shows both a picture from the experiment with evaders in orange jerseys and the game application interface. The scoring scheme was designed so that the followers had to occasionally change evaders—enticed by bonus points and negative points for staying with the same evader for longer than 45 s. This information was communicated to the participants by the game application via vibration and a color scheme. These elements were also designed to keep the players attending to the app and update the information continuously to score points. However, to minimize timing flaws in the participants' labeling, we chose to remove labels for the first seconds before/after a change and to not compute patterns for these points in time.

5.2. Results

To evaluate the proposed detection method, we compare detection results to the collected ground truth information. We compute the detection error rate of the detection individually for the two datasets, and also individually on both location and signal data for the considered feature types: Euclidean Distance (ED), Dynamic Time Warping (DTW), Derivative Dynamic Time Warping (DDTW) and Cross Correlation (CC). We evaluate the methods using several window sizes w but the main results are given for $w = 10$ s. To reflect that we defined for the experiments the spatial constraint of the patterns δ as seven meters, we set the absolute time lag maximum z to seven seconds, which corresponds to the maximal temporal distance between persons in a valid following–leading relationship, when assuming a walking speed of around 1 m/s. To demonstrate that our method generalizes in the sense that it works in various environments with different building elements and access point densities, we always trained the *co-moving* classifiers using data from an experiment different from the one we test the method in: e.g., for the given test results for the scripted experiment, we trained with data solely from the unscripted experiment. We compare our method to two state-of-the-art detection methods. First, the method proposed by Chandrasekaran et al. [11] for *co-moving* detection based on cross correlation, and second, the geometric method by Andersson et al. [3] for both *co-moving*, *following* and *leadership* pattern detection.

Following detection. As *co-moving* detection is a necessary pre-step for the following detection, we also present results for this step in Figs. 8 and 9 for the scripted and the unscripted experiments, respectively. As can be seen for both the scripted and unscripted experiments, overall the DTW feature outperforms the other features with error rates down to 7%. Applied to our datasets, it provides an error reduction of up to five percentage points when compared to [11] and up to eleven percentage points when compared to [3]. We attribute the superiority in performance, consistent over both experiments, to the DTW method's mentioned low sensitivity to data fluctuations, which occur even when the two entities are in a solid and natural *co-moving* relation, namely typical measurement noise as well as small fluctuations in relative motion and distance between the entities. Only for the scripted experiment exists a contender with similar performance, namely the WD feature.

The results for *following* pattern detection are given in Figs. 10 and 11 for the scripted and unscripted experiments, respectively. Once more, the DTW-based feature provides the best performance, i.e. an error reduction of up to 11% points

Fig. 9. Error rates for co-moving detection for the unscripted experiment.

Fig. 10. Following detection accuracy for the scripted experiment.

Fig. 11. Following detection accuracy for the unscripted experiment.

when compared to [3], i.e. a relative error rate reduction of 45% and 37% for the scripted and unscripted experiments, respectively.

Selection of window size. Fig. 12 shows results for different window sizes w . These results reveal that for the unscripted experiment a value of $w = 10$ s gave the best results whereas for the scripted experiments a value of $w = 15$ s provided an improvement of a few percentage points. We reason that this difference can be attributed to the difference of average time between changes of *following* relations. Therefore, for scenarios which show less changes in the *following* relations, as in the scripted experiment, a larger window size is beneficial, whereas for scenarios with more frequent changes, as in the unscripted experiment, a smaller window size is suitable. Overall, the window size is an application-specific choice and resembles a trade-off between accuracy and detection latency. Additionally, a larger window size has an impact on the runtime performance as discussed in later sections; for this reason, it is beneficial to keep it as small as possible. For automatically determining a suitable window size we thus suggest a dynamic calculation based on the average time between changes.

