Missing Information in Trajectories: An Application to Avionics

Francisco Moreno, Sebastián Múnera, and Jesús Hernández

Abstract—**A trajectory records the evolution of the position of a moving object in a space during a time interval. In Spaccapietra's trajectory model, trajectories are segmented in subintervals called stops and moves. On the other hand, during some periods failures can occur in the transmission of data of the trajectory causing missings of information. In this paper, we extend Spaccapietra's model by incorporating the missing information as a component of a trajectory. We consider this issue not only with regard to the object position but also with regard to other attributes of the trajectory (complementary attributes). We propose a classification for these attributes, depending on whether they are constant or variable during the stops and the moves. We also propose two algorithms: i) to convert a sequence of observations of a trajectory into stops, moves and missings. ii) to check that the data recorded for the attributes whose value must be constant is consistent.**

Index Terms—**Avionics, spatiality, temporality, trajectories, missing data.**

I. Introduction

Today, thanks to GPS (Global Positioning System) [1], [2]; and other related technologies, it is possible to record information of the trajectory followed by moving objects like people, animals, airplanes, ships, natural phenomena. Indeed, recent advances of these technologies, along with their low cost, have generated an explosion of spatio-temporal data that requires appropriate tools for their analysis. These analysis can help understand both individual and group behavior of moving objects in areas such as population movements, monitoring of animal migration, air and maritime traffic control, movement of natural phenomena (hurricanes, tsunamis), among others.

Informally, a trajectory is the record of the evolution of the position of a moving object in a space during a time interval [3]. A trajectory can be represented as a set of consecutive observations in time, i.e., a set of 3-tuples (x, y, t), where x and y represent the object position and t the observation time: $\{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)\}$ where $t_j > t_i$. $\forall j > i$.

In a trajectory there can be periods during which the object is fixed (its position does not change). In [3], the authors proposed a model to represent trajectories, where the consecutive observations during which the object was fixed make up a stop, and the consecutive observations during which the object was moving make up a move.

However, during some periods, failures can occur in the transmission of data of the trajectory. As a consequence, we can speak about missings of information in a trajectory. Some works consider this issue. [4] proposes a method to update the position of a moving object by comparing its real position with a precalculated one. Other works present some models to calculate the position of a moving object considering the uncertainty in the measurement and sampling [5]-[10]. Some mechanisms are also provided, e.g., interpolations to query the object position between consecutive observations. Some of these works [5]-[8] also present query operators which consider the uncertainty of the object position.

TABLE I: Data of an Airplane Trajectory with Different Attributes Values as a Function of time

Time (hh:mi)	Position (x, y) (km)	# Passengers	Airplane Speed (m/s)	Wind Speed (m/s)	Airplane Id
11:00	(110,120)	10	0	5.7	123
11:30	(110,120)	49	0	7.2	123
12:00	(110,120)	68	0	8.2	123
12:30	(110,120)	72	0	8.7	123
13:00	(110,120)	89	0	8.7	123
13:30	(110,120)	110	59.2	6.7	123
14:00	(210,300)	110	74.4	10.8	123
...	...	...	...	...	...
21:00	(420,560)	105	125.6	6.2	123
21:30	(490,610)	105	61.1	5.1	123

On the other hand, it is possible to enrich a trajectory with complementary data specific to the application. For example, consider the trajectory of an airplane. For each of its observations, it is possible to record data such as speed (of the plane and the wind), and air temperature, among others. However, previous works only focus on the missing information regarding the object position and do not consider this analysis for other attributes. In fact, none of the previous works considers complementary attributes as inherent elements of a trajectory. In this paper, we discuss the missing information regarding such attributes. For this purpose, we propose a classification for the complementary attributes depending on whether they are constant or variable during the stops and the moves of a trajectory, and thereby analyze their behavior during the missings. In addition, we propose two algorithms. The first one converts a sequence of observations of a trajectory into stops, moves, and missings. The second one checks that the information recorded for the attributes whose value must be constant (either during the trajectory or a stop or a move) is consistent. Note that in our model, unlike that of [3], both the missings and the complementary attributes are incorporated as components of a trajectory.

Manuscript received May 1, 2012; revised June 2, 2012.

The authors are with Universidad Nacional de Colombia, Sede Medellín(e-mail: fjmoreno. @unal.edu.co, sfmunera. @unal.edu.co, jawhernand. @unal.edu.co)

motivating example and our trajectory model along with the algorithms for conversion and consistency checking. Finally, we conclude the paper and outline future work.

