Chapter 3
Uncertainty in Spatial Trajectories

Goce Trajcevski

Abstract This chapter presents a systematic overview of the various issues and solutions related to the notion of *uncertainty* in the settings of moving objects trajectories. The sources of uncertainty in this context are plentiful: from the mere fact that the positioning devices are inherently imprecise, to the pragmatic aspect that, although the objects are moving continuously, location-based servers can only be updated in discrete times. Hence come the problems related to modelling and representing the uncertainty in Moving Objects Databases (MOD) and, as a consequence, problems of efficient algorithms for processing various spatio-temporal queries of interest. Given the ever-presence of uncertainty since the dawn of philosophy through modern day nano-level science, after a brief introduction, we present a historic overview of the role of uncertainty in parts of the evolution of the human thought in general, and Computer Science (CS) and databases in particular, which are relevant to this chapter. The focus of this chapter, however, will be on the impact that capturing the uncertainty in the syntax of the popular spatio-temporal queries has on their semantics and processing algorithms. We also consider the impact of different models in different settings – e.g., free motion; road-network constrained motion – and discuss the main issues related to exploiting such semantic dimension(s) for efficient query processing.

3.1 Introduction

Historically, the impact of the imperfect knowledge on the reasoning and belief has been a topic that has attracted a lot of research interest among both philosophers and logicians [41, 63, 44, 123]. With the advent of the computing technologies, as various domains of Computer Science (CS) have emerged, the importance of

---

Goce Trajcevski
Dept. of EECS, Northwestern University, 2145 Sheridan Rd. Evanston, Il 60208 USA (research supported by NSF-CNS-0910952), e-mail: goce@eecs.northwestern.edu

capturing the uncertain/probabilistic nature of the data has been recognized in many of them:

- Artificial Intelligence which, in a sense popularized the Possible-Worlds semantics [6, 40, 158].
- Knowledge Representation and Reasoning along with Logic Programming and Deductive Databases [1, 7, 98, 173].
- Incorporating it on top of the traditional database technology [16].

to list but a few.

Due to the novel application domains along with advancements in database technology, a lot of recent research has been undertaken, addressing problems in modelling and efficient querying of imprecise/uncertain data [2, 20, 111, 125, 133, 134].

In the past two decades, the advances in sensing and communication/networking technologies, along with the miniaturizations of computing devices and development of variety of embedded systems have spurred the recognition of the importance of Location Based Services (LBS) [126] in a plethora of applications. From military, through structural and environmental monitoring, disaster/rescue management and remediation, to tourist information-providing systems – the efficient management of large amount of *(location, time)* data pertaining to mobile entities over (large) periods of time is a paramount. After several works and development of some ad-hoc solutions [89], the field of Moving Objects Databases (MOD) [56, 163] emerged in the late 1990's as an enabling technology for the LBS-related applications, providing formal foundations and bringing about development of prototype systems [51, 69].

Contrary to the typical assumptions in:

1. Spatial databases [19, 76, 130, 142], where the data items may have dimensionality and extent, but are (relatively) static over time;
2. Temporal databases [35, 72, 132], where the main objective is capturing the time-varying nature of the data in various application domains; and
3. Time-series [79, 78, 116, 177], where the values of the data samples over time often pertain to a single dimension,

in MOD-settings, the objects are assumed to move, either freely in the 2D (or even 3D) space [90, 52, 131], or constrained by a road network [38, 50, 28, 154]. The main features of spatio-temporal data sets:

1. The discrete data samples are expected to represent a continuous motion over the given space, thereby necessitating some type of an interpolation; and
2. The typical queries of interest (e.g., whereabouts-in-time, range, (k)nearest-neighbor, reverse nearest-neighbor, skyline) are *continuous* – which is, their answers need to be re-evaluated in time, or even *persistent* (cf. [131]) – which is, in addition to re-evaluating the answers over time, one may need to take into consideration the entire history of the motion;

have influenced a large body of works addressing issues related to modelling/ representation, indexing and querying such data [9, 14, 24, 80, 37, 99, 109, 59, 112, 71, 93, 60, 97, 105, 164, 102, 119, 106, 140, 139, 141, 155, 169, 172]

In practice, the location data at different time-instants is obtained by some positioning devices like, for example, a GPS-enabled device on-board a moving objects, which eventually transmits the location to a MOD server(s). However, GPS receivers only approximate the actual position of the respective sensor or object due to physical limitations and measurement errors of the sensing hardware [27].

The position data may be obtained by some other (collaborating) tracking-devices e.g., roadside sensors [26]. Even more so in application settings like intrusion-detection and environmental monitoring, where the tracking of the moving object of interest is based on collaborative trilateration and continuous hand-off among the participating sensors [39, 137, 62, 110, 136, 176, 179, 180]. In addition to the interpolation in-between consecutive location samplings, the imprecision of the devices involved, both GPS-based as well as sensor-based, is yet another source of uncertainty. But another example of location imprecision is the investigation of trajectories of various particles in physical and chemical processes [121].

Motivated by these observations, many researchers have focused on addressing the problem of uncertainty management in MOD settings [5, 21, 31, 30, 67, 66, 83, 85, 86, 91, 94, 108, 113, 115, 114, 143, 149, 145, 159, 160], also considering the impact of the restriction of the motions to road networks [5, 31, 30, 45, 84, 83, 178]. While it is often the case to assume that the possible locations of a given object at a particular time-instant is obeying a uniform distribution within certain bounds, recent works have addressed the impact of the different *pdf*s [21, 67, 159].

An important observation when it comes to incorporating uncertainty into the processing of spatio-temporal queries is that, in order to relieve the user from factoring it out from the answers, it needs to be incorporated in the very syntax of the given query [18, 88, 103, 115, 149, 145, 148, 171]. Although many of the existing works have focused on uncertain point-objects with (mostly) linear motion with constant speed in-between updates, some recent results have addressed the uncertainty aspects of points/lines with extent [157], and even uncertain fields [43, 170]. Such models are necessary to capture, for example, the trajectory of the "eye" of a given hurricane [152] – however, in addition to the "eye" being uncertain, the (moving) spatial zone affected by that hurricane is also uncertain.

This chapter gives an overview of the research results in the field of managing and querying uncertain trajectories data, in a manner that will strike a balance between the breadth and the depth of the different topics presented in the existing literature. The intended goal is to present a body of materials in a manner that will be suitable for both non-specialists to get introduced to the field, and specialists to get a coherent presentation that could help influence the selection of research directions.

In the rest of this chapter, in section 3.2.3, we will overview the historic evolution of the incorporation and treatment of uncertainty in the philosophy and logic[1], as well as certain CS-areas related to AI and databases, along with time-geography and geometry. Subsequently, Section 3.3.2 will address the role and impact of uncertainty in spatial databases and temporal databases, paving a way for the crux of the material of this chapter which will follow in Sections 3.4 and 3.5. After a

[1] It is well beyond the scope of this chapter to discuss the importance and the treatment of uncertainty in all the different scientific fields like, for instance, physics, chemistry, etc.

brief formal overview of the notion of trajectories and spatio-temporal queries[2], we will focus on a thorough analysis of the issues related to incorporating the uncertainty into the trajectories' data model, queries-syntax, and the corresponding processing algorithms. Specifically, Section 3.4 will present uncertainty models for free-space motion as well as models of uncertainty for motion restricted to road network. Subsequently, in Section 3.5, we consider the issues that uncertainty brings in the query processing, and we present examples of different (type_of _query, uncertainty_model) couplings. Section 3.6 concludes this chapter and gives a brief overview of the role of trajectories uncertainty in a broader context and application settings.

3.2 Uncertainty Throughout the History

We now discuss the evolution of the treatment of the uncertainty along different aspects of the evolution of the human thought. Firstly, we will review the uncertainty of the knowledge/belief and how philosophers and logicians throughout the history have addressed its formalization(s). Subsequently, we follow up with discussing the role and treatment of the uncertainty in the fields of Artificial Intelligence and Databases. The last part of the section touches upon the fields of time-geography and inexact geometries.

3.2.1 Philosophy and Logic

As part of the philosophy focusing on principles of valid reasoning, inference and demonstration, *logic* has had its presence in many of the ancient civilizations – Babylon, Egypt, China and India – clearly demonstrated in some inference rules related to geometric and astronomical calculations. However, the philosophical form of logic that is likely the most influential one for the Western and Islamic cultures, bringing about a symbolic and purely formal axiomatic treatment – is the one developed in ancient Greece, as first formalized by Aristotle. Even the earliest works, however, observed that some aspects of formalizing the thought required the concepts of *knowledge* and/or *belief*, leading to the so called *epistemic logic* [13] which focuses on their systematic properties. The syntax of epistemic logic extends the propositional logic with the unary operator K_a (or B_a) applied to the traditional propositions. Thus, B_aP denotes *"the agent a Believes P"*, where P is any proposition, and its meaning/semantics is: *in all possible worlds compatible with what the agent a believes, it is the case that P* [64].

Although the epistemic logic has found numerous applications in fields like CS (AI, Databases) and economics, it was the *modal logic* [17, 42] that provided a

[2] Addressed in greater detail in Chapter 2.

perspective for incorporating the uncertainty into a systematic framework, enabling a qualification of the truth/falsity. Intuitively, a modality is any word or phrase that can be applied to a given statement S which, in a sense, creates a new statement that makes an assertion *about the mode of truth of S* – when, where or how S is true (or about the circumstances under which S may be true) [42]. For a given proposition P, the two main operators of the modal logic are:

1. $\Box$: denoting *necessity* – e.g., $\Box P \equiv$ *it is necessary that P*.
2. $\Diamond$: denoting *possibility* – e.g., $\Diamond P \equiv$ *it is possible that P*.

