An algorithm to identify avoidance behavior in moving object trajectories

**Luis Otavio Alvares · Alisson Moscato Loy ·
Chiara Renso · Vania Bogorny**

Abstract Research on trajectory behavior has increased significantly in the last few years. The focus has been on the search for patterns considering the movement of the moving object in space and time, essentially looking for similar geometric properties and dense regions. This paper proposes an algorithm to detect a new kind of behavior pattern that identifies when a moving object is avoiding specific spatial regions, such as security cameras. This behavior pattern is called avoidance. The algorithm was evaluated with real trajectory data and achieved very good results.

Keywords Trajectory behavior · Spatiotemporal pattern · Moving objects · Trajectory data mining · Avoidance behavior

1 Introduction

Usage of location aware devices such as GPS and mobile phones has significantly increased in the last few years.

L.O. Alvares (✉) · V. Bogorny
INE/UFSC, Florianopolis, Brazil
e-mail: alvares@inf.ufsc.br

V. Bogorny
e-mail: vania@inf.ufsc.br

L.O. Alvares · A.M. Loy
II/UFRGS, Porto Alegre, Brazil

A.M. Loy
e-mail: amloy@inf.ufrgs.br

C. Renso
KDD Lab, Pisa, Italy
e-mail: chiara.renso@isti.cnr.it

These kinds of devices can generate sequences of space-time points capturing the trajectories of the object that carries the device. This kind of data—trajectory data—acquired for operational level use, is being generated at an incredible rate, and can be analyzed to obtain new knowledge; a higher level knowledge for decision making processes.

There are several real-world situations that consider spatiotemporal phenomena that are a target of analysis and research, as the pattern of humans buying items in a supermarket or a shopping center, animal migration behavior monitoring, human behavior in parks and cities, vehicle traffic, boat movement, etc. The study of trajectory behavior of these moving objects intends to transform these enormous quantity of raw data in useful information to the decision making process, knowledge discovery, and reality interpretation. It can contribute to problem solving (for instance, identifying fishing areas [19]), to identify standards and tendencies, or to discover outliers, for instance. Trajectory data are obtained as a sequence of points (id, x, y, t), where (x, y) represent the geographic coordinates of the object id in the time instant t. We call this data as raw trajectories.

Many works have been developed over the last years considering the study of trajectory behavior. These works have been developed according to two major research perspectives: a geometric one [3, 6, 8, 9, 12] and a semantic one [2, 4, 5, 17, 18].

Some works analyze one trajectory at a time while others evaluate sets of trajectories using, for instance, clustering techniques. Several works search for some kind of similarity between trajectories: spatial format, time interval, velocity, stops at the same points, and so on. Those works discover different types of patterns, such as: flocks, convergence, leadership, encounter, co-location episodes, and so on.

However, as far as we know, there are no works that identify, in trajectory data, the behavior of moving objects that
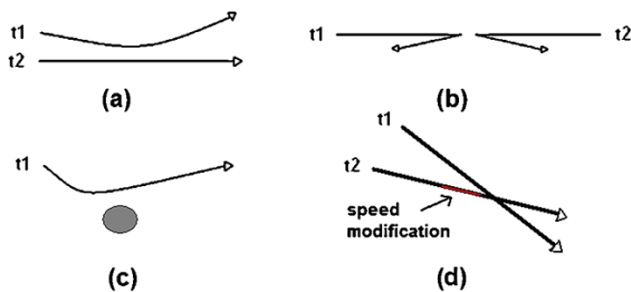
Fig. 1 Examples of the avoidance behavior

avoid some regions or that avoid other trajectories. For instance, people avoiding to collide with other people during a walk in a park, vehicles that change their route in situations of low speed traffic, or individuals that move in a suspicious manner avoiding vigilance cameras or security points.

An avoidance behavior can occur, for instance, when a trajectory avoids a specific spatial region, when one or more trajectories change their direction to avoid intersecting each other, or when one or more trajectories change their speed to avoid other moving objects, as can be seen in Fig. 1. In Figs. 1(a) and (b), the avoidance is between two trajectories and by direction changing. Figure 1(d) presents an example of avoidance between two trajectories by speed changing, while (c) presents another kind of avoidance where a moving object avoids a static region.

This work presents a new algorithm able to identify an avoidance behavior where a moving object avoids a specific spatial region, as in the example shown in Fig. 1(c).

The discovery of avoidance patterns in moving objects may be useful in several application domains, such as the discovery of individuals with suspicious behavior, avoiding monitoring cameras, police stations, blitz, and so on. Avoidance pattern discovery can be interesting for traffic applications, with cars avoiding low traffic regions.

The remainder of this paper is organized as follows: Sect. 2 shows the main related works, Sect. 3 presents the heuristics used to identify an avoidance, Sect. 4 presents the developed algorithm to recognize an avoidance pattern, Sect. 5 shows some experiments, Sect. 6 presents a discussion on the parameters of the algorithm, and Sect. 7 concludes the paper.

2 Related work

Detecting patterns of movement has been of interest since 1970, when Hagenstrand posed the bases of Time-Geography [10], where he first proposed the idea of "spatiotemporal prism" to represent the human movement. From that time, a number of approaches tried either to represent the human movement or to detect patterns from datasets of movement traces. The paper from Dodge et al. [7] presents

a taxonomy of movement patterns. The interesting part of this work is that they first proposed a systematic vision of the movement patterns distinguishing between generic and behavioral patterns, and the generic pattern is divided into compound and primitive. For example, a moving cluster is classified as a primitive pattern whereas a flock is a behavioral pattern. Despite the fact that this proposal is interesting as a tentative to classify the many movement patterns proposed in the last decade, we believe that some important patterns are not included as, for example, the avoidance pattern.

Several recent works define trajectory patterns basically considering the geometric part of trajectories. Laube in 2005 introduced the mobile group pattern, which is a set of trajectories close to each other, with distance less than a given threshold, for a minimal amount of time (minTime) [13]. In this approach, the direction is not considered and frequent groups are computed with the algorithm Apriori [1]. Laube also [12] proposed five types of *geometric* trajectory patterns based on movement, direction, and location, convergence, encounter, flock, leadership, and recurrence. A flock pattern has at least m subtrajectories within a region of radius r that move in the same direction during a certain time interval. The leadership pattern must have at least m subtrajectories within a circular region of radius r that move in the same direction, and at least one of the entities is heading in that direction for at least a certain time. Encounter is the pattern characterized by at least m subtrajectories that are concurrently inside the same circular region of radius r, assuming they move with the same speed and direction. Reccurrence patterns occur when at least m entities visit a circular region at least k times.