II. MOTIVATING EXAMPLE AND PROPOSED MODEL

Consider the trajectory of an airplane during a day. Complementary information of the trajectory such as airplane identification, number of passengers, speed of the airplane, and wind is shown in Table I with one observation every 30 minutes.

Let us assume now that some failures occurred in the transmission of the GPS in the intervals: (11:30, 13:00), (14:00, 16:00), (16:30, 18:00), and (18:30, 21:00). To represent this missing information, first we present the essential elements of the Spaccapietra's model. Then, we extend this model in order to support the missing information and the complementary attributes of a trajectory.

In Spaccapietra's model, a trajectory is represented by stops and moves. A stop is defined by an interval I [start time, end time] and by a position (x, y). A move is defined by an interval I [start time, end time] and by a function: Position(t) → (x, y), $\forall t \in I$. Both start and end time of a move are delimited by two stops, so that the end time of the first stop (the lowest in time from the two stops) coincides with the start time of the move and the start time of the second stop coincides with the end time of the move.

In addition to the stops and the moves, a trajectory has a start point and an end point, each one represented by the coordinates (x, y) and an instant t. The start time of a move may coincide with the instant of the start point of the trajectory and the end time of a move may coincide with the instant of the end point of the trajectory.

Next, we propose a classification for the attributes of a trajectory, depending on whether the object is moving or not, see Fig. 3. An attribute can be classified as:

- *Constant in stop and constant in move* (CS-CM): It is an attribute whose value does not change during the trajectory, e.g., the airplane identification.
- *Variable in stop and variable in move* (VS-VM): It is an attribute whose value may change regardless of whether the object is moving or not, e.g., the wind speed.
- *Constant in stop and variable in move* (CS-VM): It is an attribute whose value is constant while the object is fixed, but that may change when the object is moving, e.g., the airplane speed is zero while it is fixed, but may change when the airplane is moving.
- *Variable in stop and constant in move* (VS-CM): It is an attribute whose value is constant while the object is moving, but that may change when the object is fixed, e.g., the number of passengers may change while the airplane is fixed (note that this value is very unlikely to change when the airplane is moving).

Thus, for each complementary attribute whose value may change during a stop or a move, its information is stored in each observation included in the stop or in the move respectively, i.e., there is a time-dependent function: a(t) → Dom(a), $\forall t \in I$, where I is the corresponding interval of the stop or the move, and Dom(a) is the domain of a.

In order to deal with missing information, we propose to incorporate a missing component to the Spaccapietra's trajectory model, i.e., a part of a trajectory during which no information was received from the object. Thus, a trajectory is represented by a start point, stops, moves, missings, and an end point.

Formally, a trajectory is a 3-tuple (Obj, Comp, Attr) where Obj is the moving object, Comp is a set of components $\{c_1, c_2 \ldots c_n\}$ and Attr is a set of attributes $\{a_1, a_2, \ldots, a_m\}$.

Each component c_i belongs to one type, i.e., there exists a function Typecomp: Comp → TC, where TC = {start, stop, move, missing, end}. If Typecomp(c_i) = start or end then the temporal extention of c_i is an instant and it is an interval otherwise.

In the Comp set, there must exist a single component with start type and one with end type (because a trajectory only has a start and an end). The instant of the start component and of the end component represents both the start time and the end time of the trajectory, respectively.

The components in Comp make up a temporal sequence, i.e., the end instant of each component (except for the component of end type) is equal to the start instant of another component. Thus, if the end instant of a component c_i corresponds to the start instant of a component c_j, we say that c_i is the predecessor component of c_j and c_j is the successor component of c_i. Each component has only one successor component (except the component of end type that has no successor component) and a predecessor component (except the component of start type that has no predecessor component). The functions Predecessor and Successor, both with signature Comp → Comp, return the predecessor and successor component of a component, respectively. Given two components c_i and c_j, where Predecessor(c_j) = c_i, it follows that Typecomp(c_i) $\neq$ Typecomp(c_j).