Several approaches have been undertaken towards axiomatization of the modal logic, however, it was not until the work of Kripke [81] that the semantics and model-theoretic aspects were fully considered. Given that a particular semantics is as good as its entailment relation (within a given model) can "mimick" the consequence relation in terms of syntactic derivability, the main novelty of the, so called, Kripke structures (or frames) is that they provided a foundation for connecting a particular modal logic to a corresponding class of frames, thereby enabling reasoning about its (in)completeness. As a specific example, instead of evaluating a composite formula based solely on the *true/false* value of its primitive constituents, one may consider the behavior of the composite formula when the truth values of the constituents are *changing gradually* from false to true according to some "scenario" (cf. [29]). A thorough treatment of the topic is well beyond the scope of this chapter – however, the main influence of this line of works is that they brought *in agnitio* one concept that has been widely used since – the *possible worlds*[3]

3.2.2 Uncertainty in AI and Databases

Due to its close relationship with logic and, for that matter, extensive use of the Logic Programming paradigm, AI is one of the very first CS fields that have adopted the concept of possible worlds. The Possible Worlds Approach (PWA) is a powerful mechanism for incorporating new information into logical theories, studied by philosophers interested in belief revision and scientific theory formation [3], as well as database theorists [1, 36]. The basic premise of PWA is to keep a single model of the world that is updated when actions are performed. The update procedure involves constructing the nearest world to the current one, in which the consequences of the actions under consideration hold. As explained in [158], the PWA-based revision of a theory can be summed up as: To incorporate a set **S** of formulae into an existing theory **T**, take the maximal subset **T** of **T** that is *consistent with* **S**, and add **S** to **T**. This is one of the approaches undertaken for the problem of minimality of view updates in databases [46].

Although it aimed at bringing about computationally efficient procedures for reasoning about actions, PWA was shown to have problems when it comes to, so called,

[3] We respectfully note that philosophers and logicians are likely to disagree that semantics based on Kripke frames are model-equivalent to the one based possible worlds.

frame, ramification, or qualification issues, due to the fact that it did not distinguish between the *state of the world* and the *description of the state of the world* [158]. As a remedy, the Possible Models Approach (PMA) was introduced, which observed the *models* of a given theory **T**, rather than its formulae. The goal of PMA is to change as little as possible the models of **T** in order to make the new set of formulae **S** true. Once the focus has shifted on the models, reasoning about actions became more amenable to incomplete information.

The archetypical example of uncertainty in traditional relational databases was the one of an absence of value for a particular attribute, denoted as NULL. This value is not associated with a particular type and, more importantly, it implies involvement of the three-valued logic, adding the *unknown* value in the picture and disturbing the "cushy" Closed World Assumption (CWA) model of relational databases. With NULL value, one is no longer justified to assume that the values stored within a database correspond to a complete version of the world and everything not stored in the database is false – thereby imposing the Open World Assumption and demanding extra caution when using SQL in practice.

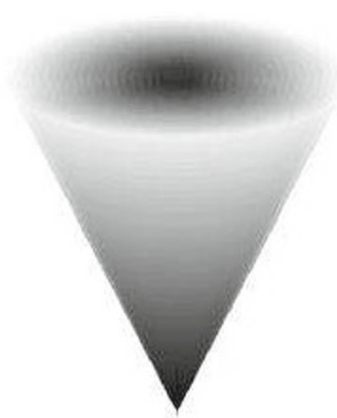
A plethora of novel application domains such as Location-Based services, health and environmental monitoring based on sensor data analysis, biological image analysis, market analysis and economics – generate a vast amount of data which is inherently uncertain due to the imprecision of measuring devices, randomness and delays in data updates. This has spurred a tremendous research interest in probabilistic databases [2, 11, 10, 20, 92, 125, 133, 167]. In these settings, one typical feature is that some attributes are probabilistic, in the sense that their values are given by a probabilistic density function (*pdf*) – however, in practice, one cannot hope to have the *pdf* available and must rely on samples instead. In general, a probabilistic database can be thought of as finite set of probabilistic tables – one for each plausible value of the uncertain tuples, associated with membership probability. If the probability of a particular instantiation for the objects in the database is greater zero, then that particular instantiation constitutes one of the possible world. The main problem is that the cardinality of the set of all the possible worlds is exponential in the number of uncertain objects [6]. In addition to complicating the issue of the semantics to the answers of the queries, the large number of possible worlds clearly imposes computational costs in their processing – enumerating the answers in all the possible worlds is infeasible in practice. Hence, the researchers have resorted to balancing tradeoff between accuracy and computational cost, e.g., retrieving only objects with highest likelyhood to be in the result; reporting only answers the probability of which exceeds a given threshold; returning approximate answers, etc. A recent approach addressing a generic query optimization for uncertain databases, introducing a threshold operator (τ -operator) to the query plan and demonstrating that it is generally desirable to push it "down" as much as possible, is presented in [118].

Getting into a detailed discussion on the topic of probabilistic databases is beyond the scope of this Chapter, and for a comprehensive overview the reader is referred to the recent tutorials [134, 111, 120], along with a cohesive recent collection of works with an extensive list of references available in [58].

3.2.3 Time-Geography and Inexact Geometries

Time geography [61] addresses questions like:
Given a location of a mobile agent at time t_0, where is the agent at a later time $t_1 > t_0$, or where was the agent at a previous time $t_2 < t_0$? [160]. Assuming the agent can move in any direction and is limited only by a maximum speed v_{max}, time geography represents the reachable locations of this agent by a right cone in $(X, Y, Time)$-space. The cone apex represents the agents location at t_0, and the aperture represents the maximum speed of the agent – specifically, a cone base B_i represents the set of locations the agent may settle at a time $t_i > t_0$.

Fig. 3.1 Possible Whereabouts of an Agent: different shades correspond to the *pdf*'s emanating among consecutive time-instants (cf. [160], with permission)



Focusing on the discrete probabilistic space-time cone approximation of the continuous *pdf* of the location of a mobile agent, in [160] three approaches are undertaken for deriving that approximation: (1) from a random walk simulation, (2) from combinatorics, and (3) from convolution The results are targeting some basic questions of interest for time-geography like, for example, *what is the most probable arrival time of an agent A at a particular location B*? We will discuss the space-time prisms and their implications to trajectories uncertainty in more detail in Section 3.4.

While the interest of time geographers is on the uncertainty of location of mobile agents as the time evolves, a specific type of handling imprecision was considered by the GIS (Geographic Information Systems) researchers, focusing on the spatial properties of the basic primitives. Namely, in a vector GIS, the representation is based on the type of an infinitely small point, in accordance with Euclidean. However, more often than not it is the case that GIS maps are representing geographic entities that have spatial extent. In [123], an axiomatic *tolerance geometry* was developed, aiming at formalizing the limited capability of distinguishing stimuli in visual perception. Intuitively, the work "blurred" the concepts of *proximity* and *i-dentity*, and developed corresponding primitives for a formalism that can substitute the traditional concept of "between"-ness, with an ε-between-ness.

Recently, an attempt of formalizing the geometric reasoning (and computing) for uncertain object was presented in [174], offering an approach based on multiple modalities of uncertainty in position.

The Euclidian geometry is an axiomatic well-founded logical theory, and it has interpretation of its primitives satisfying its axioms. For example, the Cartesian model of Euclidean geometry provides an interpretation of the geometric primitives point, line, equality, incidence, congruency, etc..., in the real 2D plane. However, once the uncertainty is allowed for the basic primitives, the logical foundations need to be revisited and novel derivation rules (for developing theorems based on valid proofs) are needed. A very recent formalization of the uncertainty into the foundation of the Euclidian geometry – the first postulate – and giving interpretations that capture the GIS-intuition of *points with extension* and *lines with extension* is presented in [157]. The work extends the Boolean-based reasoning by translating it into fuzzy logic, providing means of approximating and propagating positional tolerance within sound inference system.

3.3 Uncertainty in Spatial and Temporal Databases

Two fields that have emerged in the mid/late 1980s – *Spatial Databases* and *Temporal Databases* – became, in some sense, precursors to the spatio-temporal databases. In the rest of this section, we present some issues addressed in each field, which are of relevance to the context of this Chapter.

3.3.1 Spatial Databases

Spatial databases [49, 122, 130] deal with efficient storage and retrieval of objects in space that have identity and well-defined extents, locations, as well as certain geometric and/or topological relationships among them, owing to developments in application fields (GIS, VLSI design, CAD) that needed to deal with large quantities of geometric, geographic, or spatial entities. In addition to some stable and mature prototypes prototypes based on solid algebraic type-foundation [48, 55] commercial Database Management System (DBMS) vendors have provided extensions to their products, supporting spatial types and operations (Oracle Spatial, DB2 Spatial Extender, PostgresGIS, Microsoft SQL server, MySQL). Without a doubt, the results in spatial databases have spurred several important research avenues in MOD settings, e.g.:

- Many popular types of MOD queries (e.g., range, nearest-neighbor) have variants that were studied in spatial databases context [65, 124].
- Indexing structures developed for facilitating the efficient data access for processing spatial queries [8, 57] served as foundations for spatio-temporal indexes.

- Topological properties of and relationships among spatial types [32], as well as generalization issues [156], have also found their ”counter-parts” in spatio-temporal data.

