Abstract
For many years trajectory data have been treated as sequences of space-time points or stops and moves. However, with the explosion of the Internet of Things and the flood of big data generated on the Internet, such as weather channels and social network interactions, which can be used to enrich mobility data, trajectories become more and more complex, with multiple and heterogeneous data dimensions. The main challenge is how to integrate all this information with trajectories. In this article we introduce a new concept of trajectory, called multiple aspect trajectory, propose a robust conceptual and logical data model that supports a vast range of applications, and, differently from state-of-the-art methods, we propose a storage solution for efficient multiple aspect trajectory queries. The main strength of our data model is the combination of simplicity and expressive power to represent heterogeneous aspects, ranging from simple labels to complex objects. We evaluate the proposed model in a tourism scenario and compare its query performance against the state-of-the-art spatio-temporal database SECONDO extension for symbolic trajectories.

1 | INTRODUCTION

For many years, moving object database research has focused on how to represent, store and query the physical movement of objects, limiting movement data to the spatial position of an object in time. The physical movement is well known as raw trajectory, and is generally represented as a sequence of points $T = \langle p_1, p_2, \ldots, p_n \rangle$, with $p_i = (x_i, y_i, t_i), p_i \in T$, where $x, y$ denote the position of the object in space at time instant $t$. 
With the explosion of the Internet of Things and the flood of big data generated on the Internet, such as weather channels and social network interactions (e.g., Flickr, Facebook, Twitter, Foursquare), it is now possible to collect huge volumes of movement data about people, animals, and objects such as cars, buses, and drones. Sensors installed either indoors (e.g., smart homes) or outdoors allow the collection of data about place, such as temperature, air pollution, noise, and luminosity, or about the object that is moving around or inside this place, such as heart rate (with a smart watch), emotional status (with a microphone that analyses voice intonation), blood pressure, and sleeping stages. By collecting all this information we have a new type of movement data, (i.e., a trajectory enriched with different semantic aspects). As an example, let us consider the trajectory of Peter, shown in Figure 1.

Peter stays at home from 11 p.m. to 8 a.m., goes to work from 8:30 a.m. to 6 p.m., and then goes for dinner at a Japanese restaurant. He has a smart watch that constantly collects his heart rate and sleeping stages at home during the night. He goes to work on foot when the weather is sunny, and he tweets a message that characterizes his mood. His working place is a smart office equipped with numerous sensors that collect information about the place on noise, temperature, air pollution and humidity, as well as the emotional status of the staff using microphones. After work, the weather changes to rainy, and Peter decides to take a taxi to go to a Japanese restaurant. The restaurant has its own attributes, such as opening hours, average price, spatial location, and reviews.

We can observe from Figure 1 that a trajectory is a complex object with numerous complex data dimensions that are contextual to the movement and heterogeneous in form, which we define in this article as aspects. The more aspects we have, the more complete is the representation of the real movement of an object, and the more useful and interesting information we can infer about objects and places. The challenge is how to integrate all this heterogeneous information in a single trajectory representation, and the two main questions we want to answer in this article are as follows. First, is it possible to define a data model that is simple in structure, but generic enough to represent any aspect related to movement, and covers a large number of applications? Second, is there a way to efficiently query and extract patterns from data represented in this model?

A number of data models have been proposed to consider limited semantic information about movement, such as the well-known model of stops and moves (Spaccapietra et al., 2008) and the generalized CONSTANT model (Bogorny, Renso, de Aquino, de Lucca Siqueira, & Alvares, 2014). Based on these data models, works such as Fileto et al. (2015) and Nogueira, Braga, de Oliveira, and Martin (2018) propose frameworks and ontologies to enrich trajectories with Linked Open Data (LOD). Existing trajectory data models essentially add semantic labels to trajectories, such as the means of transportation, the aim of the trip, and the category of places visited (Bogorny et al., 2014). However, these models have some limitations. First, they focus on the conceptual representation only, and do not go deeper into the logical level and storage technologies. Furthermore, they do not consider dynamic and complex aspects that involve movement, not being flexible to represent heterogeneous aspects.

We claim that multiple aspect trajectories represent a new view of trajectories, and a new paradigm for mobility data. These aspects not only are simple semantic labels, but also may be complex objects and/or heterogeneous information intrinsically associated to the physical traces of the moving objects.
In this article we introduce the concept of multiple aspect trajectory and propose a novel approach to modeling trajectories of this kind called MASTER. MASTER comprises a conceptual and a logical data model for multiple aspect trajectories, as well as a storage solution that is appropriate for multiple aspect trajectory queries.

The main contributions of this article are summarized as follows. First, our concept of multiple aspect trajectory generalizes the state-of-the-art semantic trajectory definition into a more complex but realistic trajectory based on the notion of aspect. We define different kinds of aspects, and we discuss the semantic meaning of an aspect to distinguish between different possible meanings for an aspect name. We provide a conceptual data model for multiple aspect trajectories with large expressive power, which combines simplicity with a powerful representation of different types of moving objects and a variety of spatial, temporal and semantic aspects that are relevant to a vast range of applications. Second, we convert the conceptual data model to a logical data model in the Resource Description Framework (RDF) format to adhere to the Semantic Web standards (Berners-Lee, Hendler, & Lassila, 2001). Third, we use a triplestore based on NoSQL databases for maintaining RDF data, since these databases represent a new and efficient technology for maintaining and querying very large data volumes, including trajectory data. Finally, we conduct a twofold evaluation: we first analyze the expressive power of MASTER for representing multiple aspect trajectories in a tourism application and discuss different kinds of queries on multiple aspect trajectories with increasing complexity, and then carry out a query performance comparison with the well-known spatio-temporal database SECONDO extended for symbolic trajectories (Güting, Valdés, & Damiani, 2015).

The remainder of this article is organized as follows. Section 2 presents related work. Section 3 introduces multiple aspect trajectories and details the MASTER approach. Section 4 is dedicated to MASTER evaluations. Section 5 concludes.

2 | RELATED WORK

In this section we discuss existing initiatives for trajectory data modeling. Hornsby and Cole (2007) proposed to model trajectories as sequences of events. Although this approach leads to the notion that movement is based on a sequence of happenings, the main drawback is that events are all homogeneous elements, so that it is not possible to represent the diverse heterogeneous information extracted from different data sources (e.g., social media, sensor networks, weather channels) that characterize different aspects related to a movement.