Leadership detection. Fig. 13 compares the error rate as exhibited in the two experiments, respectively, for different *leadership* detection methods, namely for Andersson et al.'s method (labeled 'ALGW') [3], an instance of the PageRank algorithm [23], and our method (labeled 'LDP') using the techniques described in Section 4.2.³ In blue is shown as a baseline Andersson et al.'s method for *leadership* detection, showing error rates of about 46% for the scripted and 24% for the gaming experiment, respectively. Provided as input to the detection algorithm were the *following* relations, as detected

³ The results shown for the PageRank algorithm are obtained using the implementation within Networkx [24] and parameter choices, resulting in superior detection accuracy, i.e., a maximum of 200 iterations and $\alpha = 0.85$.

Fig. 12. Follower detection accuracy for different window sizes.

Fig. 13. Leadership detection accuracy for all experiments.

by Andersson et al.'s method. If instead the output from our proposed *following* detection methods, specifically from the DTW-based method, is used as input, accuracy is improved: for the scripted experiment by 11% points when running the DTW-based *following* detection on location data, and yet by 20% points when running on raw signal data instead. For the gaming experiment, the improvements are 1% and 7% points, respectively. Furthermore, when now using PageRank instead of Andersson et al.'s method for graph link analysis, we see a slight improvement only in the unscripted experiment, but an increased error rate in the scripted experiment. Finally, our proposed *leadership* detection method generally improves over the two other graph analysis methods, but never by more than 2% points. Thus, a general conclusion from the depicted evaluation results for *leadership* detection is that the accuracy of the latter is highly impacted by the accuracy of the utilized *following* pattern detection method—whereas the difference between graph link analysis methods, when applied to the same set of *following* relations, is comparatively small.

5.3. Computational efficiency

In this section, we analyze the computational effort for the proposed means of computing *following* and *leadership* patterns. We first determine the computational effort for detecting all *following* relations within a group of observed targets. Let the primitive for testing for a *following* relation between two targets a and b at a time t be denoted by $p(a, b, t)$. The overall effort corresponds to the number of primitive calls $p(a, b, t)$ for testing two targets a and b for a *following* relation between them at a given time instance t . Since this primitive is called per timestamp for which the targets were observed and for which *following* relations are to be determined, the computational effort is linear in the number of timestamps of interest. For each timestamp, the primitive is called for each pair of input targets. Thus, the overall computational effort is quadratic in the number n of targets, since it is linear in the number of target pairs. The overall effort is also naturally linear in the length of the observation period for which *following* relations are to be computed. More specifically, the effort is linear in the length $\mathcal{T}$ of the sequence of observation times for which *following* relations are computed. The overall complexity for testing for follower relations among n targets at a given timestamp t is thus in $\mathcal{O}(\mathcal{T} \cdot n^2 \cdot \text{cost}(p))$.

We now analyze the computational cost of a single primitive call $p(a, b, t)$ for testing targets a and b for a *following* relation at time t . The effort of a feature computation is quadratic in the dimension d of the input data. The dimensionality d of the input data is here (for spatial input) the number of spatial dimensions or (for signal strength input) the number of access points, respectively. The effort is furthermore dependent on the chosen window size w around the timestamp t . This dependency on w is linear for most of the investigated features—including the one which showed the highest detection accuracy in our evaluation, the DTW method [25]. Similarly, the number of similarity feature values to be computed for a primitive call $p(a, b, t)$ is linear in the maximal time lag z for which the primitive tests. This maximal time lag z specifies the range of time lags for which feature values in $p(a, b, t)$ are considered. The mentioned primitive call $p(a, b, t)$ for the methods proposed here, as described in Section 4.1, involves an initial check for a co-moving relation and, if successful, a subsequent check for a *following* relation. For both checks a time-lagged feature as described in Section 3 is computed. For

Fig. 14. Evaluation effort for various numbers of tracked targets in seconds, normalized per analyzed timestamp of data.

Fig. 15. Evaluation effort in dependence of the configured maximal time lag, normalized per analyzed timestamp of data.

the preceding co-moving check, only one (i.e. not time-lagged) feature for a and b is computed; thus the co-moving check serves as a heuristic to avoid the $\mathcal{O}(z)$ feature computations in the *following* test. Overall, the effort for each primitive call $p(a, b, t)$ using the DTW method is thus of complexity $\mathcal{O}(d^2 \cdot w \cdot z)$. The total effort for computing all *following* relations for n targets for T observation timestamps is in $\mathcal{O}(\mathcal{T} \cdot n^2 \cdot d^2 \cdot w \cdot z)$.