What is of a most specific interest for this Chapter, is that the various concepts of *uncertainty* that were investigated in the context of spatial databases have been, in one way or another, applied and/or adopted in the context of uncertain spatio-temporal data.

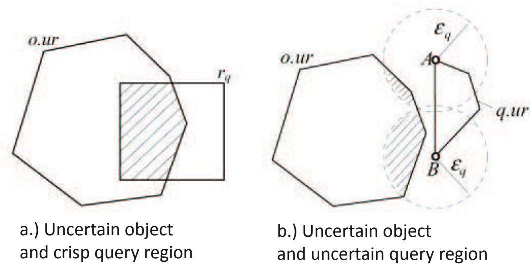


Fig. 3.2 Processing of Spatial Range Query for Uncertain Objects: a.) crisp range; b.) uncertain range (cf. [142], with permission)

The first such concept is the one of *location uncertainty*. Namely, if one cannot specifically determine the values of the coordinates of a given point in a reference coordinate system, then the specification of that point must incorporate the accompanying uncertainty. We already touched upon the issue of tolerance geometry in Section 3.2.3, which generalized the concept of a point into a point with extension and investigated the impacts on the formal reasoning in such geometries. In practice, however, in addition to capturing the uncertainty – e.g., via *pdf*, or histogram, alongside with some boundary – an important aspect is how to incorporate it in the query processing techniques. The first observation is that the answer to the query must somehow reflect it. An illustrating example is shown in Figure 3.2, pertaining to processing of spatio-temporal range queries for objects with uncertain locations. Part a.) of Figure 3.2 shows the uncertain object *o.ur* whose possible locations are bounded by a heptagonal region. For as long as the query region *r.q* is crisp, one can determine the probability that *o.ur* is inside the range *r.q* – e.g., if the *pdf* of *o.ur* is uniform, the probability is: $|o.ur \cap r.q|/|o.ur|$. However, once the query region itself is uncertain – e.g., its boundaries have some ε bound of possible whereabouts (cf. Figure 3.2.a.)), then the calculations of the probability become more computationally expensive. To cater for this, it was observed in [142] that if one is interested only in objects whose probability of being inside the range exceeds certain threshold, then a pruning could be applied, for which the U-Tree indexing structure was introduced.

Many entities such as regions of toxic spread, temperature maps, water-to-soil boundaries and boundaries among different types of soil, cannot be exactly determined. One of the approaches to address the storing and querying of such data was to introduce the concepts of *fuzzy points*, *fuzzy lines*, and *fuzzy regions* in the Euclidian space. Along with that, fuzzy spatial set operations like union, intersection, and difference, as well as fuzzy topological predicates were introduced to manage spatial joins and selections over fuzzy objects [77, 114, 127, 128].

3.3.2 Temporal Databases

Many applications of databases like, for example, accounting, portfolio management, medical-record and inventory management record information that is time-varying in nature [35, 72]. At the heart of the temporal databases is the distinction between:

- *valid time* of a fact, which is the time at which a particular data item is collected and becomes true as far as the world represented by the database is concerned – possibly spanning the past, present, and future. However, the valid time may not be known, or recording it may not be relevant for the applications supported by the database or – in the case that the database models possible worlds – it may vary across different possible worlds.
- *transaction time* of a fact, which is the time that a given fact is current in the database. Transaction time may be associated with different database entities like, e.g., objects and values that are *not* facts because they cannot be true or false in isolation. Thus, all database entities have a transaction-time aspect, which has a duration: from insertion to (logical) deletion of a given entity.

Capturing the time-varying nature using traditional data models and query languages can be a cumbersome activity and, as a consequence, constructs are needed that will enable capturing the valid and transaction times of the facts, leading to temporal relations. In addition, query languages [23, 132] are needed with syntactic extensions that enable database operations on temporal models.

As an example of uncertainty in temporal databases, consider the following scenario (cf. [12]):

Transportation companies (e.g., UPS, DHL) have massive databases containing information about the various packages they are shipping or have previously shipped and, most importantly, how long it takes for packages to get from a given origin to a given destination. In such cases, based on the existing data about the valid times, it may be the case that the database has the following information regarding the arrival time for packages departing from o_i at 10AM and arriving at d_j:

$$\{(12:30[0.4,0.6]),(12:35[0.3,0.5])\}$$

indicating that the probability of a package arriving at 12:30PM is between 0.4 and 0.6, and the probability of a package arriving at 12:35PM is between 0.3 and 0.5.

A similar scenario in the context of trajectories uncertainty occurs, for example, when the *(location,time)* data is obtained via tracking. In addition to the location imprecision, due to the clocks-skew among the sensors participating in tracking [135]. In such cases, even if a crisp location is detected after trillateration, the value of the *time* attribute will be bound to an interval instead of an instant.

3.4 Modelling Uncertain Trajectories

We now focus on the first part of the main topic of this chapter – modelling of uncertainty in spatial trajectories. After a brief overview of some basic spatio-temporal concepts and definitions, we proceed with detailed discussion of the main aspects of some of the existing models for capturing the uncertainty of spatio-temporal data. The last part of this section is dedicated to the uncertainty aspects when the motion of the objects is constrained to a road network.

As mentioned in Section 3.1, the *(location, time)* data capturing the motion of moving objects is subject to uncertainty for a variety of reasons, at every stage of its generation. The GPS receivers only approximate the actual position [27] hardware. The precision of motion sensors deteriorates with the distance from their own location and, moreover, typically the localization of a tracked object is done via trillateration, without guaranteeing that every participating node is reliable [62, 73, 175, 179]. Aside from the location determination per se' additional issues arise due to timing synchronization [135], as well as the protocols used for transmitting the *(location,time)* information from sensors to MOD or LBS servers [37, 162]. Last, but not the least – since it is impossible to record the location for every single time-instant, the interpolation in-between consecutive records yields an uncertainty of the trajectory [74, 88].

In a similar spirit to the works that have developed formal models for representing and querying ”crisp” trajectories – i.e., ones without any uncertainty of the moving objects whereabouts(e.g., [53, 154]), researchers have addressed the problem of generic representation of uncertainty, along with a framework for syntactic categorization of spatio-temporal queries [88, 103, 171].

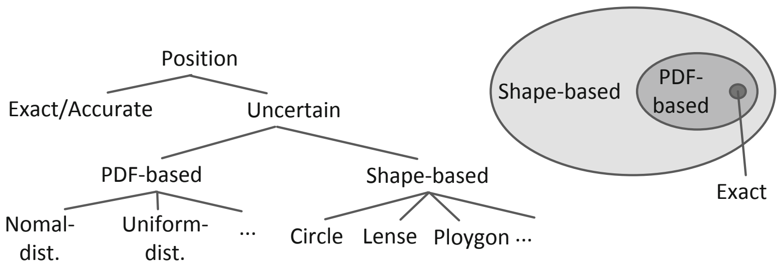


Fig. 3.3 Categorization of Uncertainty Models (cf. [88], with permission)

An example is shown in Figure 3.3 where, in addition to the exact/accurate model in which the location as a given time is assumed to be the actual one without error, two broad categories of location uncertainty models are identified [88].

1. *pdf-based models*: Motivated by the GPS-based location uncertainty, within these models the position/location at time t is described by a two-dimensional probability density function (pdf) $\ell_t : R2 \rightarrow [0, +\infty)$.

2. *shape-based models*: Bounding the possible locations by geometric shapes (e.g., circle, lens, polygon), these models may have associated probability values. However, in contrast to the pdf-based models, no claims are made about the spatial pdf within a shape.

Once a model of uncertainty is established, its impact on the syntax of the queries needs to be considered, in conjunction with the type of a particular query – e.g., position, range, nearest neighbor (cf. [88, 103]).

In practice, there is a coupling between selecting the model for the motion plan of the moving objects – affecting the choice of the representation [54, 52] of trajectories in a MOD and, consequently, the overall strategy of query processing – and the uncertainty model. One of the common definitions of a moving object trajectory is as follows:

Definition 3.1. A *trajectory* ***Tr*** is a function *Time* $\rightarrow \mathscr{R}^2$, represented as a sequence of 3D (2D spatial plus time) points, accompanied by a unique ID of the moving object:

$$\boldsymbol{Tr}_i = (oid_i,\ (x_{i_1}, y_{i_1}, t_{i_1}),\ \ldots,\ (x_{i_k}, y_{i_k}, t_{i_k})),$$

where $t_{i_1} < t_{i_2} < \ldots < t_{i_k}$.

We note that Definition 3.1 can be used to represent both *past* trajectories (i.e., ones whose motion is completed relative to the *now*-time) as well as *future* trajectories. In future-trajectories settings, users transmit to the MOD server: (1) the *beginning location*; (2) the *ending location*; (3) the *beginning time*; and (4) possibly a set of points to be visited. Based on the information available from electronic maps and traffic patterns, the MOD server will construct and transmit the *shortest travel time* or *shortest path trajectory* to the user. This model is applicable to the routing of commercial fleet vehicles (e.g., FedEx and UPS) as well as to web services for driving directions, where tens of millions of computations of shortest path trajectories are executed monthly by services such as MapQuest, Yahoo! Maps, and Google Maps [70].

Following are the two noteworthy observations regarding Definition 3.1:

O1: What is (the description of) the location of a given object at a time instant in-between two consecutive points ($t_{i_j} \leq t \leq t_{i_{j+1}}$)?