Spaccapietra et al. (2008) introduced the new concept of semantic trajectories, representing movement data as a sequence of stops and moves. Alvares et al. (2007) was the first instantiation of the model of stops and moves, and originated several works in which stops have been labeled as episodes, points of interest (POIs), hot spots, etc. Bogorny, Avancini, de Paula, Kuplich, and Alvares (2011) proposed an architecture to store and analyze semantic trajectories, but it is also limited to trajectories represented as stops and moves. Yan, Macedo, Parent, and Spaccapietra (2008) defined an ontological infrastructure for trajectory data. They consider spatial ontologies to enrich trajectories from the geometrical point of view (e.g., spatial relationships), geographical ontologies to enrich the trajectories from the semantic point of view (e.g., POIs such as places and buildings), and domain ontologies to enrich the trajectories with specific domain information (e.g., traffic management, bird migration). This ontology infrastructure can be used, for example, to detail some aspects in our proposed MASTER data model, as discussed in Section 3.2.

Yan, Chakraborty, Parent, Spaccapietra, and Aberer (2011) proposed the SEMITRI framework for labeling stops and moves, and Parent et al. (2013) present a survey of semantic trajectory modeling that in general focuses also on stops and moves. Bogorny et al. (2014) introduced a semantically richer conceptual model, called CONSTANT, in which a trajectory can be associated to a limited set of predefined aspects: the activities performed by the object, the means of transportation, the POIs visited, the aim of the trip and some behavior-specific patterns. This model is limited to a subset of aspects that are related to subtrajectories or to the entire trajectory, which
are also presented as labels, and only the weather conditions and the POIs can be associated to trajectory points. Additionally, CONSTANT does not support social media data such as relationships between moving objects, opinions and feelings, nor does it support aspect types nor the concept of permanent aspect, as introduced in Section 3.1.

Fileto et al. (2015) presented BAQUARA to instantiate the CONSTANT model. The idea is to annotate semantic trajectories with LOD. It is based on a predefined ontological model of semantic trajectory, where we can find a preliminary notion of “aspect.” Here an aspect is a semantic label associated to the trajectory points. However, BAQUARA does not support relationships between moving objects and does not propose a solution for storing and querying huge volumes of trajectories, aspects or moving objects.

More recently, Ruback, Casanova, Raffaetà, Renso, and Vidal (2016) proposed an alternative framework for enriching movement data with LOD. While BAQUARA is based on a predefined ontology representing, in a monolithic way, the enriched trajectory, in Ruback et al. the enrichment is dynamically created through the use of ontology mashups. Despite these innovations, this approach suffers from the same limitations as BAQUARA.

Noël, Villanova-Oliver, Gensel, and Quéau (2015) introduced the idea of modeling trajectories from different points of view. In this approach, each point of view of a trajectory is represented as a sequence of events and episodes. Their data model, however, is limited to representing trajectory as sequences of events and episodes, and does not consider the relationships between moving objects, between trajectories or aspects. It provides an efficient solution for storing and querying the so-called “life trajectories.” Ferrero, Alvares, and Bogorny (2016) raised the need to consider multiple and heterogeneous aspects that characterize movement, and to integrate all these data together for a more complete and realistic analysis of mobility data. This work, however, shows only the need to integrate multiple aspects, and presents neither a conceptual data model nor a physical strategy for efficient multiple aspect trajectory querying.

In terms of spatio-temporal database systems, HERMES (Pelekis, Theodoridis, Vosinakis, & Panayiotopoulos, 2006) and SECONDO (Güting et al., 2005) are two well-known software prototypes, but developed for raw trajectories. As a trajectory in SECONDO is represented as a single data type moving object, it does not support the semantic enrichment of individual trajectory points. To overcome this limitation, another representation of trajectories is used to store semantic trajectories in SECONDO, referred to as symbolic trajectories (Valdés, Damiani, & Güting, 2013). These trajectories are represented as subtrajectories annotated with single flat labels indexed by time intervals. In order to represent, for example, visited POIs and the transportation mode, the same trajectory must be segmented by POIs and by transportation mode.

In summary, apart from the work of Pelekis and Güting that focused on the physical level for representing and querying mobility data, the other previous works have focused on the conceptual representation only.

### 3 | THE MASTER APPROACH

This section presents our approach to modeling multiple aspect trajectory data called MASTER. We define a conceptual and a logical model for multiple aspect trajectories, and we also detail the storage solution adopted.

The main strength of our conceptual model is the combination of simplicity and expressive power for representing aspects. An aspect may be related to a moving object, to the entire trajectory or any trajectory point, and may hold any type of data, ranging from simple labels to complex objects. It is important to emphasize that the focus of this article is not on how to obtain these heterogeneous aspects, which may be provided by specific extraction and transformation tools, but on how trajectory data are related to these aspects. Our focus here is on how to represent any aspect data, independent of the application domain, in a single simple data model.

For the logical model, we consider a graph-based representation (the RDF standard; Pan, 2009) that is generic enough to model trajectories and aspects extracted from heterogeneous data sources, such as geolocated structured record files and geolocated social media posts (e.g., tweets). Finally, we consider NoSQL databases for efficient storage and retrieval of large amounts of trajectory data. Our inspiration comes from the polyglot persistence
approach (Sadalage & Fowler, 2013), which states that a conceptual data model can be split and mapped to several database models to maximize query performance.

3.1 The conceptual model

In this subsection we introduce a conceptual data model for multiple aspect trajectories, which is shown in Figure 2. We start the description of the model with the new concept introduced in this article: the aspect.

An aspect is a real-world fact that is relevant for trajectory data analysis, and it is characterized by an aspect type. For instance, the aspect train belongs to an aspect type transportation mode, and the aspect rainy belongs to an aspect type weather condition. An aspect type has a set of attributes and it may also be a subtype of a more general aspect type, allowing the modeling of an aspect type subtypeOf hierarchy, such as POI←accommodation←hotel.

More formally, an aspect type is defined as follows.

**Definition 1** An aspect type asp_type = (desc, ATT, asp_supertype) is a categorization of a real-world fact with a description desc, a set of attributes ATT = \{a_1, a_2, ..., a_z\} that hold its properties, and a (possibly empty) supertype aspect asp_supertype.

An aspect type and its attributes act as a metadata definition for an aspect. As a consequence, an aspect is always related to at least one aspect type and its attributes. For example, given an aspect type weather condition, some of its attributes could be temperature, wind speed and climate. In the following, we define an aspect.