In [6], collocation episodes in spatiotemporal data are computed, where groups of trajectories are spatially close in a time window and move together.

Another approach is the T-pattern [8]. It is a sequential trajectory pattern mining algorithm that first generates regions of interest considering dense regions in space, and then computes sequences of regions visited, taking into account transition time from one region to another and minimum support.

In trajectory data analysis, there are no works that define avoidance patterns. The few works about avoidance concern collision-avoidance. The idea is to develop real-time systems, called "collision-avoidance systems" that proactively detect the risk of collision between vehicles, and are intended to be used by pilots or automatically during their travels to avoid the collision with other vehicles. The focus has been on avoidance of different types of collision on cars [11], ships [15], and air traffic [20]. Also, in transportation systems, collision-avoidance is studied [16] for pedestrians.

Some works in robotics [14, 21, 22] use the idea of avoidance in trajectories, but instead of analyzing a trajectory to identify an avoidance, as proposed in our work, they use the

concept of collision-avoidance for planning the future trajectories of a robot.

On the contrary, our proposal aims at analyzing historical GPS traces (trajectories) in order to detect the presence of avoidance patterns. This method is not intended for real time collision avoidance, but to detect a specific avoidance behavior in past trajectories. To the best of our knowledge, there are no similar approaches in the literature.

## 3 Heuristics to identify an avoidance behavior

The avoidance behavior pattern occurs when a mobile object is moving toward an object of interest or target object (as a surveillance or security camera, for example), shifts to avoid passing the object of interest, and after that goes back to its original path. The challenge is to differentiate what is really a shift to avoid the object of interest from a natural path change caused by another reason.

Some aspects that have to be considered are the following.

The moving object should not cross (intersect) the object of interest called *the target object* (the region covered by the security camera, for instance), because if the moving object changes its direction but yet crosses the target object, it did not stay away from the target and, therefore, does not characterize an avoidance.

The moving object should keep going in the direction to the target object (the object to be avoided) and deviate from the target object relatively close to it to be considered an avoidance. A counterexample is a person walking and one or more kilometers away he/she deviates from a security camera; this person is probably changing his/her direction by any other reason and not for escaping the security camera, therefore, not characterizing an avoidance behavior. To materialize this idea, we created the concepts of *target object* and *region of interest*. *Target object* is a convex spatial location that a trajectory could avoid. The *region of interest* is defined by a distance d from the target object. Any behavior outside the region of interest is not considered because it is too far from the target object. Then a moving object must intercept the region of interest in order to characterize an avoidance.

Figure 2 shows these intuitions. Trajectory t_1 was moving in the direction to the target object, deviated to avoid it, and after a while continued more or less on its original path, characterizing a case of avoidance. Trajectory t_2 moves in the direction to the target object and intersects it, without avoiding it and, therefore, not characterizing a case of avoidance. Finally, trajectory t_3 was moving in the direction to the target object but changed its direction far away from the target, and the deviation occurred outside the region of interest and, therefore, not characterizing an avoidance behavior.
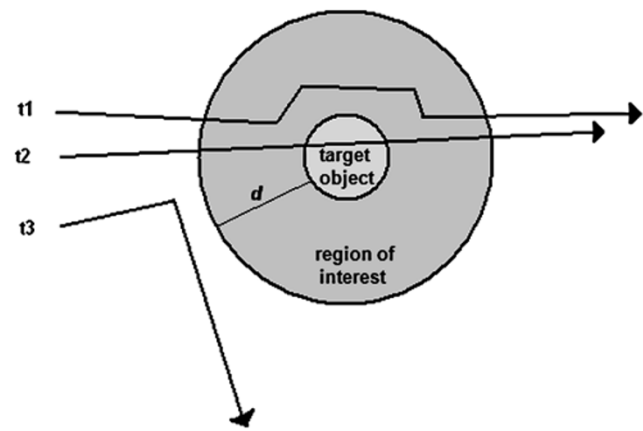
Fig. 2 Example of target object, region of interest, and trajectories behavior
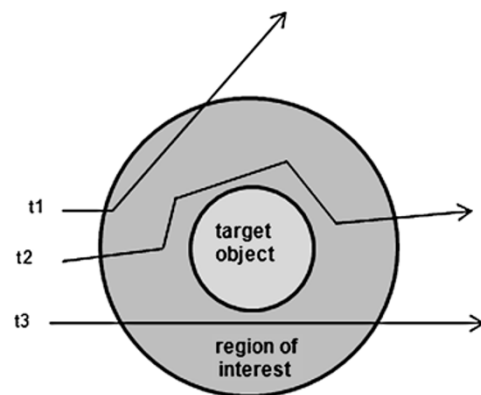
Fig. 3 Examples of trajectory behaviors

Even with the definition of region of interest, we observed that some trajectories, although intersecting the region of interest and not crossing the target object, did not present a clear behavior of moving toward the target object before changing direction. In these cases, it is not possible to give to these trajectories a suspicious behavior, considering the example of security cameras. Figure 3 shows some examples. The trajectory t_2 clearly presents an avoidance behavior, but for the trajectories t_1 and t_3 this is not so obvious.

In order to make more robust the identification of an avoidance behavior, we introduce the notion of subtrajectory directed to the target. The *subtrajectory directed to the target* is the longest subtrajectory that is moving in the direction to the target object, inside the region of interest, with length greater or equal to a minimum length l. Then a new condition to characterize an avoidance is: The trajectory should have a *subtrajectory directed to the target*. Figure 4 presents some examples of this concept. The trajectories t_1 and t_2 have a subtrajectory directed to the target but the trajectories t_3 and t_4 do not.

After these considerations, we can define the heuristic to characterize an avoidance: A *trajectory t* has an avoidance in

relation to a *target object o* if it has a *subtrajectory directed to the target o* and does not intersect the target object *o*.

However, if we observe the avoidance behavior of the trajectories in Fig. 5, intuitively we can say that the avoidance of trajectory t_1 is stronger than the avoidance of trajectory t_2, because t_1 returns to its original path after deviating the target object (the security camera), with a clear intent of avoiding the target. In the case of trajectory t_2, this is not so obvious, because the trajectory deviated the camera and follows this new direction without returning to its original path.

To know if a trajectory returns to its original path or not after deviating the target, we create the notion of *a confidence incremental region*, denoted by a region inside the region of interest that does not contain the target object and is situated between the target object and the edge of the region of interest, on the opposite side of the subtrajectory directed to the target, with width equal the diameter of the target object. An example is shown in Fig. 6. This region is unique for each trajectory considering a target object.