A very common assumption is that in-between two consecutive points, the objects move along straight line segments and with constant speed, calculated as:

$$v_{i_k} = \frac{\sqrt{(x_{i_k} - x_{i_{(k-1)}})^2 + (y_{i_k} - y_{i_{(k-1)}})^2}}{t_{i_k} - t_{i_{(k-1)}}} \tag{3.1}$$

Thus, the coordinates of an object oid_i at time $t \in (t_{i_{(k-1)}}, t_{i_k})$ can be obtained by linear interpolation:

$$\begin{aligned} x_i(t) &= x_{i_{(k-1)}} + v_{i_k} \cdot (t - t_{i_{(k-1)}}) \\ y_i(t) &= y_{i_{(k-1)}} + v_{i_k} \cdot (t - t_{i_{(k-1)}}) \end{aligned} \tag{3.2}$$

However, researchers have observed that the linear interpolation assumption need not be suitable for certain applications, especially if prediction of future locations is

needed. Hence, techniques have been proposed for using different (hybrid) models based on representing the objects whereabouts with other algebraic functions [74, 138].
O2: Are the points arriving at a MOD server in a batch manner, i.e., portions of, or the entire trip – as opposed to streams of individual (location,time) data values [100, 102, 39, 162]?

Catering to observations **O1** and **O2**, researchers have proposed several models of uncertainty of motion, which we address in detail next.

3.4.1 Cones, Beads and Necklaces

An idea discussed in the works of Hägerstrand in the early 1970s in time-geography [61], was the first one to have found its way into MOD research. The first consideration of the implications of the fact that the object's motion was constrained by some maximal speed v_{max} *in-between* two updates was presented in [113]. Based on its definition as a geometric set of 2D points, it was demonstrated that the objects whereabouts are bound by an ellipse, with foci at the respective point-locations of the consecutive updates, as illustrated by the spatial (X-Y) projection in Figure 3.4. Subsequently, [66], presented a spatio-temporal version of the model, naming the volume in-between two update points a *bead*[4], and the entire uncertain trajectory, a *necklace*. Note that, in a sense, a bead is a "backward-extension" of the concept of spatio-temporal cone as discussed in Section 3.2.3. Namely, the assumption is that for as long as there is no new *(location, time)* update, the object can be located anywhere inside the cone emanating from the last-known update. However, once a new update arrives, in addition to the possible-future locations, it also constraints the possible locations from the past (since the previous update). A thorough analysis of the properties of the beads was recently done in [85, 86, 108].

Definition 3.2. Let v^i_{max} denote the maximum *speed* that an object can take within the time-interval (t_i, t_{i+1}). A *bead* $B_i = ((x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1}), v^i_{max})$ is defined as the set of all the points (x, y, t) satisfying the following constraints:

$$\begin{aligned} t_i \leq t \leq t_{i+1} \\ (x - x_i)^2 + (y - y_i)^2 \leq [(t - t_i)v^i_{max}]^2 \\ (x - x_{i+1})^2 + (y - y_{i+1})^2 \leq [(t_{i+1} - t)v^i_{max}]^2 \end{aligned} \tag{3.3}$$

The first and the second constraint of Equation 3.3, when taken together describe a cone emanating upwards from (x_i, y_i, t_i), with a vertical axis and with circles whose radius value at time t is $(t - t_i)v^i_{max}$, whereas the first and the third constraint together, specify a cone emanating downwards from $(x_{i+1}, y_{i+1}, t_{i+1})$, with a vertical axis and with circles whose radius at time t is $(t_{i+1} - t)v^i_{max}$. Hence, the bead B_i can be

[4] We note that, more recently, this model is also called *space-time prism*.

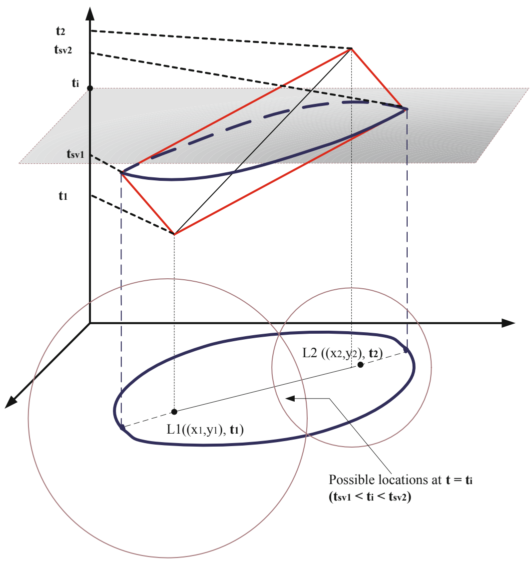


Fig. 3.4 Spatio-temporal Beads and their (X,Y) Projection

viewed as volume defined by the intersection of those two cones. We note that at $t = t_i$ (resp. $t = t_{i+1}$) the locations of the object are crisp (i.e., no uncertainty).

For a given bead B_i, let $d_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$ denote the distance between locations of the starting location (at t_i) and ending location (at t_{i+1}). Also, let $t_{sv_i} = (t_i + t_{i+1})/2 - d_i/2v^i_{max}$ and $t_{sv_{i+1}} = (t_i + t_{i+1})/2 + d_i/2v^i_{max}$. We observe that each bead had two distinct types of volumes:

1. *Single disk volumes:* For every $t \in [t_i, t_{sv_i}]$, the spatial boundary of the bead at t is a circle with radius $r(t) = v^i_{max}(t - t_i)$, centered at (x_i, y_i). Similarly, for every $t \in [t_{sv_{i+1}}, t_{i+1}]$, the spatial boundary of the bead at t is a circle with radius $r(t) = (t_{i+1} - t)v^i_{max}$, centered at (x_{i+1}, y_{i+1}). Hence, throughout $[t_i, t_{sv_i}]$, the 3D volume of the bead consists of a single cone, with a vertex at (x_i, y_i, t_i) (similarly for $[t_{sv_{i+1}}, t_{i+1}]$).
2. *Two-disks volume:* In-between t_{sv_i} and $t_{sv_{i+1}}$, (i.e., $t \in [t_{sv_i}, t_{sv_{i+1}}]$), the boundary of the bead at t is an intersection of two circles: $C^i_{down}(t)$, centered at (x_i, y_i), with radius $r_{down}(t) = (t - t_i)v^i_{max}$, and $C^i_{up}(t)$, centered at (x_{i+1}, y_{i+1}), with radius $r(t) = (t_{i+1} - t)v^i_{max}$.

We conclude this section with noting one more property of the beads: the projection of the bead B_i onto the the (X,Y) plane is an ellipse (cf. [86, 113]), with foci at (x_i, y_i), and (x_{i+1}, y_{i+1}), and with equation:

$$\frac{(2x - x_i - x_{i+1})^2}{(v^i_{max})^2(t_{i+1} - t_i)^2} + \frac{(2y - y_i - y_{i+1})^2}{(v^i_{max})^2(t_{i+1} - t_i)^2 - (x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2} = 1 \tag{3.4}$$

We will use El_i to denote the ellipse resulting from projecting the bead B_i in the (X,Y) plane. Figure 3.4 provides an illustration of the different components of the (volume of the) bead and its corresponding shapes at different time-points, as projected on the horizontal (X,Y) plane.

Definition 3.3. Given a trajectory $\boldsymbol{Tr}$, its corresponding *uncertain trajectory* UTr is a *sequence of beads*, $B_1, B_2, \ldots, B_{n-1}$.

A *Possible Motion Curve* $PMC(\boldsymbol{Tr})$ of UTr is any function *f: Time* $\rightarrow \mathbf{R}^2$ for which every point (x,y,t), is either a vertex of the polyline of Tr, or it satisfies $(x,y) = f(t)$ and is inside the corresponding bead – i.e., $(\forall t)(t_i < t < t_{i+1}) \Rightarrow ((x,y,t) \in B_i)$.

The concept of a possible motion curve is illustrated in Figure 3.4 and we note that, in a sense, each possible motion curve corresponds to a "possible world" of the object's motion in-between two updates.

We note that in a recent work [94], an analogy is used between the expected-trajectory (i.e., the line segment between consecutive points) where necklace is reserved for the "known" part of the motion, and the uncertain part termed "pendant".

3.4.2 Sheared Cylinders

Another uncertainty model is the one in which an uncertain trajectory is represented as a *sheared cylinder* in the 2D-space + Time coordinate system. This is obtained by associating a fixed uncertainty threshold r at each time-instant with each line segment of the trajectory. Formally (cf. [149]):

Definition 3.4. Let r denote a positive real number and $\boldsymbol{Tr}$ denote a trajectory between the times t_1 and t_n. An *uncertain trajectory UTr* is the pair $(\boldsymbol{Tr}, r)$. r is called the *uncertainty threshold*.

For each point (x,y,t) along $\boldsymbol{Tr}$, its *r-uncertainty area* (or the *uncertainty area* for short) is a horizontal disk (i.e. the circle and its interior) with radius r centered at (x,y,t), where (x,y) is the expected location at time $t \in [t_1, t_n]$.

Let $UTr = (T, r)$ be an uncertain trajectory between t_1 and t_n.

A *Possible Motion Curve* $PMC(\boldsymbol{Tr})$ of UTr is any *continuous function* $f_{PMC^{\boldsymbol{Tr}}}$: *Time* $\rightarrow R^2$ defined on the interval $[t_1, t_n]$ such that for any $t \in [t_1, t_n]$, the 3D point $(f_{PMC(\boldsymbol{Tr})}(t), t)$ is inside the uncertainty disk around the expected location at time t.