We now consider the computational complexity for computing all *leadership* patterns among n targets. The three methods considered for this in this paper, namely *FollowingFlow*, *PageRank* and Andersson et al.'s approach, all compute these patterns by analyzing the time-variant graph of following relations. Naturally, for all these approaches, the actual computational effort needed to compute *leadership* patterns is dependent on the number of edges of the graph, i.e., on the number of detected *following* relations. Generally though, the worst-case complexity for computing leadership patterns (w.r.t. the parameters considered above) is dominated by the computation of the underlying *following* relations, and is thus also in $\mathcal{O}(\mathcal{T} \cdot n^2 \cdot \text{cost}(p))$.

Complementing the above complexity analysis, we also measured empirical run times of the algorithms while varying several input parameters. We used (subsets of) the data from the gaming experiment as input data. The running times were obtained on a MacBook with a 2.6 GHz Intel Core i7 processor and 8 GB RAM.

Fig. 14 presents the running times for various numbers n of tracked targets, normalized per analyzed timestamp of the input data. The results in Fig. 14 are furthermore averaged over all target subsets of a depicted size n , respectively. For the method for detecting *following* relations we used the DTW time-lagged feature method. The four lines in Fig. 14 show the results using either signal strength input data or location input data, and either with or without the heuristics of pre-checking for co-moving before checking for following relations. The figure clearly shows that the co-moving heuristic can reduce run times significantly, i.e., by 31%, averaged over all shown data points.

Similarly, Fig. 15 shows the running time for a given maximal time lag z . Each line corresponds to a choice of window size w , with either signal-strength-based input data or location input data. The figure reveals the expected quasi-linear dependencies on window size and maximal time lag. Furthermore, both figures show the large difference in the run time between signal-strength-based and location-based data, which is due to the quadratic dependency on the dimension d of the input data. Thus, the results suggest for future work to consider a reduction in dimensionality of the signal strength data: in order to reduce run time, one can consider only a subset of access points as input dimensions. However, if such a subset is chosen poorly, the resulting lower running time may significantly lower the detection accuracy. Therefore, the challenge for future algorithm improvement is here to choose subsets which offer good trade-offs between running time and detection. Note that the priorities for the two parameters to be traded off here may differ in different application scenarios.

6. Discussion

In the last section, we have presented and in part analyzed the results of our evaluation, which demonstrated the good performance of the proposed method. In this section, we will discuss the differences which show in the results across the datasets and for using raw signal strength data compared to using derived locations.

Results from scripted vs. unscripted experiments

In regards to lowest error rates for *co-moving* detection, we achieve the best results in the scripted experiment, whereas the lowest error rates in the unscripted experiment are five percentage points higher. Three of the main reasons for this difference are: first, the unscripted experiments were conducted in a more open spaced building rendering signal strength measurements less descriptive over similar distances. This led to more *co-moving* detection errors in the unscripted experiment. Second, the density of participants was slightly higher in the unscripted experiment. The effect of this can be observed by comparing the two unscripted scenes to a third unscripted scene where participants were allowed to utilize not one, but two floors.⁴ We achieved *co-moving* error rates of 7% for this scene—considerably lower than for the scenes restricted to one floor, since the wider separation between participants enabled better *co-moving* detection. Third, at some points in the game experiment several evaders were walking so close to each other that some of their followers factually co-moved with not one (as labeled in the ground truth) but with multiple evaders.

Naturally, for both most of the *following*, as well as for the *leadership* detection methods, their performance is impacted by the accuracy with which the more basic patterns are detected, i.e. *co-moving* for *following* detection, and *following* for *leadership* detection. For the *following* detection method, though, which performs best in both experiments, we obtain slightly better accuracy improvements—both for *following* and for *leadership* detection compared to the *co-moving* performance—in the unscripted experiment. The main reason for this being that the participants in the scripted experiment were on average walking closer to each other than in the unscripted experiment, thereby making it potentially harder for the algorithms to identify who is following who among a pair or group of persons, who are detected as *co-moving*.