Intuitively, trajectory t_1, in Fig. 6(a) presents an avoidance with greater degree of certainty than trajectory t_2 in
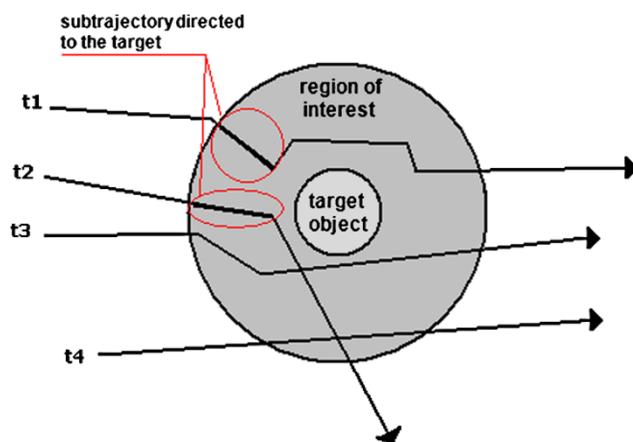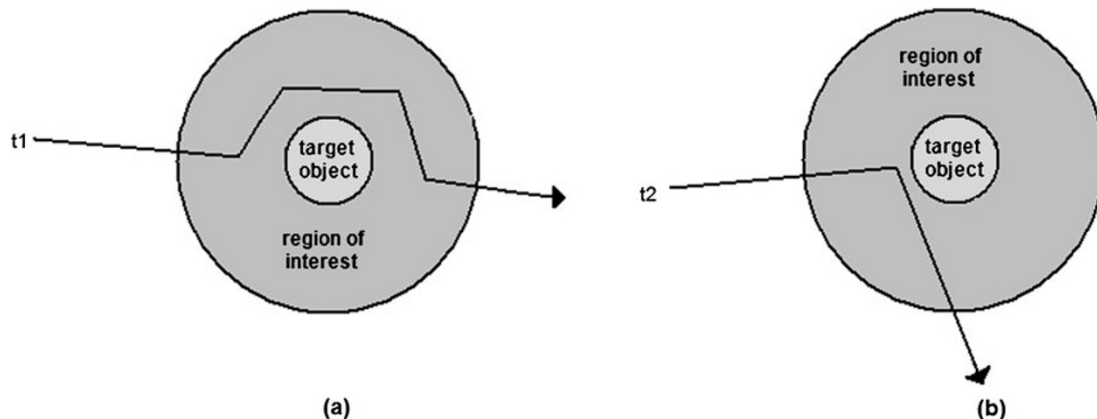
Fig. 6(b), because t_1 intersects the confidence incremental region. In this case, we are more sure that trajectory t_1 has an intentional avoidance and we give an weight for this situation. We create two levels of avoidance of one trajectory in relation to a target object: *strong avoidance*—when the trajectory intersects the confidence incremental region, and *weak avoidance*—when a trajectory moves in direction to a target, has a subtrajectory directed to the target, but neither intersects the target nor the confidence incremental region. This idea is mapped to a value, called *local avoidance confidence* (in relation to a specific target object): 0.0 for no avoidance, 0.5 for weak avoidance, and 1.0 for strong avoidance.

A last heuristic is that if there is a region with several security cameras we can analyze the behavior of the whole trajectory in relation to the whole set of target objects. To do this, if a trajectory crosses the region of interest of several target objects, we can have a *global avoidance value* for the whole trajectory, considering each *local avoidance value*. As a first approximation, we define the equation

$$Avt_i = \frac{\sum_{k=1}^{n} Av_{i_k}}{n} \qquad (1)$$

where Avt_i stands for the avoidance confidence for the whole trajectory i. Av_{i_k} is the value of local avoidance for trajectory i in relation to the target object k, and n is the number of regions of interest intersected by trajectory i.

Figure 7 shows an example of avoidance confidence for a whole trajectory. Trajectory t_1 has a strong avoidance considering the target object 1, a weak avoidance in relation to the target object 2, and has no avoidance for targets 3 and 4. Therefore, its global avoidance confidence is $(1 + 0.5 + 0)/3 = 0.5$. The target object 4 is not counted in the denominator because its region of interest is not intersected by the trajectory.


Fig. 4 Example of subtrajectory directed to the target

Fig. 5 Two different examples of avoidance
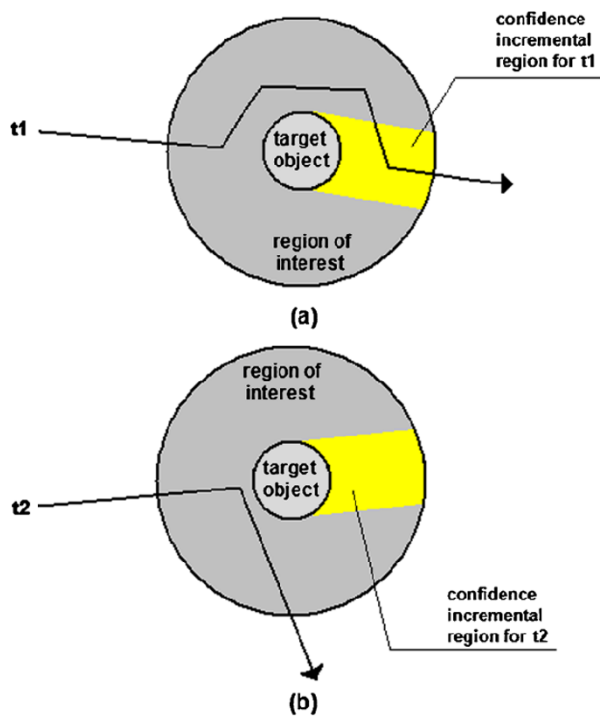
Fig. 6 Examples of confidence incremental regions

4 An algorithm for avoidance detection

Based on the heuristics presented in Sect. 3, we propose an algorithm to detect avoidance patterns, in the pseudo-code shown in Listing 1.