**Definition 2** An aspect asp = (desc, SAT) is a relevant real-world fact, where desc is the aspect description and SAT = \{at_1, at_2, ..., at_x\} is a non-empty set of aspect types, with their attributes and respective values, which the aspect may hold; here at_i = (asp_type_k, ATV_k), at_i ∈ SAT, is a tuple with an aspect type asp_type_k and a non-empty set ATV_k = \{a_1: v_1, a_2: v_2, ..., a_n: v_n\} of attribute-value pairs such that each pair (a_i: v_i) ∈ ATV_k is an instantiation of a property a_i of asp_type_k with an (atomic or multi-valued) value v_i.

An aspect definition supports numbers, ranges, text, geometries (e.g., when an aspect describes the shape of a hurricane at a specific time instant), or any type of complex object. Some examples of aspects, aspect types and their attributes are as follows. An aspect type hotel may have the following attributes: geographic coordinates, address, stars, types of rooms (multi-valued), and facilities (multi-valued). An aspect related to this type could be Il Campanario Resort with the following attribute-value pairs: geographic coordinates, -27.439771, -48.500802;
address, Buzios Ave., Florianopolis; stars, S; types of rooms, {suite, junior suite}; facilities, {gym, swimming pool, restaurant, bar, beach service}. Another aspect type, mood, may have the attributes emoticon and intensity. An aspect of this type could be happy with the attribute–value pairs: emoticon, :-D; intensity, high.

One aspect may hold several meanings in the real world. For instance, an aspect São Paulo may have the meanings town, state, soccer team or the holy São Paulo. These meanings are categorized by a specific aspect type. As the same aspect may hold different meanings, if we relate the aspect directly to a trajectory or a trajectory point, we lose this meaning. In order to solve this problem, we introduce the concept of semantic meaning. Semantic meaning represents the relationship between an aspect and an aspect type, as shown in Figure 2. It provides the exact meaning of the aspect related to a trajectory or some of its points. In the following we define semantic meaning more formally, which could hold (SãoPaulo,Town) as an instance example.

**Definition 3** A semantic meaning \( sm = (asp, asptype) \) is an association between an aspect \( asp \) and an aspect type \( asptype \) that gives the context of the aspect, so that \( asptype \) belongs to the aspect types of the aspect \( asp \).

An aspect with a semantic meaning can be associated to a multiple aspect trajectory, a trajectory point, a moving object, or a relationship between moving objects in our conceptual model (see Figure 2). When an aspect varies frequently during the object movement, the aspect with its semantic meaning is associated to each trajectory point and is called a volatile aspect (VA). Examples are places visited (or stops) and heart rate. An aspect is also associated to a point when it represents a sparse and instant happening, such as a social media post or check-in. When an aspect does not change during an entire trajectory, it is called a long-term aspect (LTA) and is associated to the multiple aspect trajectory. Examples of this kind of aspect are the town in which the trajectory occurs or a person’s occupation. When an aspect holds during the entire life of an object, it is called a permanent aspect (PA) and is associated to the object and not to the trajectory. One example is a person’s birthplace. These aspect categories are directly related to query performance. Queries on VAs (i.e., queries related to trajectory points) will be more time-consuming, while LTAs and PAs will be retrieved more quickly.

Based on these foundations, we now define a multiple aspect trajectory.

**Definition 4** A multiple aspect trajectory \( mat = (P,S\_LTA,mo,desc) \) is a sequence of points \( P = (p_1, p_2, ..., p_n) \) of a moving object \( mo \), a (possibly empty) set of long-term aspects \( S\_LTA \), where \( S\_LTA = \{sm_1, sm_2, ..., sm_p\} \) is a set of semantic meanings, and a description \( desc \), with \( p_i = (x_i, y_i, t_i, S\_VA) \), \( p_i \in P \), where \( x \) and \( y \) constitute the spatial position of \( mo \) at the time instant \( t \), and \( S\_VA = \{sm_{1t}, sm_{2t}, ..., sm_{pt}\} \) is the set volatile aspects related to \( p_i \) that is, a set of (possibly empty) semantic meanings.

A multiple aspect trajectory belongs to a moving object. A moving object is any entity that moves in space and time. This object is always associated to a type, which can be a person, a drone, an animal, a car, or even a natural phenomenon, such as a hurricane. We formally define it as follows.

**Definition 5** A moving object \( mo = (mo\_type, desc, S\_PA) \) is an entity that can physically move in space and time, having a description \( desc \), a set of (possibly empty) permanent aspects \( S\_PA \), where \( S\_PA = \{sm_1, sm_2, ..., sm\} \) a set of semantic meanings, and a type \( mo\_type \) that categorizes it.

A new feature in MASTER when compared to the state-of-the-art data models for trajectories is the moving object relationship. A moving object may hold any type of relationship with other objects, and these relationships may also be characterized by different aspects such as the type of relationship (e.g., friendship, professional, family). We define a moving object relationship as follows.
Definition 6 A moving object relationship \( \text{mor} = (\text{mo}_1, \text{mo}_2, S_{\text{RA}}) \) is a relevant association between two moving objects \( \text{mo}_1 \) and \( \text{mo}_2 \) that holds a (possibly empty) set of relationship aspects \( S_{\text{RA}} \), where \( S_{\text{RA}} = \{s_{\text{m}1}, s_{\text{m}2}, \ldots, s_{\text{m}j}\} \) is a set of semantic meanings.

Finally, we model spatial features and events. A spatial feature denotes any relevant POI that is not spatially related to trajectory points, so it is not an aspect. Instead, it means any POI located in the trajectory neighborhood, such as a nearby restaurant. In Figure 1, an example of spatial feature is the church located between the POIs work and restaurant. So, when a trajectory point intersects a relevant POI, it is modeled as an aspect. Otherwise, it is a spatial feature. Spatial features are useful for answering spatial queries such as which restaurants are located at a distance less than \( \alpha \) from the trajectory of object A? and which trajectories have an envelope whose area is higher than avenue B? Similarly, an event denotes a happening that does not have a relationship with trajectories, but it is relevant for queries that investigate events in the trajectory neighborhood. An event occurs at a spatial feature and is valid for the period in which it happened.