In addition, each attribute a_i is associated with a set of values, i.e., a domain Dom(a_i). Each attribute a_i also has a type, i.e., there exists a function Typeattr: Attr → TA, where TA = {CS-CM, VS-VM, CS-VM, VS-CM}. The set Attr has an attribute of type CS-VM that represents the object position. The rest of the attributes (if any) are called complementary. The following rules apply to the values of the attributes:

- If Typeattr(a_i) = CS-CM then the value of a_i during the whole trajectory is constant.
- If Typeattr(a_i) = CS-VM then the value of a_i during the temporal extention of a component c_i, where Typecomp(c_i) = stop, is constant.
- If Typeattr(a_i) = VS-CM then the value of a_i during the temporal extention of a component c_i, where Typecomp(c_i) = move, is constant.
- If Typeattr(a_i) $\neq$ CS-CM, then the values of a_i during the temporal extention of a component c_i, where Typecomp(c_i) = missing, are unknown.

In Fig. 1 we show the Algorithm 1 which receives a sequence of observations of a trajectory including their complementary attributes, if the interval between two consecutive observations is greater than the set time (phase) for the reception between the observations; the interval is marked as missing. Otherwise, the sets of consecutive observations in time whose position does not change are marked as stops and those whose position change are marked as moves. The algorithm returns the trajectory represented by the start point, stops, moves, missings, and end point.

Algorithm 1: Generates a trajectory made up of start point, stops, moves, missings, and end point.

1) Input: T0: Sequence of n observations of a trajectory.
2) Phase: Time between observations.
3) Output: T: Trajectory.
4) start point ← first observation of T0
5) current component ← start point
6) for k ← 2 to n
7) if time of observationk - time of observationk-1 > phase then
8) Append current_component to T
9) Let ms be a new missing
10) Append observationk to ms
11) Append observationk-1 to ms
12) current component ← ms
13) else
14) if position of observation = position of observationk-1 then
15) if Type comp(current component) ≠ "stop" then
16) Append current component to T
17) Let s be a new stop
18) Append observationk-1 to s
19) current component ← s
20) end if
21) Append observation to s
22) 22: else
23) if Type comp(current component) ≠ "move" then
24) Append current component to T
25) Let mv be a new move
26) Append observationk-1 to mv
27) current component ← mv
28) end if
29) Append observation to mv
30) end if
31) end if
32) end for
33) Append current component to T
34) endpoint ← last observation of T0
35) Append endpoint to T

Fig. 1. Algorithm to build a trajectory.

Next, in Fig. 2 we present a second algorithm which checks that the information recorded for the attributes whose value is constant (either during the trajectory or a stop or a move) is consistent. For example, Algorithm 2 checks that the value of an attribute of type CS-VM is the same during all the observations of a stop. If indeed this value remains constant, this value is stored only once in the stop (this value is a "common factor").

Algorithm 2: Checks consistency of constant attributes during moves and removes redundant values.

1) Input: T: a trajectory made up of start point, moves, stops, missings, and end point.
2) Output: T: trajectory T with redundant values removed.
3) // Check consistency of CS-CM attributes:
4) ∀attr a_i ∈ T | Typeattr(a_i) = "CS-CM"
5) if value of a_i in each observation of T is the same then
6) Store value of a_i only once in T
7) else
8) Print "Attribute: " a_i "is expected to be constant during the
9) trajectory"
10) end if
11) end ∀
12) // Check consistency of CS-VM attributes:
13) ∀attr a_i ∈ T | Typeattr(a_i) = "CS-VM"
14) ∀stop s ∈ T
15) if value of a_i in each observation of s is the same then
16) Store value of a_i only once in s
17) else
18) Print "Attribute: " a_i "is expected to be constant during stop: " s
19) end if
20) end ∀
21) end ∀
22) // Check consistency of VS-CM attributes:
23) ∀attr a_i ∈ T | Typeattr(a_i) = "VS-CM"
24) ∀move m ∈ T
25) if value of a_i in each observation of m is the same then
26) Store value of a_i only once in m
27) else
28) Print "Attribute: " a_i "is expected to be constant during move: " m
29) end if
30) end ∀
31) end ∀

Fig. 2. Algorithm for checking trajectory attributes.

III. Conclusions and Future Work

In this paper we extended Spaccapietra's trajectory model [3] with the aim of representing the missing information in the trajectory of a moving object. In the Spaccapietra's model, the consecutive observations during which the object remained fixed make up a stop and the consecutive observations during which the object was moving make up a move. In our work we include the missing information as a component of the model. Our proposed model considers the missing information not only regarding the object position but also with regard to other attributes of the trajectory (complementary attributes). A classification of these attributes, depending on whether they are constant or variable during the stops and the moves, was also proposed. Starting from their classification, their behavior was analyzed during the missings of information.