Initially, the algorithm tests the intersection of the trajectory points with the regions of interest of each target object (line 12). Only the trajectory points that intersect any region of interest are considered in the rest of the algorithm, which significantly decreases the processing time. If the trajectory intersects the region of interest, then it is not an avoidance (lines 14–15). The function SubtrajDT() (line 17) returns the longest subtrajectory that goes to the target inside the region of interest. The pseudocode is presented in Listing 2 and is detailed later in this section. The function ConfIncrR() (line 18 of Listing 1) determines the confidence incremental region, and is detailed later in Listing 3. If the trajectory intersects the increasing confidence region in a time period after the time period of the subtrajectory directed to the target, then the avoidance is strong (lines 19–20); otherwise, the avoidance is weak (lines 21–22). The global confidence of the avoidance of a whole trajectory i is computed in line 29, using (1).
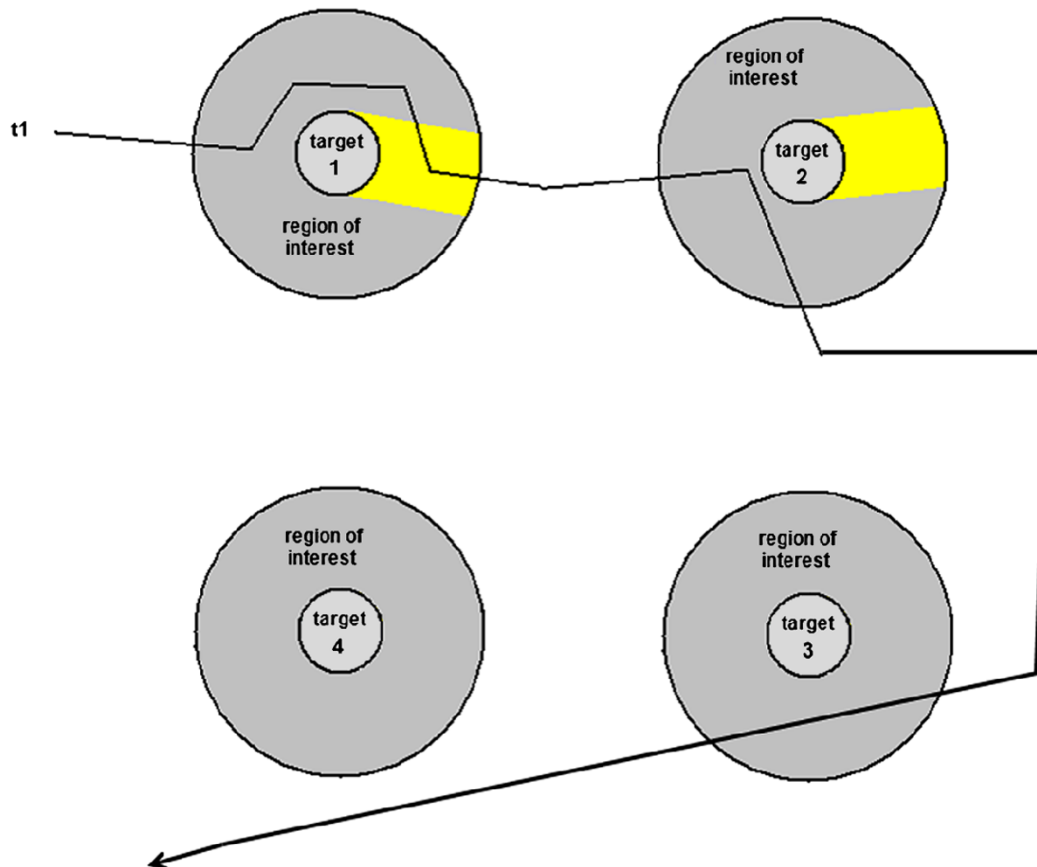
Fig. 7 Example of a trajectory in a region with four target objects

Listing 1 Pseudocode of the proposed avoidance algorithm

```
1   INPUT: T //set of trajectories
2          O //set of target objects
3          d //size of the buffer for the region of
4             //interest around the target object
5          subt //minimal size of the subtrajectory
6             //directed to the target
7
8   OUTPUT: Avt //set of degrees of avoidance
9
10
11  METHOD:
12  FOR each tᵢ∈T | intersects(tᵢ,buffer(O,d)) DO
13    FOR each oₖ∈O DO
14      IF intersects(tᵢ,oₖ)
15        avᵢₖ=none
16      ELSE
17        IF SubtrajDT(tᵢ,oₖ,d) >= subt
18          CIR=ConfIncrR(tᵢ,oₖ,d)
19          IF intersects(tᵢ,CIR)
20            avᵢₖ=strong
21          ELSE
22            avᵢₖ=weak
23          ENDIF
24        ELSE
25          avᵢₖ=none
26        ENDIF
27      ENDIF
28    ENDFOR
29    calculate Avtᵢ
30  ENDFOR
31  return Avt
```

Listing 2 Pseudocode of the SubtrajDT function

```
1   INPUT: P //set of trajectory points that
2          //intersect the region of interest
3          o //target object being analyzed
4          d //size of the buffer for the region of
5          //interest around the target object
6
7   OUTPUT: dist //greater euclidean distance in
8          //direction of the target object
9
10  METHOD:
11    i = P.firstPoint()
12    next = P.nextPoint()
13    dist,auxdist = 0
14    REPEAT
15      ap = azimuth(pᵢ,pₙₑₓₜ)
16      pₐᵤₓ.x = sen(ap)*2*d+(pᵢ.x
17      pₐᵤₓ.y = cos(ap)*2*d+(pᵢ.y
18      IF intersects(makeline(pᵢ,pₐᵤₓ),o)
19        auxdist = calcDistance(pᵢ,pₙₑₓₜ)
20        IF auxdist >dist
21          dist = auxdist
22        ENDIF
23        next = P.nextPoint()
24      ELSE
25        P.point = i
26        i = P.nextPoint()
27        next = P.nextPoint()
28      ENDIF
29    UNTIL the end of P
30    return dist
```

The function SubtrajectoryDT() shown in Listing 2, considers the first trajectory point inside the region of interest and takes the next points, one by one, while the direction of the line segment from the initial point to the last considered point intersects the target object. This procedure is repeated for the next points to determine the longest subtrajectory directed to the target inside the region of interest.

As shown in Listing 2, the procedure starts with the two first trajectory points that intersect the region of interest of the target object being considered (lines 11–12) and a loop is performed for all points in P (lines 14–29). Inside the loop, the first step is to calculate the azimuth between the two points to determine the direction of the trajectory and to extend this line segment until a possible intersection with the target object occurs (lines 15–17). If the line segment intersects the target object, then the mobile object is moving in the direction to the target (line 18). The next step is to calculate the Euclidean distance between the points and save the longest subtrajectory directed to the target (lines 20–21). While the line segment is moving in the direction to the target, the initial point is kept and the next point is taken. If the line segment is not moving in the direction to the target (line 24), the initial point becomes the next after that one that was the initial (lines 25–27). This procedure continues until all points of P have been evaluated.