It is worth mentioning that we do not define the concept of subtrajectory in MASTER, as other data models do, such as CONSTANT (Bogorny et al., 2014). Instead, our purpose is to model multiple aspect trajectories at the lowest granularity level for all aspects, allowing any possible generalization later during an analysis phase; that is, we consider the segmentation process as an analytical step. Another important point is that our model is expressive enough to describe both sparse and dense trajectories.

The idea behind MASTER is to represent trajectories for any type of application, not being limited to a subset of semantic information such as stops, (i.e., POIs), or information inferred from the raw trajectory such as speed, acceleration or transportation mode. The variety and amount of aspects will depend on what is important for the application. In tourism applications, for instance, one can model all aspects related to tourists such as the POIs visited, their categories, prices and reviews, the means of transportation, the social status of the tourist (which can be inferred from the types of POI visited), the weather conditions in order to find POI visits influenced by weather conditions, and the general mood or opinions of the tourist in relation to the town or local services. In a smart city application one might want to add the aspects of home and work, working hours, transportation mode, social status, weather condition, etc. In recommendation applications one might want to represent the moving objects and their physical relationships (when objects have encounters detected in GPS trajectories), or virtual relationships in social media, the opinions about things and places, etc. In a bird migration application one can model the bird trajectories with aspects such as the regions where they fly, regions where they eat, relationships with other species where they eat or rest, and the types of vegetation where they eat or rest.

It is also important to highlight that there are many ways to extend or instantiate MASTER for different application domains. One way is to associate existing URLs from LOD (as in Fileto et al., 2015), social media or Open Street Maps as instances of aspects or aspect attributes. These URLs point to previously defined entities or properties that are relevant to the application. Another way is to use existing ontologies, such as DBPedia, for either data extension or instantiation. For example, the Hotel class available at DBPedia Ontology (DBPedia, 2019) provides several properties, such as starRating and numberOfRooms. We could define, for example, a subclass of the class Attribute in MASTER (see Figure 2), called OntologyAttribute, to represent predefined and useful ontology data. This subclass could maintain the URL to the definition of these DBPedia Hotel properties. Furthermore, for ontology classes where the whole definition is relevant, we could define a subclass of MASTER AspectType called, for example, OntologyAspectType, to align to external classes we intend to reuse. We do not address these possible MASTER enhancements in detail in this article so as to maintain the focus on our conceptual data model, although these issues are highly relevant as future research regarding modeling of multiple aspect trajectories.
3.2 | The logical model

In this subsection we present the MASTER logical model for the conceptual model introduced in the previous section. As stated before, we adopt the Resource Description Framework (W3C, 2018a) as our logical data model because RDF data can be modeled as a graph, which is a flexible data structure to represent the high heterogeneity of possible aspects, as well as the great number of aspect relationships with trajectories, points and moving objects. Furthermore, in using RDF we conform to the Semantic Web standards of the WWW Consortium (W3C) for publishing and manipulating data on the Web (W3C, 2018b). Our intention is to allow publicly available trajectory data such as LOD in RDF format to be accessed through existing Semantic Web tools.

RDF is expressed by triples that define a relationship between two resources, where the resources, called subject and object, are nodes, and the relationship, called the predicate, is a directed edge from the subject to the object. For instance, we can define a predicate :has between two resources: :aspectType (subject) and :attribute (object).

Figure 3 shows the proposed logical model, where dotted arrows represent an entity–attribute relationship, continuous arrows represent relationships between entities, and ellipses represent entities or attributes. A predicate label followed by a cardinality pair denotes a multi-valued relationship. One example is a point that may be enriched with zero to several semantic meanings. An RDF triple schema in such a modeling is represented by two ellipses connected by an arrow. One example is a moving object (subject) that is the owner (predicate) of a multiple aspect trajectory (object).

The conversion of the conceptual model to a logical schema in RDF was inspired by several related approaches (Bagui & Bouressa, 2014; Choi, Moon, & Baik, 2013; Daniel, Sunye, & Cabot, 2016), which propose the following mapping rules:

- an entity is converted to a node;
- an attribute of an entity (or relationship) erm is converted to a not n, and an edge is defined from erm to n in order to connect them;
- a relationship between entities is converted to an edge that connects the entities, or an intermediate node between the entities.

![Figure 3](image-url) The logical model for multiple aspect trajectories
There is only one rule for the mapping of entities and attributes, so their conversion is straightforward. Even entities without attributes, such as MovingObjectRelationship (see Figure 2), become nodes in the logical model because they have relationships with other entities.

Different from the mapping of entities and attributes, there is no consensus on the mapping of relationships. In this case, we decided to consider conversion to an edge if the relationship has no properties related to it, and conversion to a node otherwise. In doing so, we avoid the generation of too many nodes in the RDF schema, and only the relationships that hold semantic meaning and hasValue as properties (see Figure 2) were mapped to nodes, and the second property was renamed Value for the sake of clarity. We also decide to maintain only the connections of Aspect and Attribute with the Value node to avoid a redundant edge between Aspect and Attribute.

Another logical design decision is related to the edge direction in the mapping of binary relationships. As a relationship is not directed at the conceptual level, we may think of two mapping alternatives: (1) the definition of two predicates; or (2) the definition of one predicate to represent one of the two possible relationship directions. Alternative (1) provides a better understanding of the relationship semantics in both directions, but the logical modeling becomes more complex and requires an integrity constraint control to keep cross-references consistent. On the other hand, alternative (2) may reduce readability, but provides a more simple logical modeling and avoids the need for consistency checking. We adopt alternative (2), giving preference to the relationship direction that is more likely to be traversed according to the expected workload for queries over multiple aspect trajectories, besides reducing the size of the logical schema. Nevertheless, SPARQL queries (SPARQL is the query language for RDF data) are able to traverse RDF graphs in both edge directions, not being limited to the predicate direction.

3.3 | The storage solution

In this subsection we present the storage solution adopted for maintaining data represented in the MASTER logical model. This solution is called Rendezvous (Santana & dos Santos Mello, 2017). Rendezvous is a triplestore based on NoSQL databases for querying large RDF data sets. NoSQL databases have been proposed for managing big data efficiently (Sadalage & Fowler, 2013). Therefore, as multiple aspect trajectories are highly heterogeneous and multidimensional data, NoSQL databases are a suitable storage resource for this new type of trajectory. Compared to related work, Rendezvous was chosen due to its multimodel NoSQL support for storing RDF data and its efficient processing of typical SPARQL queries.