Two algorithms were also proposed. The first one converts a sequence of observations of a trajectory into stops, moves, and missings. The second one checks that the data recorded for the attributes whose value must be constant (either during the trajectory, or a stop, or a move) is consistent.

Our contributions can be summarized as follows. i) the proposal of a classification for the complementary attributes of a trajectory according to their behavior during the stops and moves, ii) the representation of the missing information as a component in a trajectory, iii) the algorithm that generates a trajectory represented according to our model from a sequence of observations, and iv) the algorithm that checks that the data recorded for the attributes whose value must be constant is consistent.

As future work, we plan to validate our proposal with real data and develop an appropriate method for "reconstructing" the values of the attributes during the missings, possibly using data imputation techniques [11]-[14].

REFERENCES

[1] B. H. Wellenhof, H. Lichtenegger, and J. Collins, "Global positioning system: theory and practice," *4th ed*, pp. 11–23, Springer-Verlag, Wien, Austria. 1997.

[2] A. Leick. *GPS satellite surveying*, 3rd ed, 72–85, John Wiley and Sons, New York, USA. 2004.

[3] S. Spaccapietra, C. Parent, M. L. Damiani, J. A. Fernandes de Macêdo, F. Porto, and C. Vangenot, "A conceptual view on trajectories," *Data and Knowledge Engineering*, vol. 65, no. 1, pp. 126-146, 2008.

[4] O. Wolfson and H. Yin, "Accuracy and Resource Consumption in Tracking and Location Prediction," In: *Proceedings of the 8th International Symposium on Spatial and Temporal Databases*. Santorini, Greece, pp. 325–343, 2003.

[5] D. Pfoser and C. S. Jensen, "Capturing the Uncertainty of Moving-Object Representations," In: *Proceedings of the 6th International Symposium on Advances on Spatial Databases*, Hong Kong, China. pp. 111–132, 1999.

[6] G. Trajcevski, O. Wolfson, F Zhang, and S. Chamberlain, "The Geometry of Uncertainty in Moving Object Databases," In: *Proceedings of the 8th International Conference on Extending Database Technology*. Prague, Czech Republic, pp. 233–250, 2002.

[7] G. Trajcevski, O. Wolfson, H. Cao, H. Lin, F. Zhang, and N. Rishe, "Managing Uncertain Trajectories of Moving Objects with DOMINO," In: *Proceedings of the 4th International Conference on Enterprise Information Systems*, Ciudad Real, Spain, pp. 218–225, 2002.

[8] O. Wolfson, "Moving Objects Information Management: The Database Challenge," In: *Proceedings of the 5th Workshop on Next Generation Information Technologies and Systems*. Caesarea, Israel. 2002, pp. 15–26.

[9] V. Almeida and R. Hartmut, "Supporting Uncertainty in Moving Objects in Network Databases," In: *Proceedings of the 13th annual ACM international workshop on Geographic information systems*, Bremen, Germany, pp. 31-40, 2005.

[10] O. Abul, F. Bonchi and M. Nanni, "Never Walk Alone: Uncertainty for Anonymity Inmoving Objects Databases," In: *Proceedings of the 24th IEEE International Conference on Data Engineering*. Cancun, Mexico, pp. 376–385, 2008.

[11] A. Farhangfar, L. A. Kurgan, and W. Pedrycks, "A novel framework for imputation of missing values in databases," *IEEE Transactions on Systems Man and Cybernetics*, vol. 37, no. 5, pp. 692–709, 2008.

[12] S. Laaksonen, "How To Find The Best Imputation Technique. Tests with Three Methods," In: *Proceedings of the International Conference on Survey Nonresponse*. Portland, Oregon, USA. 1999.

[13] R. J. Little and D. B. Rubin, *Statistical analysis with missing data*, 2nd ed, John Wiley & Sons, New York, USA. pp. 59–89, 2002.

[14] C. M. Musil, C. B. Warner, P. K. Yobas, and S. L. Jones, "A comparison of imputation techniques for handling missing data," Western Journal of Nursing Research, vol. 24, no. 7, pp. 815–829, 2002.