Figure 8 exemplifies the calculus of the subtrajectory directed to the target. Figures 8(a), (b), and (c) show the same trajectory t_1 and the line segment calculated to test the intersection with the target at each cycle of the repeated loop. In Fig. 8(a), the line segment is created between the two first points inside the region of interest, p_2 and p_3. As it intersects the target object, the point p_2 is kept and the procedure continues with the points p_2 and p_4, as shown in Fig. 8(b), where the line segment intersects the target. In Fig. 8(c), the line segment is between the points p_2 and p_5, where the expanded line does not intersect the target. The procedure continues between the points p_3 and p_4 and after between p_3 and p_5, and so on. The subtrajectory directed to the target will be the distance between p_2 and p_4.

The function ConfIncrR() (line 18 of Listing 1) calculates the confidence incremental region. Its pseudo-code is presented in Listing 3. The function initially computes the azimuth between the first point of the trajectory inside the region of interest and the centroid of the target object. In the sequence, two line segments are computed, tangent to the target and with the calculated azimuth, from the target object to the exterior boundary of the region of interest (lines 13–14). Finally, the geometry of the confidence incremental region is computed (line 15).

Fig. 8 Example of subtrajetory directed to the target calculus

(a)

(b)

(c)
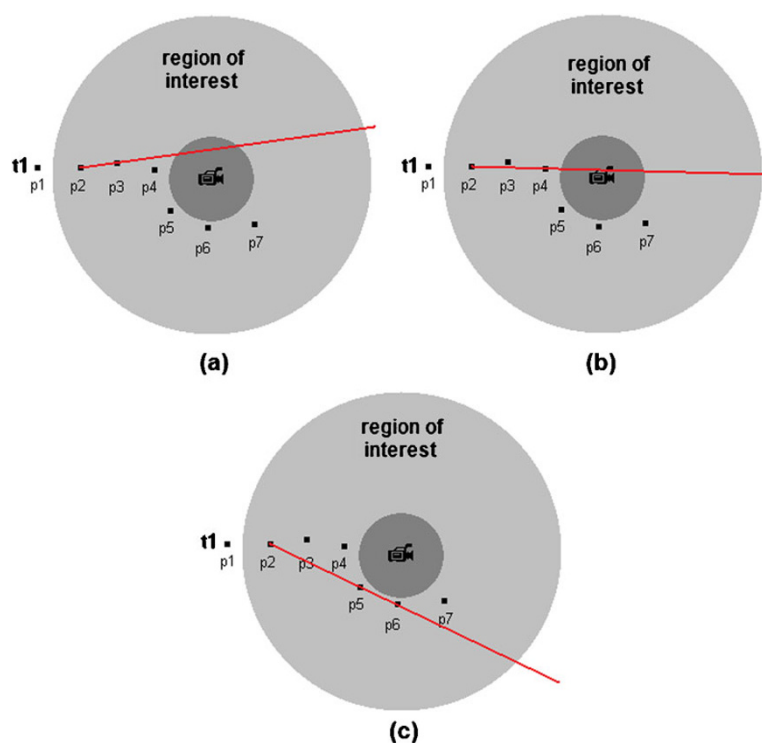
Although all examples of target objects are circular, this function works for any convex target object.

Listing 3 Pseudocode of the ConfIncrR function

```
1    INPUT: P //set of trajectory points that
2            //intersect the region of interest
3        o //target object being analyzed
4        d //size of the buffer for the region of
5            //interest around the target object
6
7    OUTPUT: reg // confidence incremental region
8
9    METHOD:
10    i = P.firstPoint()
11    O_c = centroid(o)
12    az = azimuth(i,O_c)
13    lim1 = makeLine1(az,o,exteriorRing(buffer(o,d)))
14    lim2 = makeLine2(az,o,exteriorRing(buffer(o,d)))
15    reg = computeRegion(lim1,exteriorRing(o),lim2,
16            exteriorRing(buffer(o,d)))
17    return reg
```
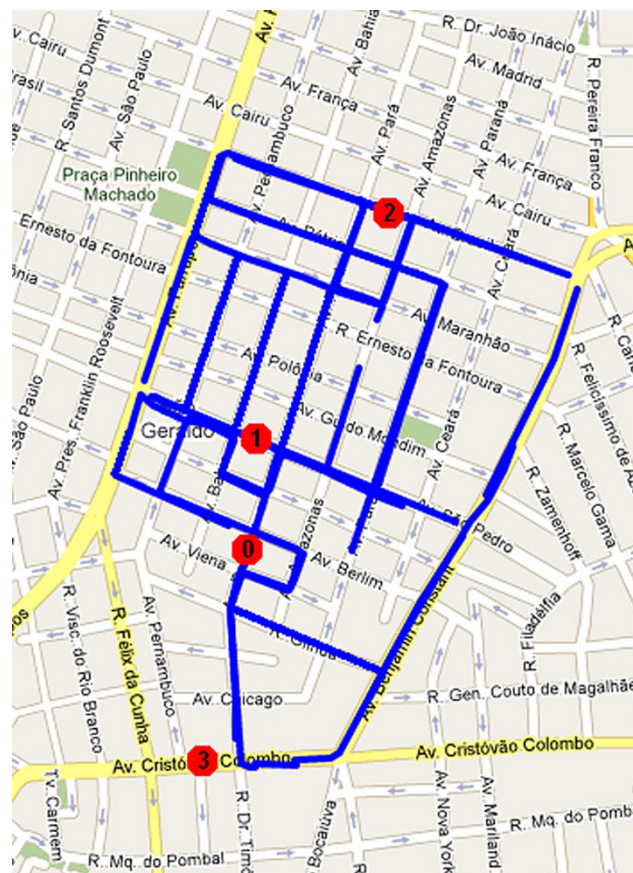
5 Experiments

In order to evaluate the results of the proposed algorithm, two different experiments where performed with real GPS data collected at the rate of one point per second. One dataset was collected by cars and the other one by pedestrians.

5.1 Experiment I—car trajectories

The car trajectories were collected in the city of Porto Alegre, with and without restrictions, i.e., avoiding or no spe-

Fig. 9 Car trajectories in Porto Alegre

cific regions mapped as the target objects. The target object may be any convex geometry, but in our experiments we considered this object as a circle with a radius of 20 meters and 80 meters around the target as the region of interest. As the minimal length for the subtrajectory directed to the target, we defined 8 meters. The 8 meters represents 10% of the size of the buffer around the target. Figure 9 shows these trajectories.