Rendezvous manages RDF data in a distributed database architecture that is flexible enough to store RDF data on multiple NoSQL databases, with different data models, according to the application workload. Rendezvous adopts SPARQL and its extension to query geographic data called GeoSPARQL (OGC, 2018). SPARQL queries can be categorized into star, chain and complex queries. These query categories depend on the location of the variables in a query condition. The star category is characterized by queries that retrieve a node based on filters over nodes very close to it (we consider a traversal of at most two predicates). Chain queries are composed by a long chain of joins (distance of three predicates or more) between nodes. Complex queries are combinations of the two previous types with (optional) additional conditions with simple filters.

As a workload-aware triplestore, Rendezvous stores RDF triples that occur typically in star-shaped queries in NoSQL document databases. This kind of NoSQL database model is suitable for storing and retrieving complex data with all related properties, such as a trajectory and its related aspects. Indeed, RDF triples occurring typically in chain-shaped queries are stored in NoSQL graph databases, which benefit from keeping the subgraph that composes this chain besides the high performance for graph traversal operations.

On considering multiple aspect trajectories, Rendezvous is able to store the moving object, the trajectory and the aspect data together in one or more NoSQL databases if they are frequently filtered and/or retrieved in SPARQL queries. For example, Figure 4a shows part of an RDF graph that maintains data about a person trajectory (MAT) and its related semantic meanings (the means of transportation, city of location and person mood aspects). This RDF graph is in accordance with our logical model defined in the previous subsection, and it has a star shape.
whose center is $MAT_x$. Supposing that this shape is commonly requested in queries that intend to retrieve city trajectories made by bus where the person is happy, this star-shaped RDF fragment is stored in a NoSQL document as a typical JSON document (see Figure 4b). The document stores together the trajectory data as well as the data relevant to the semantic meanings, providing a fast retrieval for this kind of query as only one document must be accessed.

Another example is shown in Figure 5a. It presents a chain-shaped RDF fragment denoting that John has a trajectory that passed by the Eiffel tower. Supposing again that this shape is frequently requested in queries that need to retrieve people who visit this tourist point, this chain-shaped RDF fragment is stored in a NoSQL graph document as a summarized graph composed of the first and last RDF fragment nodes and an edge with a property that holds the ordered list of covered nodes and edges in the chain (see Figure 5b). This summarized stored graph also provides fast data access and retrieval for this kind of query because it maintains fewer nodes and edges than the original RDF chain (only two nodes and an edge are stored), where the edge property keeps all the chain instead of several (and additional) nodes and edges in the database. The idea is to store in this way all RDF chains that are typically requested by SPARQL queries over multiple aspect trajectory data, saving database space and retrieval time for long chains requested by frequent queries.

For both previous examples, Rendezvous manages in-memory indexes that maintain the identification of the nodes present in star-shaped RDF fragments, as well as the identification of the nodes and edges occurring in chain-shaped RDF fragments. These indexes, as well as indexes that indicate the RDF partitions in one or more NoSQL documents or NoSQL graph databases where these RDF fragments are stored, allow an efficient processing of the queries that compose the typical application workload.

**FIGURE 4** A star-shaped RDF instance (a) and its storage in a NoSQL document database (b)

**FIGURE 5** A chain-shaped RDF instance (a) and its storage in a NoSQL graph database (b)
Having detailed our strategy for storing trajectories modeled according to the MASTER data model, we next present a MASTER evaluation.

4 | A MASTER EVALUATION

The aim of MASTER is to formulate an expressive data model for multiple aspect trajectories, as well as an efficient solution for storing and querying multiple aspect trajectories. With this objective in mind, we evaluate MASTER from two perspectives: a qualitative analysis at the conceptual level in which we model a tourism application, similarly to the approach used in Bogorny et al. (2014) to evaluate the CONSTANT model, as well as an evaluation at the logical level to attest that an RDF-based storage strategy is suitable for answering the main types of multiple aspect trajectory queries; and a quantitative evaluation at the storage level in which we compare the query running time performance of our storage solution with a baseline. With this experiment we show the feasibility of MASTER as a data model that can be efficiently stored and accessed.

4.1 | Qualitative evaluation

In this subsection we model a tourism application with MASTER and present the main categories of queries related to multiple aspect trajectories.

4.1.1 | Instantiating MASTER in a tourism application scenario

In order to show how to use MASTER to model a tourism application, let us assume that tourists track their visiting experience in the city of Paris with smartphones, and that these data are later enriched with different information collected from many data sources, including social media interactions such as Foursquare, Facebook, Flickr and Twitter. These trajectories have details about the POIs visited (obtained from Foursquare check-ins as the hotel name, star rating and price; restaurant name, price, and whether it accepts credit cards; museum name and type; etc.), photos and comments posted on Flickr and Facebook, and the messages posted on Twitter by the tourists.

Figure 6 shows an example of part of a trajectory generated by John Smith, who starts his tour in Paris at Ritz Paris hotel. From the hotel he goes to the Louvre museum where he tweets and posts a photo on Flickr, just before entering the museum, where he turns his smartphone off. After leaving the museum, he takes a picture of the Arc de Triomphe du Carrousel, which he posts on Facebook, and crosses the Jardin des Tuileries to go to the Le Baudelaire restaurant, where he has lunch.

Figure 7 shows an example of the MASTER instantiation in RDF, presenting a very small part of the trajectories of John and his sister Mary. We consider two trajectories just as an example, because, unlike other state-of-the-art applications, MASTER supports the definition of relationships between moving objects.

In this figure, colors highlight PAs (gray nodes), LTAs (pink nodes) and VAs (blue nodes), respectively. This instantiation shows trajectories of two moving objects, (John and Mary), of type person, which have a relationship that is enriched with an aspect brother of type kinship. This is represented by the semantic meaning SM4. Mary has a multiple aspect trajectory MAT1 with two volatile aspect types connected to its points p11 and p12 (Mood and Transportation). They denote that part of Mary’s movement was performed by train and that she was happy at point p11. MAT1 also holds a long-term aspect type occupation, which means that Mary was retired for the entire trajectory.

On the other hand, John was born in Florianopolis (SM5) and his gender is male (SM6). His trajectory MAT2 has two VAs: SM7 indicating that at points p121 and p122 he was at the Ritz Paris hotel, which is an expensive five-star hotel, and SM3 indicates that he was happy at point p923 when he posted on Facebook a photo and the message I love Paris!. Also, it is important to notice that the aspect Florianopolis has two semantic meanings: Town
and BirthPlace. In the context of John, Florianopolis is his birthplace. This example highlights the relevance of the semantic meaning modeling in MASTER.