Using signal vs. location data

For the more accurate among the evaluated *following* detection methods, specifically for the superior DTW-based one, error levels for signal data input are slightly lower than for location data input.⁵ For *following* detection these gains are visibly higher for the unscripted experiment, which was conducted in a more open spaced building part and for which the collection of location fingerprints was carried out in a less dense grid. For the detection of *leadership* patterns in both experiments, see Fig. 13, the accuracy gains when using signal instead of location data are even larger than for *following* detection. We conclude from these results that (i) for *following* detection it becomes necessary to provide a well calibrated location system in order to match the accuracy that can be achieved from raw signal data (without any area-specific calibration), and (ii) for *leadership* detection the latter data type may still provide superior accuracy. However, note that positioning capabilities prove useful as they allow us to associate the detected movement patterns with the spatial location where they occurred—which may be relevant in various use-case scenarios.

7. Visualization

The method proposed in the paper enables the detection of following and leadership patterns. A meaningful way for users to utilize the detection results is to visualize the detected patterns. In this section we consider how to visualize the patterns both temporally, person-focused, and spatially either in aggregated form or animated. For illustration purposes, we focus on one of the datasets from the unscripted game experiment, as we then can relate the visualizations back to the game. Furthermore, the perspective here is not an additional evaluation of the methods but considers what information can be read out of the visualizations. Additionally, the visualization also allows us, even when not given ground truth, to assess the accuracy of the detection methods.

Focusing on the temporal development of *following* or *leadership* patterns involving individual persons allows us to assess, e.g., the behavior of a person in relation to others, and which individuals did the person follow or lead. An example of a related visualization is given in Fig. 16. The figure shows for each game participant a row spanning the game in time and the person-specific colored legend on the left denotes their role as either evader or follower. In a person's row are indicated periods in which the person was followed, respectively lead by other person (identified by their color). Furthermore indicated are periods of detected *leadership*, by a black line. The shown data is from the unscripted game experiment horizontal B and the detection shown is based on DTW features computed from signal strength data.

We illustrate the use of this visualization example by illustrating what can be analyzed deduced from it in regards to the game mechanics shown in the experiment. Regarding the dynamics of the game, initially only a few of the followers started following an evader; then a big group formed which followed evader 4. Subsequently, over a period the evaders were largely able to avoid followers, e.g., because the big group abandoned evader 4 and did not find other evaders to follow instantly. As the evaders and followers in the remainder of the game were more dispersed, no such big group formed again.

⁴ We did not include the data for this scene in the main results due to lack of location fingerprints for the whole area.

⁵ This difference in accuracy is even more visible for CCORR and DDTW, but their classification performance generally profits from the higher dimensionality of input data (here: number of observable access points vs. at most three spatial dimensions).

Fig. 16. Visualization focusing temporally on persons, showing for each game participant a row spanning the game in time, and *following* via colored blocks and *leadership* via black lines.

Fig. 17. Visualization focusing on the spatial extent of patterns. The visualization shows for three time intervals continuous series of patterns as uniquely colored connected dots.

Second, we consider the performance and strategy of the players. From the figure one can clearly see the difference between players reaching the goals of either evading or following. For instance, follower 2 only achieved in shorter periods to follow an evader, whereas follower 4 managed to follow an evader most of the time.

Focusing on the spatial aspects of the patterns allows us to assess, e.g., where a respective pattern started and stopped and over how long a distance the pattern persisted. An example of a visualization focusing on the spatial extent is given in Fig. 17. The figure shows for each pair, constituted by a follower and an evader, the spatial extent of each continuous series of *following* events involving this pair. Each *following* event is shown as a dot, located half-way between the involved leading and following target. Furthermore, each continuous series is drawn with a new color and with arrows connecting consecutive *following* events in the series. The shown data is from the unscripted game experiment horizontal B and the detection shown is based on DTW features computed from signal strength data drawn at the computed positions based on location fingerprinting. As the spatial extent of the patterns overlap, this type of visualization is best viewed either for a shorter time interval or as an animation. Therefore, Fig. 17 shows three different time intervals of 2 min and furthermore, an animated version is available online using diminishing opacity levels to enable pattern series to fade into the background as time elapses.