Table 1 shows the results of the first experiment that found avoidance patterns in 6 of 21 trajectories, with the respective global confidence.

Figure 10 shows the avoidance patterns for trajectories in Table 1. Figure 10(a) shows trajectory t_7 that intersects only the region of interest of target object 2, and Fig. 10(b) shows trajectory t_{13} that intersects only the region of interest of target 1. Both are cases of *strong* avoidance because the trajectories crossed the respective confidence incremental region. Figure 10(c) shows trajectory t_{18}, which

intersects all 4 regions of interest. For target objects 0, 1, and 2, it also intersects the confidence incremental region, therefore, getting a local confidence as *strong*. Because this trajectory does not intersect the confidence incremental re-

Table 1 Result of the first experiment considering 20 meters as the radius of the target object, 80 meters for the buffer of the region of interest around the target object, and 8 meters as the minimum length for the subtrajectory directed to the target

Tid	Global confidence
t_7	1
t_{13}	1
t_{18}	0.875
t_{21}	0.5
t_{15}	0.333
t_{11}	0.25

Fig. 10 Avoidance patterns for trajectories in Table 1
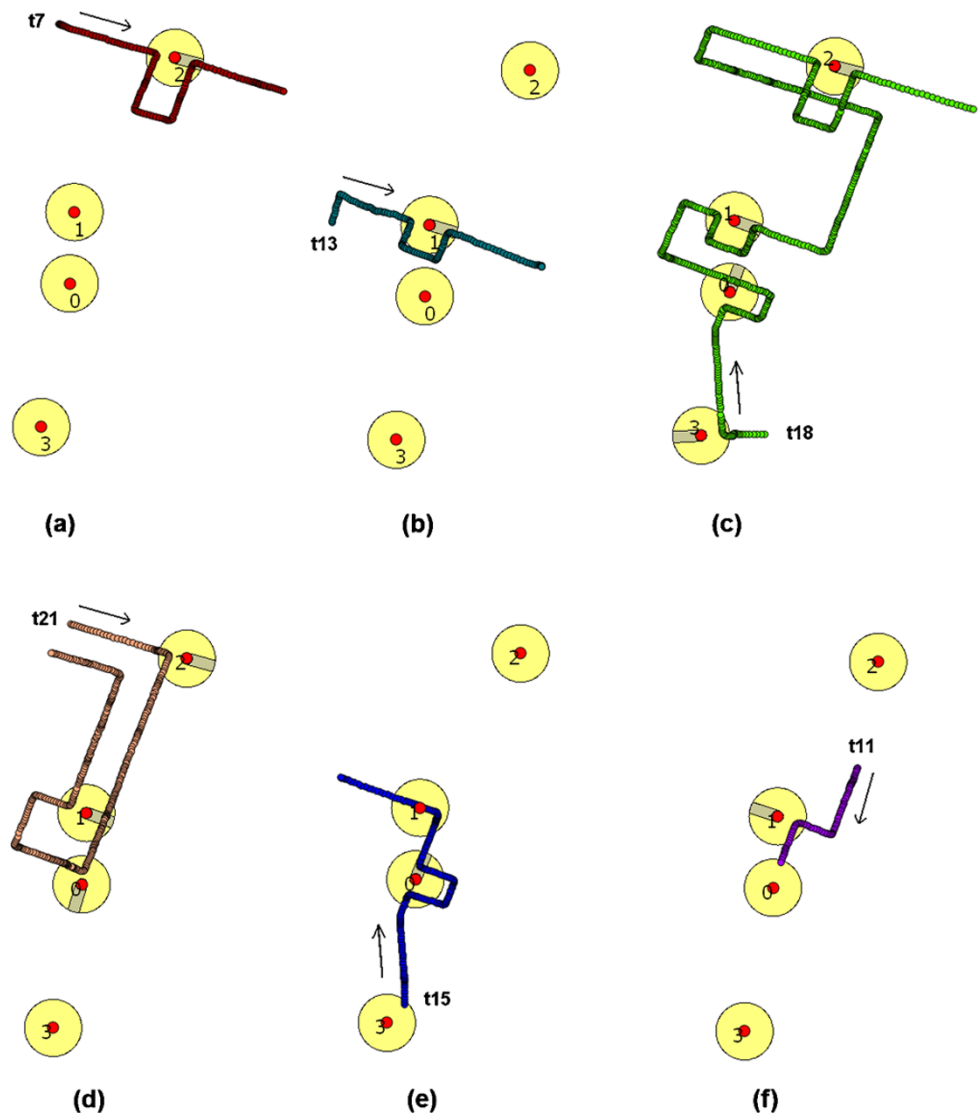
gion of the target object 3, it has a *weak* local avoidance at this point and, therefore, the global avoidance confidence is $(1 + 1 + 1 + 0.5)/4 = 0.875$.

Figure 10(d) shows trajectory t_{21}, which intersects 3 regions of interest, all with *weak* local confidence, since there was no valid intersection of any confidence incremental region. This trajectory intersects the region of interest of target 1 twice. For the first time, it does not have any subtrajectory directed to the target. The second intersection of trajectory t_{21} with the region of interest of target 1 had a subtrajectory directed to the target that was long enough, but after that the trajectory did not intersect the confidence incremental region.

Trajectory t_{15}, shown in Fig. 10(e), presents one *strong* local avoidance considering the target 0. In relation to the target 1, the local avoidance is *none*, since the trajectory intersects the target object. In relation to the target 3, the local avoidance is also *none*, i.e., there is no avoidance since the trajectory intersects the region of interest but it has no subtrajectory moving in direction to the target with a valid length. Therefore, the global confidence for this trajectory is 1/3 (0,333), having one strong avoidance and intersections with three regions of interest.

Finally, trajectory t_{11}, shown in Fig. 10(f), intersects the region of interest of the target objects 0 and 1, and has one *weak* local avoidance in relation to target 1 and no avoidance in relation to target 0, because the subtrajectory moving to the target was less than 8 m when the trajectory finished.

For this dataset, a total of 11 avoidance patterns were computed for all trajectories.

5.2 Experiment II—pedestrian trajectories

The second dataset is a set of 17 pedestrian trajectories collected at the Germania park in the city of Porto Alegre, considering four monitoring regions located on the crossing paths of the main routes in the park. Differently from the car trajectories that follow a road network, pedestrians may follow any directions anywhere. Although there are a few main routes, the objects move in aleatory ways in the park, therefore, these trajectories present characteristics very different from car trajectories. Indeed, the speed that a pedestrian moves may affect the density of the points, since the path followed by a pedestrian during 15 minutes, for instance, will be much shorter and denser than a car traveling in a highway during the same time period. As the GPS for the pedestrians was configured as for the car trajectories, i.e., the collection of a point every one second; this trajectory dataset is much denser than the previous one.