We observe in the example that the MASTER model allows the association of any type of information to a trajectory point, which is the aspect. This information can be a photo, a message posted on a social network such as Twitter, a place visited (POI) and its category, etc. By comparison, the CONSTANT model proposed in Bogorny et al. (2014), only allows two types of information to be associated to a point, that is, two types of aspects: environment information, such as weather conditions, and the place visited (POI). CONSTANT does not support, for instance, the representation of the aspect type Mood or Photo (associated to point p923 of John’s trajectory). Finally, CONSTANT does not support friendship relationships between moving objects, so it cannot represent the kinship relationship between John and Mary.

4.1.2 | Querying multiple aspect trajectories

We claim that three main types of queries can be posed to multiple aspect trajectories: queries that return moving objects (e.g., which are the moving objects that were born in Florianopolis and are male?); queries that return trajectories (e.g., which are the trajectories that stayed in overnight accommodation?); and queries that return aspects (e.g., which accommodation was visited in Paris by persons who were born in Florianopolis and are male?). These three queries are examples of star-shaped, chain-shaped and complex queries, respectively. Table 1 shows these queries written in SPARQL or GeoSPARQL with an arbitrary complexity depending on the number of entities that must be considered to generate the query result, which allows efficient processing by our RDF storage solution (see Section 3.3.3).

The query examples shown in Table 1 are formulated in the MASTER instantiation presented in Figure 7. In the star-shaped query in Table 1a, the star center is a moving object entity and the related filters refer to aspects...
and the moving object type entities. As stated before, this kind of query can be efficiently processed by NoSQL document databases that may maintain moving object documents where the related entities are the document attributes that can be filtered. This query returns John.

In the chain-shaped query in Table 1b, the chain is defined by a traversal that starts at a trajectory node and follows through a point of it that is enriched with a semantic meaning whose aspect type is, in turn, a subtype of an Accommodation aspect type node. This kind of query can be efficiently processed by NoSQL graph databases through a depth-first traversal. This query returns the trajectory MAT2.

In the complex query in Table 1c, the star component is represented by the filters over a moving object node mo?, and the chain component is represented by the traversal from an aspect node a? to a moving object node mo?. Additionally, we have a simple GeoSPARQL filter (geo:sfIntersects) that limits the search to trajectories that visited the city of Paris. This kind of query can be processed through a join of data retrieved from NoSQL graph and document databases. It returns Ritz Paris if this aspect belongs to a trajectory that visited Paris.

These examples show the feasibility of the logical model in RDF proposed by MASTER to support the main types of queries over multiple aspect trajectories by using SPARQL and an efficient triplestore such as Rendezvous.

### 4.2 Quantitative evaluation

This subsection presents a quantitative evaluation in terms of query performance, comparing the adopted MASTER storage solution (Rendezvous) with the SECONDO state-of-the-art spatio-temporal database system for
symbolic trajectories (Valdés et al., 2013), proposed by Güting et al. (2015). The intention here is to compare two databases that are able to maintain semantic trajectories as there is no other solution for multiple aspect trajectory storage and querying apart from Rendezvous.

SECONDO was initially developed for raw trajectories, and the aim of the definition of symbolic trajectory is to associate different semantic labels to trajectory segments, which could be used to represent a limited set of aspects. Formally, a symbolic trajectory $st$ is a sequence of tuples of the form $(st_{begin}, st_{end}, label)$, where $st_{begin}$ and $st_{end}$ are the points that delimit a subtrajectory $subt \in st$, and label designates a piece of semantic information (e.g., a POI, a means of transportation). This tuple denotes a maximal subtrajectory where the label holds.

A first limitation of symbolic trajectories and, as a consequence, of the SECONDO storage strategy is the representation of the semantic information as a label. Instead, we model semantic information in terms of aspects that may have an arbitrary schema defined by an aspect type and its attributes. Another limitation is that a symbolic trajectory associates only one piece of semantic information at a time for a trajectory or subtrajectory. For example, if we consider transportation and streets as semantic information, two symbolic trajectories must be created, one for the transportation and another one for the streets, each one represented as a relational table indexed by time. For a query that involves multiple aspects in SECONDO, all time intervals must be searched to check where all aspects hold together, while Rendezvous is able to directly access any aspect node stored in an RDF graph. These limitations of the SECONDO data model generated, as expected, worse performance results than Rendezvous for querying a large volume of trajectory data enriched with semantic information, as detailed below.

In order to compare the query performance and scalability of Rendezvous against SECONDO, we consider a data set called BerlinMOD (http://dna.fernuini-hagen.de/secondo/BerlinMOD/BerlinMOD.html). BerlinMOD is a benchmark for spatio-temporal relational database management systems, created by the SECONDO team, which is able to generate semantic trajectories of vehicles. The benchmark provides more than 25 types of spatial and/or temporal and/or semantic queries. Some examples of BerlinMOD query types are as follows. (a) What are the

| Table 1 | Typical SPARQL queries over multiple aspect trajectories |
|---------|-----------------------------------------------------------|
| SELECT mo? WHERE { mo? is-a ‘Person’ } | SELECT t? WHERE { t? composedOf p? } | SELECT a? WHERE { a? rdfs:type 'Aspect' } |
| mo? enrichedWith sm? | p? enrichedWith sm? | a? is-a at? |
| sm? aspect ‘Florianopolis’ | sm? aspectType at? | at? subtypeOf ‘Accommodation’ |
| sm? aspectType ‘birthplace’ at? subtypeOf | sm? aspectType a? |
| mo? enrichedWith sm2? ‘Accommodation’) | p? enrichedWith sm? |
| sm2? aspect ‘Male’ | FILTER( sm2? aspectType ‘Gender’) |
| | t? composedOf p? |
| | mo? owner t? |
| | mo? is-a ‘Person’ |
| | mo? enrichedWith sm? |
| | sm? aspect ‘Florianopolis’ |
| | sm? aspectType ‘birthplace’ |
| | mo? enrichedWith sm2? |
| | sm2? aspect ‘Male’ |
| | sm2? aspectType ‘Gender’) |
| (a) Star | (b) Chain | (c) Complex |
pairs of vehicles of type “truck” whose trajectories have ever been as close as “10 m” or less to each other? (a spatio-semantic query); and (b) What vehicle of type “passenger” has a trajectory that reached a point \((X, Y)\) before all the trajectories of vehicles of the same type during the time interval \([t_s, t_e]\)? (a spatio-temporal semantic query). Using the BerlinMOD generator, we created a 19.45 GB data set with around 53 million trajectories.