For a given uncertain trajectory $UTr = (\boldsymbol{Tr}, r)$ and two end-points (x_i, y_i, t_i), $(x_{i+1}, y_{i+1}, t_{i+1}) \in \boldsymbol{Tr}$, the trajectory volume of UTr between t_i and t_{i+1} is the union of all the disks with radius r centered at the points along the line segment $\overline{(x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1})}$. This volume is actually what defines the "sheared cylinder" in the (X,Y,T) coordinate system. The XY projection of the trajectory volume is called an *uncertainty zone*. Figure 3.5 illustrates the basic concepts associated with the motions uncertainty under this model.

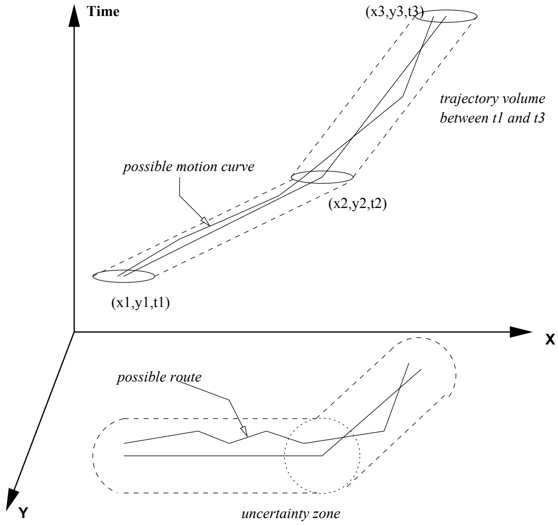


Fig. 3.5 Uncertain Trajectory bounded by Sheared Cylinders, and Possible Motion Curves

3.4.3 Uncertainty on Road Networks

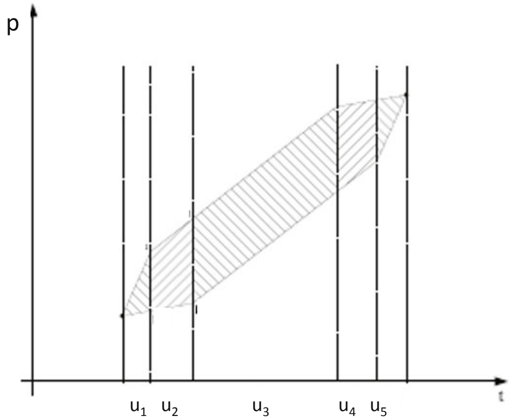


Fig. 3.6 Possible Positions of an Uncertain Object Moving Along a Road Segment: in-between two location samples, the geometry consists of a union of two triangles, two trapezoids and a parallelogram (cf. [5], with permission)

When the motion of an object is restricted by a road network [5, 31, 30, 45, 84, 83, 178], the models described so far (cones/beads and sheared cylinders) become inadequate for representing the moving objects uncertainty. To begin with, road networks are most often represented as (undirected) graph $G(V,E)$, where the vertices correspond to intersections and edges correspond to road segments in-between intersections. Often, a given edge e_{sk} is accompanied with some additional attributes, e.g.,:

– the *length* of e_{sk}, denoted $l(e_{sk})$; and

– the *maximum speed* of e_{sk}, denoted v_{sk}^{max}, which is the upper bound on how fast an object can move along edge e_{sk}.

Rich algebraic specification (types, operators) for representing and querying moving objects on road networks has been presented in [50]. In [5], the authors have extended the corresponding framework by adding data types to represent static (`unqpoint`) and mobile (`munqpoint`) point objects with location uncertainty, along with a detailed specification of the the respective set of predicates/operators. As shown in Figure 3.6, assuming that the location samples at given time-instants are "crisp", the geometric shape bounding the possible whereabouts of the object is a union of three "zones".

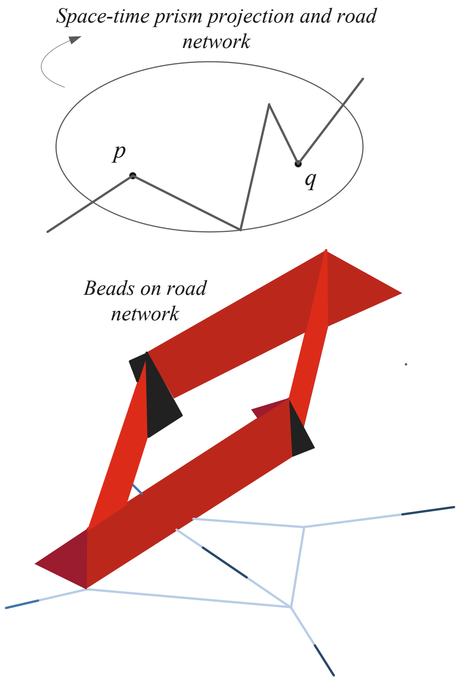


Fig. 3.7 Uncertainty on Road Networks: The possible whereabouts of the moving objects, contrary to the space-time prisms, is now only a subset of the 2D ellipse – the one intersecting the edges. The cones/beads are reduced to unions of vertical line-segments, "sweeping" along, and perpendicular to road network edges (cf. [84], with permission)

The connection (and restrictions) with the beads model is illustrated in Figure 3.7. Note that in road network settings, one cannot consider the entire ellipse (the 2D projection of the bead [113]) as a spatial range of the objects possible locations. Instead, only a subset of it intersecting the edges of the graph can be taken into account [84, 83].

An important consequence of the model of road network trajectories is that the distance between two moving objects can *no longer* be measured using the 2D Euclidian distance (L_2-norm) since the objects are constrained to move along the edges of the road network. Instead we need to rely on the *shortest network distance* which, in turn, may have a two-fold interpretation ([67, 105, 129, 168]):

– shortest path distance, or
– shortest travel-time distance.

Once the distance function is selected, one can proceed with determining the earliest-possible (respectively, the latest-possible) times that an object can be at a given location along an edge, taking the minimum and maximum speed limits along that edge. The bottom part of Figure 3.7 illustrates the shape of the uncertainty "volume" of a given object. Essentially, for each location (x_ℓ, y_ℓ) along a given edge, we have a vertical line-segment bounded by the earliest-possible and the latest-possible time that the moving object can be at that location.

3.5 Processing Spatio-Temporal Queries for Uncertain Trajectories

When it comes to querying uncertain trajectories, as pointed out in several works [88, 103, 149, 171], an important aspect is capturing the uncertainty in the very syntax of the queries. Otherwise, posing a "regular" query to a server without incorporating the uncertainty will bring a situation in which the burden of factoring it out from the answer-set is solely upon the user. As specified in [88], the basic requirements for uncertainty-aware query interface are:

> Immediacy and comprehensiveness: The query interface should immediately build upon the generic uncertainty model to minimize computational effort and exploit all information provided by the uncertainty model.
>
> Generality: The query interface has to provide all prevalent spatial query types for position information such as position query, range query, and next neighbor query

We note that the processing of the popular spatio-temporal queries for uncertain trajectories often follows the typical paradigm of *filtering* + *pruning* + *refinement*, where: (1) the filtering stage brings a subset of the total set of trajectories – candidates – from the secondary storage, which is a superset of the relevant data for the query, based on some indexing structure. The desiderata for this stage are that there should be no false-negatives and as few false-positive as possible; (2) the pruning stage is used to quickly eliminates some of the candidates – e.g., based on the assurance that the desired probability threshold cannot be met by a trajectory which satisfies some properties that are computationally easier to evaluate than the refinement algorithm; (3) the refinement stage, which eliminates all the false positives from among the candidates.

In the rest of this section, we will present examples of solutions to the problem of processing spatio-temporal range queries and nearest neighbor queries for uncertain trajectories for different models of uncertainty discussed in Section 3.4. Since explaining the details of the approaches exceeds the scope of this Chapter, we will try to highlight some of the main intuitive features of the existing results and point out to the body of references where more detailed exposition of various topics is available. We finalize this section with an overview of some miscellaneous queries for uncertain moving objects.

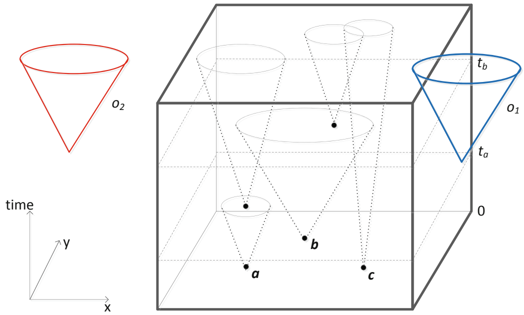


Fig. 3.8 Probability of an Uncertain Moving Object being Inside a given Region (adapted from [21], with permission)

3.5.1 Range Queries for Uncertain Trajectories

The basic form of spatio-temporal range query is:
$\mathbf{Q}_R$: *Retrieve all the trajectories inside the region R between* t_1 *and* t_2.
where R is typically a bounded region, and t_1 and t_2 denote the begin-time and end-time of interest for the query.

3.5.1.1 Instantaneous Range Query for Cones

If the model of uncertainty is the one of a *cone* and the moving objects are assumed to send *(location, time)* updates, along with a given restriction of the velocity then, for a given spatial *pdf*, one can evaluate the probability of a particular object being inside the region R at each $t \in [t_1, t_2]$. For a given t, the generic formula for calculating whether a given object o_i is inside R would be (cf. [21]):

$$\frac{Area(U_{o_i}(t)) \cap Area(R)}{Area(U_{o_i}(t))}$$

where $U_{o_i}(t)$ denotes the shape and/or *pdf* of the uncertainty zone of o_i at time t. As illustrated in Figure 3.8, all the dark objects have 100% probability of being inside the given rectangular region R for each t of interest, however: (1) the blue object (o_1) has non-zero probability of being inside R starting slightly later than the begin-time of interest for the query; (2) the red object (o_2) always has a zero probability of being inside R – hence, from scalability perspective it should be pruned out of the computation in the refinement phase. The evaluation step(s) taken throughout the refinement stage may typically involve expensive numerical integration, consequently, eliminating objects that should not be evaluated yields benefits in terms of the overall execution time.