Regarding game mechanics, we give examples of what can be analyzed from the figure. First, the spatial extent of the game play can be analyzed. For instance, we can observe that over time the play extends to more and more of the area and the start and stop positions of the series get more and more distributed (as visible even more clearly in animation). Second, the strategy of the evaders develops: whereas in the beginning they walk straight routes, they take more turns in the later intervals to make their walk less predictable and therefore more difficult to follow.

The visualization tools have been realized building on two frameworks. First, for the temporal views the tools build on previously developed modules in Python for the visualization of crowd related data [26]. These modules are designed to visualize various types of temporal data. Second, for the spatial views our tools build on a framework for animating, representing and visualizing indoor spatial information [27]. The framework was developed in Java and allows input of various data types spanning from purpose specific, e.g., *following* relation data to standardized, e.g., GPX files for GPS data. Both visualization types shown here, in Figs. 16 and 17, were built on top of these frameworks by adding needed adjustments and extensions to support the visualization of following and leadership patterns.

8. Conclusions and future work

We presented methods for accurately detecting *following* and *leadership* movement patterns. The methods can operate on various kinds of mobile sensing data, e.g. on bare bones' radio signal strength or on derived location data. We showed the superior accuracy of the method by comparing against several approaches in two experimental settings in different indoor environments. Based on our evaluation, we also discussed factors impacting detection accuracy to be expected in different scenarios. The promising results suggest that the proposed method should be further tested in the outlined application domains in order to enable new applications based on real-time analysis of *following* and *leadership* behavior of pedestrians. Furthermore, to provide a wider applicability of the proposed methods, we present support for visually analyzing the obtained pattern data through aggregating and through animated visualizations. However, there remain challenges for future research, e.g. in interpreting the semantics of detected movement patterns in relation to the build environment and social and cultural differences. Such differences are hard to capture for the proposed methods alone; therefore, one could enrich the method with the detection of spatial characteristics or clues from other modalities such as audio and video when available, in order to achieve a more context-tailored and more differentiated, and thereby more accurate, pattern detection.

Acknowledgments

We would like to thank Markus Wüstenberg, Kaj Grønbaek, Martin Wirz, Daniel Roggen and Gerhard Tröster for their contributions to the conference version of this paper. This work was supported by the Danish Council for Strategic Research for the project EcoSense, the FP7 ICT Future Enabling Technologies Programme under grant agreement No 231288 (SOCIONICAL), and by a grant from the Carlsberg Foundation.