We ran experiments considering both Rendezvous and SECONDO as distributed infrastructures in the cloud in order to obtain better performance for large data volumes. Rendezvous uses MongoDB 3.4.3 and Neo4J 3.2.5 as the document and graph NoSQL databases, respectively, given their maturity as representatives of these families of NoSQL data models. All the distributed data nodes are Amazon m3.xlarge spot instances (https://aws.amazon.com/ec2/instance-types/) with 7.5 GB of memory and 1x32 SSD capacity. For all the experiments, we define nodes that represent MongoDB + Neo4J servers, and the Rendezvous servers were also installed on each node. In order to provide an equivalent test environment, we installed SECONDO, following its tutorial (http://dna.ferunci-hagen.de/secondo/DSECONDO/DSECONDO/Website/index.html), in a cluster with the same size as the Rendezvous installation (the same number of nodes of Amazon m3.xlarge spot instances). In both cases, all the queries were issued from a server in the same network, so the latency between the client and the clusters was inexpressive.

We divided the experiments into four rounds. We first inserted 10% of the data set and ran all the queries in order to initiate the workload awareness (we call it Rendezvous warm-up). Then, with an existing workload awareness, we ran all the queries four more times over increasing fractions of the data set (40%, 70% and 100%) and ascertained the average processing times. The results are shown in Figure 8.

As can be seen, we obtained an average query processing time that outperforms SECONDO from round 2 onwards. We obtain worse performance only in round 1 because Rendezvous is not aware of the typical query workload at this time and spends extra time analyzing the query shapes as well as storing and indexing trajectory data usually accessed by these queries in an efficient way for further retrieval. Once aware of the typical workload, Rendezvous ran twice as fast as SECONDO on average.

This first set of experiments was carried out over four Amazon nodes. In order to evaluate if the performance is positively affected by an increase in the number of nodes, we duplicated them (eight nodes), maintaining the same setup (memory, storage and servers) on each. Then we again executed all the queries four times and ascertained the average processing time on each round. The results are shown in Figure 9.

It is evident that our storage solution takes more advantage of the increase in the distributed infrastructure than SECONDO. Both approaches spent less time processing the queries, but the processing time for Rendezvous is faster and does not increase so much with the increase of the data set as compared to SECONDO. This is justified mainly by the efficient query execution strategy followed by Rendezvous, which benefits from the increase
in the number of nodes to parallelize the processing of the parts of a query. The efficient storage into NoSQL databases and workload-aware indexing scheme also contribute to this better performance.

This quantitative evaluation shows that our solution for storing and querying trajectories with multiple semantic aspects is promising since it demonstrates good performance for processing queries over increasing fractions of the data set, and better scalability compared to the baseline. In fact, Rendezvous is suitable for applications whose data grow exponentially, which may be the case for applications that deal with multiple aspect trajectories, since the addition of new machines to process an increasing number of trajectories improves its query processing time performance.

5 | CONCLUSIONS

This article presents an innovative perspective on movement data, introducing the concept of multiple aspect trajectories. An aspect is any kind of semantic information related to the movement and should be managed jointly with the spatio-temporal facet. We distinguish volatile, long-term and permanent aspects, as well as the semantic meaning of an aspect. We provide a complete solution, called MASTER, for modeling these new types of rich data, from the conceptual model to data storage and querying. The conceptual model combines intuitive formulation with very robust expressiveness of different types of trajectory aspects as compared to related work. A particular characteristic of this model is that it supports trajectories with any type of aspect annotation, from sets of attributes characterizing a POI to complex texts from social media posts, to name a few. Moreover, we show that the representation of multiple aspect trajectories is feasible in the RDF format, which can be efficiently processed by multi-model NoSQL databases. As the model represents fine-grained trajectories (i.e., trajectories at a very low granularity level), queries may roll up and down over multiple aspect trajectories. We performed a two-fold evaluation of MASTER: a qualitative one, with a tourism application domain, and a quantitative one, in which MASTER outperformed a state-of-the-art competitor in terms of efficiency of query processing.

Future work is intended include a performance evaluation over larger data sets of enriched trajectories, as well as the evaluation of other big data storage technologies, such as NewSQL databases, for maintaining multiple aspect trajectories. We also intend to extend MASTER to model data analytics information over multiple aspect trajectories, considering, for instance, dependencies among aspects. Additionally, as stated in Section 3.2, MASTER extensions to support alignment with existing ontologies or other semantic data sources (e.g., LOD) is a highly relevant research issue.
Although outside the scope of this article, it is also very important to consider privacy issues that these kinds of enriched trajectories might pose. When combining the different semantic aspects to the location information, breaches of privacy might occur. It is therefore crucial to develop methods to guarantee privacy when multiple aspects are involved.

ACKNOWLEDGMENTS

This work has been partially supported by the Brazilian agencies CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Finance Code 001), CNPQ (Conselho Nacional de Desenvolvimento Científico e Tecnológico), FAPESC (Fundação de Amparo a Pesquisa e Inovação do Estado de Santa Catarina - Project Match - Co-financing of H2020 Projects - Grant 2018TR 1266), and the European Union's Horizon 2020 research and innovation programme under GA N. 777695 (MASTER). The views and opinions expressed in this website are the sole responsibility of the author and do not necessarily reflect the views of the European Commission.

NOTE

1 We define a label hasValue for this predicate to identify it as an entity–attribute or relationship–attribute connection.

ORCID

Ronaldo dos Santos Mello https://orcid.org/0000-0003-4262-5474

REFERENCES

Alvares, L. O., Bogorny, V., Kuijpers, B., de Macedo, J. A. F., Moelans, B., & Vaisman, A. (2007). A model for enriching trajectories with semantic geographical information. In Proceedings of the 15th ACM International Symposium on Geographic Information Systems (pp. 162–169). Seattle, WA: ACM.

Bagui, S., & Bouressa, J. (2014). Mapping RDF and RDF-Schema to the entity relationship model. Journal of Emerging Trends in Computing & Information Sciences, 5(12), 953–961.

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web. Scientific American, 284(5), 28–37.