We re-iterate that the topic of efficient processing of spatio-temporal queries for trajectories is addressed in greater detail in Chapter 2, however, as but one example, we note that [21] specifically uses the VCI index [117] to aid the elimination of candidates for processing range queries. VCI is an index structure based on R-tree [57], with an extra data in its nodes, which is v_{max} - the maximum possible speed of the objects under a given node, with an extra storage of the overall maximum speed at leaf nodes. The construction of VCI is similar to the R-tree, with an additional provision of ensuring that v_{max} is correctly maintained at the root of each sub-tree – which is properly considered when a node split occurs. When VCI is used to process a given query, one must account for the changes of the position (with respect to the stored value) over time. To cater for this, in [21] the Minimum Bounding Rectangles (MBR) used to process a give query at a time instant t are expanded by a factor of $v_{max} \times (t - t_0)$, where t_0 is the time of recording the entry for a given object at VCI. We note that the discussion above illustrates techniques that can be applied for processing range queries over uncertain trajectories at a particular time-instant.

3.5.1.2 Continuous Range Queries for Sheared Cylinders

In Section 3.4, we introduced the concept of a possible motion curve for a given trajectory (PMC(***Tr***)) and hinted that, in some sense, it describes a "possible world" of a particular trip taking place. However, the generic form of a range query $\mathbf{Q}_R$ discussed above does not reflect this anywhere in its syntax. Towards that, the works in [150, 149] have identified the different qualitative relationships that an uncertain trajectory (i.e., the family of its PMC's) could have with the spatial aspect (region R) and temporal aspect (time-interval of interest $[t_1, t_2]$) of the range query.

Firstly, since the location of the object changes continuously, the condition of the moving object being inside R may be true *sometime* ($\exists t$) or *always* ($\forall t$) within $[t_1, t_2]$. Secondly, an uncertain moving object, in addition completely failing to be inside R, may either *possibly* or *definitely* satisfy the spatial aspect of the condition at a particular time-instant $t \in [t_1, t_2]$. In simpler terms, if some PMC(***Tr***) is inside R at t, there is a possibility that it has been the actual motion of the object – however, this need not be the case as there may have been another PMC(***Tr***) that the object has taken along its motion. Let VTr denote the bounding volume of (the union of) all the possible motion curves for a given trajectory ***Tr*** – i.e., the sequence of sheared cylinders (cf. Figure 3.5). Formally, the concept of *possibly* can be specified as $\exists PMC(\boldsymbol{Tr}) \subset VTr$ and the one of *definitely* can be specified as $\forall PMC(\boldsymbol{Tr}) \subset VTr$.

Given the two domains of quantification – spatial and temporal – with two quantifiers each, we have a total of $2^2 \cdot 2! = 8$ operators, and their combinations yield the following variants of the spatio-temporal range query for uncertain trajectories:

- $\mathbf{Q}_R^{PS}$: *Possibly_Sometime_Inside(T,R,t_1,t_2)* $(\exists PMC(\boldsymbol{TR}))(\exists t) Inside(R, PMC(\boldsymbol{TR}), t)$
  Semantics: *true* iff there exists a possible motion curve $PMC(\boldsymbol{Tr})$ and there exists a time $t \in [t_1, t_2]$ such that $PMC(\boldsymbol{Tr})$ at the time t, is inside the region R.
- $\mathbf{Q}_R^{PA}$: *Possibly_Always_Inside(Tr,R,t_1,t_2)*
  $(\exists PMC(\boldsymbol{Tr}))(\forall t) Inside(R, PMC(\boldsymbol{Tr}), t)$
  Semantics: *true* iff there exists a possible motion curve $PMC(\boldsymbol{Tr})$ of the trajectory T which is inside the region R for every t in $[t_1, t_2]$.
  Intuitively, this predicate captures the fact that the object may take (at least one) *specific* possible route, which is entirely contained within the region R, during

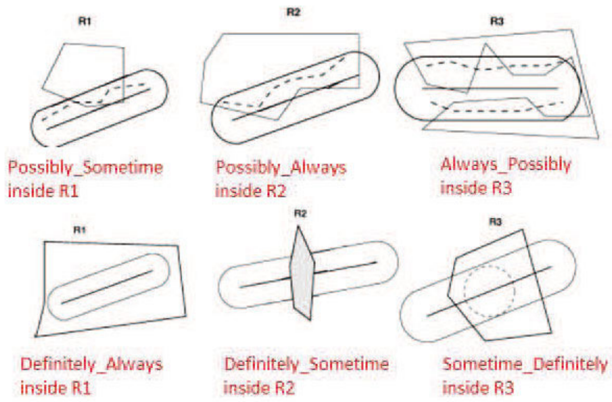


Fig. 3.9 Illustration of the Predicates Capturing the Different Quantifiers of Spatial and Temporal Domains

the whole query time interval.

- $\mathbf{Q}_R^{AP}$: *Always_Possibly_Inside*(Tr,R,t_1,t_2)
  $(\forall t)(\exists PMC(\boldsymbol{Tr}))Inside(R,PMC(\boldsymbol{Tr}),t)$
  Semantics: *true* iff for every time value $t \in [t_1,t_2]$, there exists some (not necessarily unique) $PMC(\boldsymbol{Tr})$ inside (or on the boundary of) the region R at t.
- $\mathbf{Q}_R^{AD}$: *Always_Definitely_Inside*(Tr,R,t_b,t_e)
  $(\forall t)(\forall PMC(\boldsymbol{Tr}))Inside(R,PMC(\boldsymbol{Tr}),t)$
  Semantics: *true* iff at every time $t \in [t_1,t_2]$, every possible motion curve $PMC(\boldsymbol{Tr})$ of the trajectory Tr, is in the region R. Thus, no matter which possible motion curve the object takes, it is guaranteed to be within the query region R throughout the entire interval $[t_1,t_2]$.
- $\mathbf{Q}_R^{DS}$: *Definitely_Sometime_Inside*(Tr,R,t_b,t_e)
  $(\forall PMC(\boldsymbol{Tr}))(\exists t)Inside(R,PMC(\boldsymbol{Tr}),t)$
  Semantics: *true* iff for every possible motion curve $PMC(\boldsymbol{Tr})$ of the trajectory Tr, there exists some time $t \in [t_1,t_2]$ in which the particular motion curve is inside the region R. Intuitively, no matter which possible motion curve within the uncertainty zone is taken by the moving object, it will intersect the region at some time t between t_b and t_e. However, the time of the intersection may be different for different possible motion curves.
- $\mathbf{Q}_R^{SD}$: *Sometime_Definitely_Inside*(Tr,R,t_b,t_e)
  $(\exists t)(\forall PMC(\boldsymbol{Tr}))Inside(R,PMC(\boldsymbol{Tr}),t)$
  Semantics: *true* iff there exists a time point $t \in [t_1,t_2]$ at which every possible route $PMC(\boldsymbol{Tr})$ of the trajectory Tr is inside the region R. In other words, no matter which possible motion curve is taken by the moving object, at the specific time t the object will be inside the query region.

Figure 3.9 illustrates the intuition behind plausible scenarios for the predicates specifying the properties of an uncertain trajectory with respect to a range query, projected in the spatial dimension. Dashed lines indicate the possible motion curve(s) due to which a particular predicate is true, whereas the solid lines indicate the expected routes, along with the boundaries of the uncertainty zone.

A couple of remarks are in order:

1. Although we mentioned that there are 8 combinations of the quantifiers over the variables in the predicates, we listed only 6 of them. This is actually a s-traightforward consequence of the facts from First Order Logic – namely, for any predicate P, given a constant A and variables x and y, it is true that:

$$(\exists x)(\exists y)P(A,x,y) \equiv (\exists y)(\exists x)P(A,x,y)$$

   and

$$(\forall x)(\forall y)P(A,x,y) \equiv (\forall y)(\forall x)P(A,x,y)$$

   This is regardless of the domain of (interpretation of) the variables and the semantics of the predicate P. Hence, we have that *Possibly_Sometime_Inside* is equivalent to *Sometime_Possible_Inside*; and *Definitely_Always_Inside* is equivalent to *Always_Definitely_Inside*. Hence, in effect we have 6 different predicates.
2. Similarly, the formula:

$$(\exists x)(\forall y)P(A,x,y) \Rightarrow (\forall y)(\exists x)P(A,x,y)$$

   is a tautology. In effect, this means that the predicate *Possibly_Always_Inside* is stronger than *Always_Possibly_Inside*, in the sense that whenever *Possibly_Always _Inside* is true, *Always_Possibly_Inside* is guaranteed to be true.
   We observe that the converse need not be true. As illustrated in Figure 3.9, the predicate *Always_Possibly_Inside* may be satisfied due to two or more possible motion curves, none of which satisfies *Possibly_Always_Inside* by itself. When the region R is convex, however, those two predicates are equivalent (cf. [149]).
3. For the same reason as above, we conclude that *Sometime_Definitely_Inside* is stronger than *Definitely_Sometime_Inside*, however, the above two predicates are not equivalent when the region R is convex. In Figure 3.9 this is shown by R_2 satisfying *Definitely_Sometime_Inside*, however, since it does not contain the entire uncertainty disk at any time-instant, it cannot satisfy *Sometime_Definitely_Inside*.