References

- [1] N. Eagle, A. Pentland, Reality mining: sensing complex social systems, *Personal and Ubiquitous Computing* 10 (4) (2006) 255–268.
- [2] S. Dodge, R. Weibel, A.-K. Lautenschütz, Towards a taxonomy of movement patterns, *Information Visualization* 7 (3–4) (2008) 240–252.
- [3] M. Andersson, J. Gudmundsson, P. Laube, T. Wolle, Reporting leaders and followers among trajectories of moving point objects, *Geoinformatica* 12 (4) (2008) 497–528.
- [4] T.M.T. Do, D. Gatica-Perez, Contextual grouping: discovering real-life interaction types from longitudinal bluetooth data, in: *MDM*, 2011, pp. 256–265.
- [5] S.K. Hui, P.S. Fader, E.T. Bradlow, Path data in marketing: an integrative framework and prospectus for model building, *Marketing Science* 28 (2) (2009) 320–335.
- [6] B. Adams, D.Q. Phung, S. Venkatesh, Sensing and using social context, *TOMCCAP* 5 (2) (2008).
- [7] L. Wu, B.N. Weber, S. Aral, E. Brynjolfsson, A. Pentland, Mining face-to-face interaction networks using sociometric badges: predicting productivity in an it configuration task, in: *ICIS*, 2008.
- [8] D. Sanchez-Cortes, O. Aran, M.S. Mast, D. Gatica-Perez, A nonverbal behavior approach to identify emergent leaders in small groups, *IEEE Transactions on Multimedia* 14 (3–2) (2012) 816–832.
- [9] S. Grossbart, L. Carlson, A. Walsh, Consumer socialization and frequency of shopping with children, *Journal of the Academy of Marketing Science* 19 (1991) 155–163.
- [10] N. Oliver, B. Rosario, A. Pentland, A bayesian computer vision system for modeling human interactions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (8) (2000) 831–843.
- [11] G. Chandrasekaran, M.A. Ergin, M. Gruteser, R.P. Martin, J. Yang, Y. Chen, Decode: exploiting shadow fading to detect comoving wireless devices, *IEEE Transactions on Mobile Computing* 8 (12) (2009) 1663–1675.
- [12] G. Chandrasekaran, T. Vu, A. Varshavsky, M. Gruteser, R.P. Martin, J. Yang, Y. Chen, Tracking vehicular speed variations by warping mobile phone signal strengths, in: *PerCom*, 2011, pp. 213–221.
- [13] C. Efstratiou, I. Leontiadis, M. Picone, K.K. Rachuri, C. Mascolo, J. Crowcroft, Sense and sensibility in a pervasive world, in: *Pervasive*, 2012, pp. 406–424.
- [14] D.Q. Phung, B. Adams, K. Tran, S. Venkatesh, M. Kumar, High accuracy context recovery using clustering mechanisms, in: *PerCom*, 2009, pp. 1–9.
- [15] J. Weppner, P. Lukowicz, Bluetooth based collaborative crowd density estimation with mobile phones, in: *Percom*, IEEE, 2013.
- [16] J. Krumm, K. Hinckley, The nearme wireless proximity server, in: *UbiComp*, 2004, pp. 283–300.
- [17] M.B. Kjærgaard, M. Wirz, D. Roggen, G. Tröster, Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones, in: *UbiComp'12*, ACM, 2012, pp. 240–249.
- [18] M.B. Kjærgaard, M. Wirz, D. Roggen, G. Tröster, Mobile Sensing of Pedestrian Flocks in Indoor Environments using WiFi Signals, in: *PerCom*, 2012.
- [19] M.B. Kjærgaard, H. Blunck, M. Wüstenberg, K. Grønbaek, M. Wirz, D. Roggen, G. Tröster, Time-lag method for detecting following and leadership behavior of pedestrians from mobile sensing data, in: *PerCom*, 2013.
- [20] M.B. Kjærgaard, C.V. Munk, Hyperbolic location fingerprinting: a calibration-free solution for handling differences in signal strength, in: *PerCom*, 2008.
- [21] D.H. Kim, Y. Kim, D. Estrin, M.B. Srivastava, Sensloc: sensing everyday places and paths using less energy, in: *SenSys*, 2010, pp. 43–56.
- [22] T. Shirabe, Correlation analysis of discrete motions, in: *Geographic Information Science*, Springer, 2006, pp. 370–382.
- [23] L. Page, S. Brin, R. Motwani, T. Winograd, The pagerank citation ranking: bringing order to the web, Tech. rep. 1999.
- [24] A.A. Hagberg, D.A. Schult, P.J. Swart, Exploring network structure, dynamics, and function using NetworkX, 2008.
- [25] S. Salvador, P. Chan, Toward accurate dynamic time warping in linear time and space, *Intelligent Data Analysis* 11 (5) (2007) 561–580.
- [26] M. Wirz, P. Schläpfer, M.B. Kjærgaard, D. Roggen, S. Feese, G. Tröster, Towards an online detection of pedestrian flocks in urban canyons by smoothed spatio-temporal clustering of gps trajectories, in: *GIS-LBSN*, 2011, pp. 17–24.
- [27] M.B. Kjærgaard, An api for integrating spatial context models with spatial reasoning algorithms, in: *PerCom Workshops*, 2006, pp. 18–23.