Bogorny, V., Avancini, H., de Paula, B. C., Kuplich, C. R., & Alvares, L. O. (2011). Weka-STPM: A software architecture and prototype for semantic trajectory data mining and visualization. Transactions in GIS, 15(2), 227–248.

Bogorny, V., Renso, C., de Aquino, A. R., de Lucca Siqueira, F., & Alvares, L. O. (2014). Constant: A conceptual data model for semantic trajectories of moving objects. Transactions in GIS, 18(1), 66–88.

Choi, M., Moon, C., & Baik, D. (2013). Transformation of a relational database to RDF/RDFS with ER2iDM. IEICE Transactions, E96.D(7), 1478–1488.

Daniel, G., Sunye, G., & Cabot, J. (2016). UMLtoGraphDB: Mapping conceptual schemas to graph databases. In I. Comyn-Wattiau, K. Tanaka, I. Y. Song, S. Yamamoto, & M. Saeki (Eds.), Conceptual modeling: ER 2016 (Lecture Notes in Computer Science, Vol. 9974, pp. 430–444). Cham, Switzerland: Springer.

DBPedia. (2019). DBPedia Ontology—Hotel class. Retrieved from http://dbpedia.org/page/Hotel

Ferrero, C. A., Alvares, L. O., & Bogorny, V. (2016). Multiple aspect trajectory data analysis: Research challenges and opportunities. In C. E. C. Campelo & L. M. Namikawa (Eds.), XVII Brazilian Symposium on Geoinformatics - GeoInfo 2016 (pp. 56–67). MCTIC/INPE.

Fileto, R., May, C., Renso, C., Pelekis, N., Klein, D., & Theodoridis, Y. (2015). The Baquara 2 knowledge-based framework for semantic enrichment and analysis of movement data. Data & Knowledge Engineering, 98, 104–122.

Güting, R. H., de Almeida, V. T., Ansorge, D., Behr, T., Ding, Z., Höse, T., ... Telle, U. (2005). SECONDO: An extensible DBMS platform for research prototyping and teaching. In Proceedings of the 21st International Conference on Data Engineering (pp. 1115–1116). Tokyo, Japan: IEEE.

Güting, R. H., Valdés, F., & Damiani, M. L. (2015). Symbolic trajectories. ACM Transactions on Spatial Algorithms & Systems, 1(2), 7.
Hornsby, K. S., & Cole, S. J. (2007). Modeling moving geospatial objects from an event-based perspective. *Transactions in GIS, 11*(4), 555–573.

Noël, D., Villanova-Oliver, M., Gensel, J., & Quéau, P. L. (2015). Modeling semantic trajectories including multiple viewpoints and explanatory factors: Application to life trajectories. In *Proceedings of the First International ACM SIGSPATIAL Workshop on Smart Cities and Urban Analytics* (pp. 107–113). Bellevue, WA: ACM.

Nogueira, T. P., Braga, R. B., de Oliveira, C. T., & Martin, H. (2018). FrameSTEP: A framework for annotating semantic trajectories based on episodes. *Expert Systems with Applications, 92*, 533–545.

OGC. (2018). GeoSPARQL—A geographic query language for RDF data. Retrieved from https://www.opengeospatial.org/standards/geosparql

Pan, J. Z. (2009). Resource description framework. In S. Staab & R. Studer (Eds.), *Handbook on ontologies* (pp. 71–90). Berlin, Germany: Springer.

Parent, C., Spaccapietra, S., Renso, C., Andrienko, G. L., Andrienko, N. V., Bogorny, V., ... Yan, Z. (2013). Semantic trajectories modeling and analysis. *ACM Computing Surveys, 45*(4), 42:1–42:32.

Pelekis, N., Theodoridis, Y., Vosinakis, S., & Panayiotopoulos, T. (2006). Hermes: A framework for location-based data management. In Y. E. Ioannidis, M. H. Scholl, J. W. Schmidt, F. Matthes, M. Hatzopoulos, K. Bohm, ... C. Böhm (Eds.), *Advances in Database Technology: EDBT 2006, 10th International Conference on Extending Database Technology, Munich, Germany, March 26–31, 2006, Proceedings* (Lecture Notes in Computer Science Vol. 3896, pp. 1130–1134). Berlin, Germany: Springer.

Ruback, L., Casanova, M. A., Raffaetà, A., Renso, C., & Vidal, V. M. P. (2016). Enriching mobility data with linked open data. In *Proceedings of the 20th International Database Engineering & Applications Symposium* (pp. 173–182). Montreal, Canada: ACM.

Sadalage, P. J., & Fowler, M. (2013). *NoSQL distilled: A brief guide to the emerging world of polyglot persistence*. Boston: Addison-Wesley.

Santana, L. H. Z., & dos Santos Mello, R. (2017). Workload-aware RDF partitioning and SPARQL query caching for massive RDF graphs stored in NoSQL databases. In *Proceedings of the 32nd Brazilian Symposium on Databases* (pp. 184–195). Spaccapietra, S., Parent, C., Damiani, M. L., de Macedo, J. A., Porto, F., & Vangenot, C. (2008). A conceptual view on trajectories. *Data & Knowledge Engineering, 65*(1), 126–146.

Valdés, F., Damiani, M. L., & Güting, R. H. (2013). Symbolic trajectories in SECONDO: Pattern matching and rewriting. In W. Meng, L. Feng, S. Bressan, W. Winiwarter, & W. Song (Eds.), *Database systems for advanced applications* (pp. 450–453). Berlin, Germany: Springer.

W3C. (2018a). RDF—Semantic web standards. Retrieved from https://www.w3.org/RDF/

W3C. (2018b). Semantic Web—W3C. Retrieved from https://www.w3.org/standards/semanticweb/

Yan, Z., Chakraborty, D., Parent, C., Spaccapietra, S., & Aberer, K. (2011). Semitri: A framework for semantic annotation of heterogeneous trajectories. In *Proceedings of the 14th International Conference on Extending Database Technology* (pp. 259–270). Uppsala, Sweden: ACM.

Yan, Z., Macedo, J., Parent, C., & Spaccapietra, S. (2008). Trajectory ontologies and queries. *Transactions in GIS, 12*(Suppl 1), 75–91.

**How to cite this article**: dos Santos Mello R, Bogorny V, Alvares LO, et al. MASTER: A multiple aspect view on trajectories. *Transactions in GIS, 2019;23:805–822*. [https://doi.org/10.1111/tgis.12526](https://doi.org/10.1111/tgis.12526)