The algorithms for processing the respective predicates involve techniques from Computational Geometry (CG) (Red-Blue Intersection; Minkowski Sum/Difference) and their detailed presentation is beyond the level of detail for this Chapter. Their detailed implementation, along with complexity analysis, is available in [149]. We note that in the global context of query processing, [149] focused on the refinement stage.

In a sense, the predicates described above correspond to the, so called, "MAY" and "MUST" cases for range queries over uncertain trajectories (cf. [104, 103, 162])

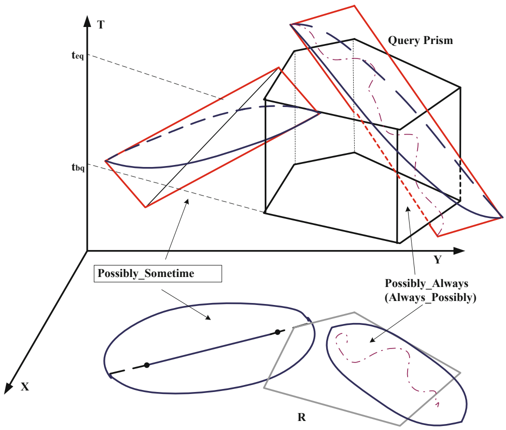


Fig. 3.10 Satisfying the Possibly_Sometimes and Possibly_Always Predicates for Beads

and, more specifically, the "Inside" property is discussed as a predicate in the generic query interface discussed in [88].

3.5.1.3 Continuous Range Queries for Beads/Necklaces

As demonstrated in [145], capturing qualitative relationships between a range query and uncertain trajectories whose uncertainty model is the one of space-time prisms (equivalently, beads) can be done using the same logical formalizations from [149]. Not only the same predicates are applicable, but also the relationships among them in terms of *Possibly_Always_Inside* being stronger than *Always_Possibly_Inside*, and *Sometime_Definitely_Inside* being stronger than *Definitely_Sometime_Inside* are valid.

The main difference is that the bead as a spatio-temporal structure yields a bit more complicated refinement algorithms than the ones used in [149] for sheared cylinders. As an illustrating example, Figure 3.10 shows Possible Motion Curves that cause the two predicates with existential quantifier over the spatial domain (*"Possibly"*) to be true.

Although the model of beads is more complicated for the refinement stage, it opens a room for improving the overall query processing when it comes to the *pruning* phase. Namely, one can utilize vertical cylinders surrounding a particular bead to eliminate a subset of the candidate trajectories from the answer-set more efficiently. As shown in Figure 3.11, the vertical cylinder surrounding the bead B_i does not intersect the query region R, hence, there is no need for detailed verification of any predicate capturing the uncertain range query with respect to B_i. The benefits of two pruning strategies are discussed in more details in [145].

As an illustrative example of pruning phase, below we show the steps of the algorithm for processing *Possibly_Sometime_Inside* predicate. Let El_i denote the ellipse which is the (X,Y) projection of the bead B_i, and let F_i^l (= (x_i, y_i)) and F_i^u

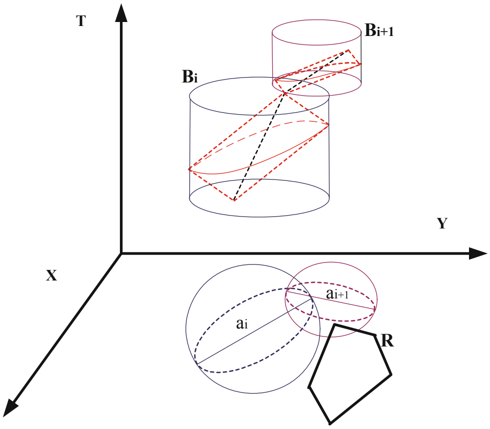


Fig. 3.11 Pruning of Beads which do not Qualify for the Answer of a Range Query

$(= (x_{i+1}, y_{i+1}))$ denote the 2D projections of its *lower* and *upper* foci in temporal sense – i.e., F_i^l occurs at time t_i and F_i^u at time t_{i+1}. For complexity analysis, assume that the region R has m edges/vertices, and an one-time pre-processing cost of $O(m)$ has been performed to determine the angles in-between its consecutive vertices with respect to a given point in R's interior [107]. The refinement algorithm can be specified as follows:

1. **If** $(t_i \in [t_1, t_1] \wedge F_i^l \in R) \vee (t_{i+1} \in [t_1, t_2] \wedge F_i^u \in R)$
2. **return true**
3. **else if** $(El_i \cap R \neq \emptyset)$
4. **return true**
7. **return false**

Each of the disjuncts in line 1. can be verified in $O(\log m)$ due to the convexity of R (after the one-time pre-processing cost of $O(m)$) [107]. Similarly, by splitting the ellipse in monotone pieces (e.g., with respect to the major axis), one can check its intersection with R in $O(\log m)$, which is the upper bound on the time-complexity of the algorithm.

We note that many of the works on formalizing the predicates that capture different types of spatio-temporal range queries are geared towards extending the querying capabilities of MODs. Consider, for example the following query:
$\mathbf{Q}_R^U$: *"Retrieve all the objects which are possibly within a region R, always between the earliest*[5] *time when the object A arrives at locations* L_1 *and the latest time when it arrives at location* L_2*".*
If the corresponding predicates are available, this query can be specified in SQL as:

```
WITH Earliest(times) AS
  SELECT When_At(trajectory,L_1)
  FROM MOD
```

[5] Observe that a given object may pass through a given point along its route more than once

```
    WHERE oid = A
  WITH Latest(times) AS
    SELECT When_At(trajectory,L_2)
    FROM MOD
    WHERE oid = A
  SELECT M1.oid
  FROM MOD as M1
  WHERE
  Possibly_Always_Inside(M1.trajectory,R,
                             MIN(Earliest.times),
                             MAX(Latest.times))
```

3.5.1.4 Uncertain Range Queries on Road Networks

When the motion of a given object is constrained to an existing road network, one of the sources of its location uncertainty is due to the fact that the objects speed may vary between some v_{min} and v_{max} along a given segment – which we described in Section 3.4.3. However, there is another source of the uncertainty of such motion – namely, the low sampling-rate of the on-board GPS devices e.g., due to unavailability of satellite coverage in dense downtown areas. The main consequence of this is that the distance between two consecutive sampled positions can be large: e.g., it can be over 1.3km when sampling every 2 minutes, even if a vehicle is moving at the speed as low as 40km/h. The additional uncertainty is reflected in the fact that there may be many possible paths connecting the two consecutively sampled positions, which satisfy the temporal constraints of the actual consecutive samples. The problem is even more severe for vehicles travelling with higher speeds, as there may be several intersections between two consecutive samples.

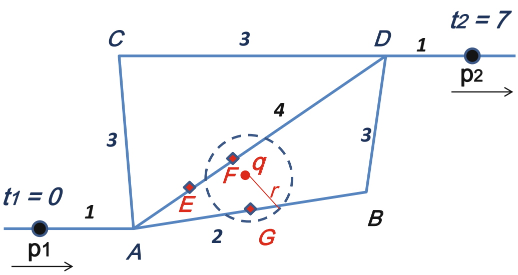


Fig. 3.12 Uncertainty on Road Networks Due to Low Location-Sampling Frequency (object may take different routes between intersections)

As an example, consider the scenario depicted in Figure 3.12. It shows two consecutive location-samples: L_1 at $t_1 = 0$, and L_2 at $t_2 = 7$. There are three possible routes between vertices (intersections) A and D: — $(\overline{AC},\overline{CD})$ with travel time 4 + 2 = 6 time units; — $\overline{AD}$ with travel time of 4; — and $(\overline{AB},\overline{BD})$, with a travel time of 2 + 3 = 5 time units. Given the information about minimum travel time cost – e.g., 1 time unit between L_1 and A, as well as between D and L_2, is 1 time unit, consider the following query:

Q_R:*Retrieve all the moving objects that are within distance r from the location Q between t' = 3.5 and t" = 4.*

Clearly, it is impossible that the object has travelled along the route $(\overline{AC}, \overline{CD})$ because with the maximum speed at each segment, the earliest time of arrival at the location L_2 would be 8. This leaves only two *possible paths*: $\overline{AD}$ and $(\overline{AB}, \overline{BD})$. Following are the main observations regarding these plausible routes:

- If object a travelled along $\overline{AD}$ with the maximum speed, it will definitely be inside the spatio-temporal cylinder (based at the 2D disk centered at q and with radius r) between t = 3.5 and t = 4. However, now the question becomes, *what if the moving object did not travel using the maximum speed*? What is the probability of a satisfying Q_R, given some *pdf* of its speed?
- If object a travelled along $(\overline{AB}, \overline{BD})$ using the maximum speed, it will *not* qualify as an answer to Q_R. However, if the moving object uses smaller speed, then there may be a possibility of it entering the spatio-temporal query cylinder sometime during the time-interval of interest. Namely, the object can be at the location L_Q along the segment $\overline{BD}$ at any time during $t = 3.3$ and $t = 3.7$. Now the question again becomes, given the *pdf* of its speed, what is the probability of a satisfying Q_R. As an additional observation, we note that the object a can be anywhere within a particular segment at a given time-instant – as illustrated in Figure 3.12 for the time $t = 3.7$.

The models for uncertain trajectories on road networks in terms of possible locations at a given time-instant have been considered in [5, 45, 83]. The combination of the effects of choosing possible path together with the location uncertainty along a particular one has been formalized in [178].