In this experiment, we considered 10 meters as the radius of the target object and 40 meters as the buffer of the region of interest, simulating a reasonable distance for a pedestrian to identify a camera and then to choose a change on her/his
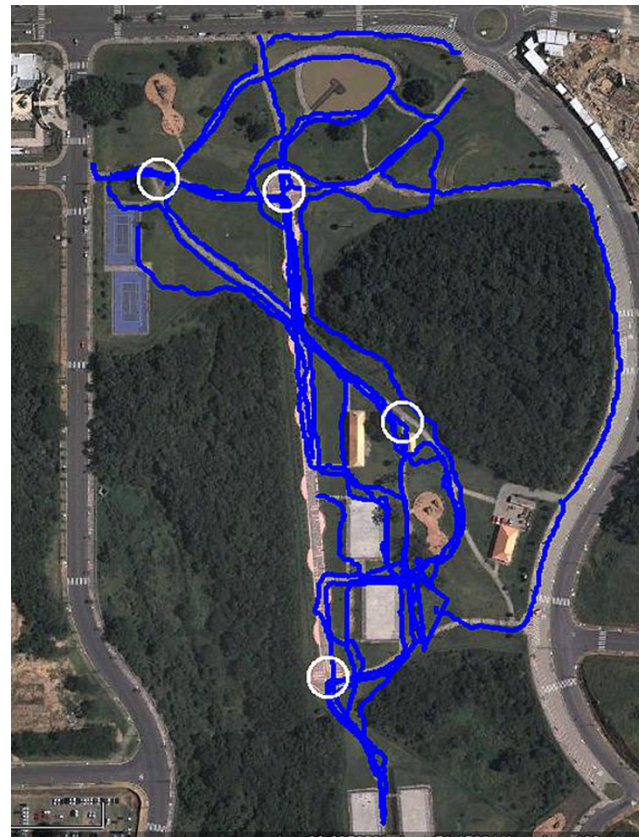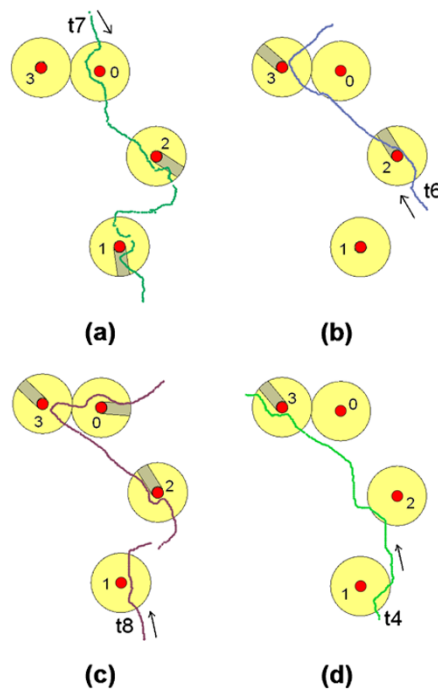

Fig. 11 Pedestrian trajectories in a park

Table 2 Result of the pedestrians experiment considering 10 meters as the radius of the target object, 40 meters for the buffer of the region of interest around the target object, and 4 meters as minimum length for the subtrajectory directed to the target

Tid	Global confidence
t_7	0.667
t_6	0.5
t_8	0.5
t_4	0.167

path. We used 4 meters as the minimal length for the subtrajectory directed to the target.

Figure 11 shows the visualization of these trajectories in Google Earth. The circles represent the monitoring regions defined as the target objects. After running the algorithm, eight avoidance patterns were found. Table 2 shows the result of the avoidance patterns of these trajectories, with its respective global confidence.

Among the trajectories with avoidance patterns, trajectory t_7 had the highest global confidence. As can be seen in Fig. 12(a), this trajectory avoided the target objects 1 and 2 with *strong* local confidence. The global confidence was reduced by the intersection of this trajectory with the region of interest of the target 0, where there was no subtrajectory

Fig. 12 Some pedestrian trajectories

directed to the target with a valid length. Additional experiments have demonstrated that when we use 2 meters for the minimal subtrajectory length directed to the target; the global confidence for this trajectory becomes maximal.

In trajectory t_6 (Fig. 12(b)), we identify one case of *weak* local avoidance, in relation to the target object 3, and one *strong* local avoidance in relation to the target object 2. This trajectory intersected the region of interest of target object 0, but has no subtrajectory directed to the target with 4 m length to characterize an avoidance. Then no avoidance was identified in relation to this target. This trajectory does not intersect the region of interest of target 1. Therefore, the avoidance level for the whole trajectory resulted in $(1 + 0.5 + 0)/3 = 0.5$.

Trajectory 8, shown in Fig. 12(c), intersects the four regions of interest but in relation to the target object 1 there is no valid subtrajectory directed to the target. Considering the targets 2 and 3, the trajectory does not intersect the confidence incremental region, receiving the value weak (0.5) as local avoidance. In relation to the target object 0, a strong avoidance has been identified because the trajectory satisfies all avoidance conditions.

In Fig. 12(d), trajectory 4 intersects three regions of interest, but in relation to the target objects 1 and 2 there is no subtrajectory directed to the target with the minimal length of 4 m, and hence has no avoidance with these targets. A weak avoidance exists with respect to the target object 3. The avoidance for the whole trajectory is then $(0 + 0 + 0.5)/3 = 0.167$.

Although in this paper we have shown the results of a few experiments on two datasets, we have performed more experiments and the results show that the algorithm correctly finds the existing avoidance patterns. Of course, the parameters—length of the buffer around the target object to define the region of interest, and the minimal length of the trajectory directed to the target—are very important and should be defined according to the specific application in hand.

6 Discussion

In this section, we present a discussion on the parameters of the proposed algorithm and an analysis of its complexity.

As in most algorithms, the choice of the values for the parameters is an important issue. First, the application domain should be considered. For instance, in the case of car trajectories in a city (i.e., in a road network), the trajectories are limited by the streets and, as a consequence, the choice of the parameters is facilitated. The size of the buffer of the region of interest and the length of the subtrajectory directed to the target should be defined considering that inside the region of interest there should be a distance greater than the minimum size of the subtrajectory directed to the target before the last crossing road, such that if the trajectory turns on that crossing road the avoidance will be detected. Considering this heuristic and knowing that the average length of the blocks in the region of experiment 1 was about 70 meters, and that the position of the target objects is in the middle of the blocks, we defined the buffer of the region of interest as 80 meters. To evaluate other values, we executed the same experiment with two other sets of parameters. First, we kept 80 meters as the buffer of the region of interest and we increased the minimum length of the subtrajectory directed to the target from 10 to 20 meters. There was no modification in the results. The same occurred with 30 meters as the minimum size of the subtrajectory directed to the target. This happens because the trajectories are constrained by the road network.

Considering the experiment with pedestrians, where there were no restrictions to the trajectories, we changed the minimum length of the subtrajectory directed to the target from 4 to 8 meters, and the results were exactly the same. In another experiment, we considered 60 meters as the buffer of the region of interest and 12 meters as the minimum length of the subtrajectory directed to the target. With these parameters, 3 new avoidances were detected, which is quite natural, since we considered a longer area and there were no limitations like a road network. Besides, the avoidance of trajectory 7 with target object 1 was no longer detected because the length of the biggest subtrajectory in the direction to the target was 11.74 meters (less than 12 meters).

Algorithm complexity, in the worst case, considering the existence of an R-Tree index and T trajectory points and N target objects, is $O(T \log N)$.

7 Conclusion and future work

Trajectory data are becoming more and more common in daily life. Several works have been developed to extract interesting information and knowledge from these data, as well as trying to infer the behavior of the moving object. Most works have focused on the common behavior of groups of trajectories of different moving objects.

In this paper, we contribute to advance trajectory data analysis one step forward proposing a novel work to identify avoidance behavior in trajectories. Our work identifies avoidance behavior of individual objects in relation to existing static targets that could be security cameras, police offices, or police controllers, and so on.

The proposed method may be useful in several application domains like the monitoring of prisoners in semi-open (prerelease) level, traffic management, and security control.

As we use raw trajectory data, without considering semantic information, the proposed heuristics cannot know if a trajectory deviated the target with the unique intention to avoid it, or if it is the normal path of the moving object. However, this method may help users of different applications to investigate these trajectories.

At this point, we have not considered any semantics to provide more information about the avoidance patterns. However, as future ongoing work, we are investigating new measures to ensure if an avoidance was intentional or forced by an event, like a blocked street, therefore, reducing false positives.

References

1. Agrawal R, Srikant R (1994) Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C (eds) VLDB. Morgan Kaufmann, San Mateo, pp 487–499
2. Alvares LO, Bogorny V, Kuijpers B, de Macedo JAF, Moelans B, Vaisman A (2007) A model for enriching trajectories with semantic geographical information. In: ACM-GIS. ACM Press, New York, pp 162–169
3. Andersson M, Gudmundsson J, Laube P, Wolle T (2008) Reporting leaders and followers among trajectories of moving point objects. GeoInformatica 12:497–528. doi:10.1007/s10707-007-0037-9
4. Baglioni M, de Macêdo JAF, Renso C, Trasarti R, Wachowicz M (2009) Towards semantic interpretation of movement behavior. In: Sester M, Bernard L, Paelke V (eds) AGILE conference, lecture notes in geoinformation and cartography. Springer, Berlin, pp 271–288
5. Bogorny V, Kuijpers B, Alvares LO (2009) ST-DMQL: a semantic trajectory data mining query language. Int J Geogr Inf Sci 23:1245–1276
6. Cao H, Mamoulis N, Cheung DW (2006) Discovery of collocation episodes in spatiotemporal data. In: ICDM. IEEE Comput Soc, Los Alamitos, pp 823–827
7. Dodge S, Weibel R, Lautenschutz A (2008) Towards a taxonomy of movement patterns. Inf Vis 8:240–252
8. Giannotti F, Nanni M, Pinelli F, Pedreschi D (2007) Trajectory pattern mining. In: Berkhin P, Caruana R, Wu X (eds) KDD. ACM Press, New York, pp 330–339
9. Gudmundsson J, van Kreveld MJ (2006) Computing longest duration flocks in trajectory data. In: de By RA, Nittel S (eds) GIS. ACM Press, New York, pp 35–42
10. Hägerstrand T (1970) What about people in regional science?. Pap Reg Sci 24(1):6–21
11. Kim DJ, Park KH, Bien Z (2007) Hierarchical longitudinal controller for rear-end collision avoidance. IEEE Trans Ind Electron 54:805–817
12. Laube P, Imfeld S, Weibel R (2005) Discovering relative motion patterns in groups of moving point objects. Int J Geogr Inf Sci 19(6):639–668
13. Laube P, van Kreveld M, Imfeld S (2005) Finding REMO: detecting relative motion patterns in geospatial lifelines. Springer, Berlin
14. Lee SW, Lee BH, Lee KD (1999) A configuration space approach to collision avoidance of a two-robot system. Robotica 17:131–141
15. Liu YH, Shi CJ (2005) A fuzzy-neural inference network for ship collision avoidance. In: Proceedings of 2005 international conference on machine learning and cybernetics. IEEE Comput Soc, Los Alamitos, pp 4754–4754
16. Nedevschi S, Bota S, Tomiuc C (2009) Stereo-based pedestrian detection for collision-avoidance applications. Trans Intell Transp Syst 10:380–391
17. Ong R, Wachowicz M, Nanni M, Renso C (2010) From pattern discovery to pattern interpretation in movement data. In: Fan W, Hsu W, Webb GI, Liu B, Zhang C, Gunopulos D, Wu X (eds) ICDM workshops. IEEE Comput Soc, Los Alamitos, pp 527–534
18. Palma AT, Bogorny V, Alvares LO (2008) A clustering-based approach for discovering interesting places in trajectories. In: ACM-SAC. ACM Press, New York, pp 863–868
19. Rocha JAMR, Times VC, Oliveira G, Alvares LO, Bogorny V (2010) DB-SMOT: a direction-based spatio-temporal clustering method. In: IEEE conference of intelligent systems. IEEE Press, New York, pp 114–119
20. Shandy S, Valasek J (2001) Intelligent agent for aircraft collision avoidance. In: Proceedings of AIAA guidance, navigation, and control conference. American Institute of Aeronautics and Astronautics, Washington, pp 1–11
21. Suh SH, Bishop AB (1988) Collision-avoidance trajectory planning using tube concept: analysis and simulation. J Robot Syst 5(6):497–525
22. Suh SH, Kim MS (1992) An algebraic approach to collision-avoidance trajectory planning for dual-robot systems: formulation and optimization. Robotica 10(02):173–182