Definition 3.5. Given two trajectory samples (t_i, p_i) and (t_{i+1}, p_{i+1}) of a moving object a on road network, the set of *possible paths (PP_i)* between t_i and t_{i+1} consists of all the paths along the routes (sequence of edges) that connect p_i and p_{i+1}, and whose *minimum time costs* (tc) *are not greater than* $t_{i+1} - t_i$, i.e.,

$$PP_i(a) = \{P_j \in Paths(p_i, p_{i+1}) | tc(P_j) \leq t_{i+1} - t_i\},$$

where *Paths* denotes all the paths between p_i and p_{i+1}, $tc(P_j)$ is the sum of all the $tc(e)$ of $e \in P_j$.

If the *pdf* of selecting a particular possible path is uniform, then:

$$Pr_{i,j}(a) = Pr[PP_i(a) = P_j] = \frac{1}{|PP_i|}$$

where $|PP_i|$ denotes the cardinality of the set of all the possible paths PP_i. As another example, if the probability of a particular path being taken by an object a is inversely proportional to time-cost of that path, then:

$$Pr_{i,j}(a) = Pr[PP_i(a) = P_j] = \frac{1/tc(P_j)}{\sum_{P_x \in PP_i(a)} 1/tc(P_x)}$$

Even if a particular path P_j is considered, the location of the moving object at a given time-instant $t \in (t_i, t_{i+1})$ need not be crisp (i.e., certain) because the speed along P_j may fluctuate. However, the set of possible locations can be restricted as follows:

Definition 3.6. Given a path $P_j \in PP_i(a)$, the *Possible Locations* of a given moving object a with respect to P_j at $t \in [t_i, t_{i+1}]$ is the set of all the positions p along P_j from which a can reach p_i (respectively, p_{i+1}) within time period $t - t_i$ (respectively, $t_{i+1} - t$) i.e.,

$$PL_{i,j}(t) = \left\{ p \in P_j \,\middle|\, \begin{array}{l} tc_{P_j}(p_i, p) \leq t - t_i \\ tc_{P_j}(p, p_{i+1}) \leq t_{i+1} - t \end{array} \right\} \tag{3.5}$$

As an example, in the case of a uniform *pdf*, the probability that the object a is between positions p_A and p_B along a possible path P_j, whose network-distance is $d(p_A, p_B)$, is:

$$Pr[p_a(t) \in [p_A, p_B]] = Pr_{i,j}(a) \cdot \frac{d(p_A, p_B)}{\overline{PL_{i,j}(t)}} \tag{3.6}$$

where $\overline{PL_{i,j}(t)}$ denotes the the network-length of $PL_{i,j}(t)$. Formula 3.6 illustrates the joint consideration of the probability that a particular path P_j is being selected from among the possible ones, together with the probability of the object being somewhere along the segment $\overline{p_A, p_B}$ at a given time-instant t [178].

Clearly, given an existing road-map along with the *(location,time)* samples, a methodology is needed to construct all the possible trajectories that satisfy the temporal constraints of the samplings. In addition, one needs to determine the *pdf*s of the location uncertainty along different possible paths. Algorithmic solutions for two types of probabilistic range queries: *snapshot* (instantaneous) and *continuous*, are presented in [178], along with a novel indexing structure – Uncertain Trajectory Hierarchy (UTH), used to index the road network, object movement and trajectories in a hierarchical style and to improve the overall efficiency of the query processing.

3.5.2 Nearest-Neighbor Queries for Uncertain Trajectories

We now present some of the techniques that have addressed variants of the problem of efficient management of Nearest-Neighbor (NN) queries for uncertain trajectories. Before we proceed with the details, we note that an assumption commonly used in the literature (e.g., [21, 142]) is that the locations of the uncertain objects are *independent* random variables.

The basic form of spatio-temporal range query is:

$\mathbf{Q}_{NN}$: *Retrieve the nearest neighbors of the trajectory* $\boldsymbol{Tr_q^u}$ *between* t_1 *and* t_2.

where $\boldsymbol{Tr}_q^{\mathbf{u}}$ denotes the uncertain querying trajectory.

3.5.2.1 NN Query for Cone Uncertainty Model

Recall Figure 3.8 used to explain the intuition behind spatio-temporal range query processing for cone-like model of uncertainty. If we take a horizontal "slice" at a particular time-instant, we will obtain all the spatial locations of the objects at that time-instant.

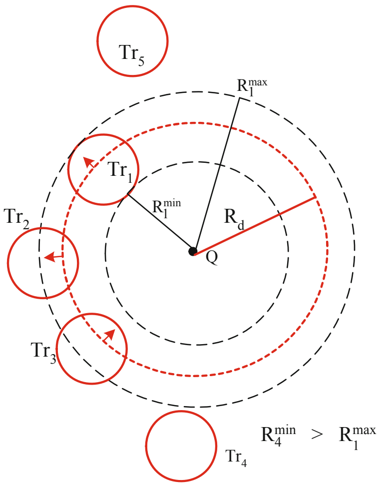


Fig. 3.13 Relationship Among Uncertain Moving Objects when the Querying Object is Crisp

Consider a scenario in which that we are given a query object o_Q whose location at a particular time instant is *crisp*, i.e., a 2D point **Q**, with no uncertainty associated with it, and assume that the possible locations of the other objects are disks with radius r (cf. Figure 3.13).

An important observation was made in [21], which can be used to effectively prune all the objects that cannot qualify to have a non-zero probability of being a nearest neighbor to o_Q. Namely, the distance between **Q** and the *most distant point of the closest disk* (e.g., R_{max} in Figure 3.13), is an *upper bound* on the distance that *any possible nearest neighbor* of $\boldsymbol{Tr}_q$ can have. Consequently, any object $\mathbf{o}_i$ (a snapshot of a trajectory $\boldsymbol{Tr}_i$) whose closest possible distance to **Q**, denoted with R_i^{min}, is larger than R_{max}, has a zero probability of being a nearest neighbor to $\boldsymbol{Tr}_q$ and can therefore be safely pruned. As can be seen from Figure 3.13, $R_4^{min} > R_1^{max}$ and similarly $R_5^{min} > R_1^{max}$, which means that $\boldsymbol{Tr}_4^{\mathbf{u}}$ and $\boldsymbol{Tr}_5^{\mathbf{u}}$ have zero probability of being a nearest neighbor of Tr_q.

Once the trajectories that do not qualify to be in the answer set of an NN query have been pruned, the next task is to evaluate the probability of a given trajectory $\boldsymbol{Tr}_i^{\mathbf{u}}$ being within a given distance R_d from **Q**:

$$P_{i,Q}^{WD}(R_d) = \int\int_A pdf_i(x,y)\,dx\,dy \tag{3.7}$$

where A, the integration bound, denotes the area of the intersection of the disk with radius R_d centered at $\mathbf{Q}$ and the uncertainty disk of Tr_i, with a corresponding $pdf_i(x,y)$.

Then, in order to calculate the probability that the trajectory of a given object, $\boldsymbol{Tr}_j^{\mathbf{u}}$, is a nearest neighbor of the crisp querying object $\boldsymbol{Tr}_q$ at a given time instant, one needs to consider:

1. The probability of $\boldsymbol{Tr}_j$ being within distance $\leq R_d$ from $\boldsymbol{Tr}_q$; combined with:
2. The probability that *every other* object Tr_i $(i \neq j)$ is at a distance greater than R_d from the location $\mathbf{Q}$ of $\boldsymbol{Tr}_q$; and
3. The fact that the distributions of the objects are assumed to be independent from each other.

Using these observations, the generic formula for the nearest-neighbor probability (cf. [21]) is:

$$P_{j,Q}^{NN} = \int_0^{\infty} pdf_{j,Q}^{WD}(R_d) \cdot \prod_{i \neq j}(1 - P_{i,Q}^{WD}(R_d))\,dR_d \tag{3.8}$$

As pointed out in [21], the boundaries of the integration need not be 0 and ∞ because the effective boundary of the region for which an object *can qualify* to be a nearest neighbor of $\boldsymbol{Tr}_q$ is the ring centered at $\mathbf{Q}$ with radii R_{min} and R_{max}. More specifically, $pdf_{j,Q}^{WD}(R_d)$ is 0 for any $R_d < R_j^{min}$ and $1 - P_{i,Q}^{WD}(R_d)$ is 1 for $R_d < R_i^{min}$.

By sorting the objects that have a non-zero probability of being nearest neighbors according to the minimal distances of their boundaries from $\mathbf{Q}$, one can break the evaluation of the integral from Equation 3.8 into subintervals corresponding to each R_{min_i} and the computation of the $P_{j,Q}^{NN}$ can be performed in a more efficient manner, based on the sorted distances and the corresponding intervals [21]. The importance of this observation is in the fact that the the integrals (cf. Equation 3.8) are likely to be computed numerically. For a uniform *pdf* of the location uncertainty, the objects can be sorted according to the distances of their expected locations from the querying object.

3.5.2.2 NN Query for Sheared Cylinders – Continuity and Time-Parameterization

While the methodology explained above is sound for evaluating a snapshot (i.e., instantaneous) NN queries, an important property of the NN queries in MOD settings is that their answer over the time-interval of interest needs to be parameterized [139]. In other words, as the querying object itself, as well as the other objects are continuously moving, the nearest neighbors will change over time. To illustrate this feature, assume that we have a MOD with four trajectory-segments
$\boldsymbol{Tr}_1 = \{(120,60,10),(220,300,20)\}$
$\boldsymbol{Tr}_2 = \{(310,100,10),(190,260,20)\}$

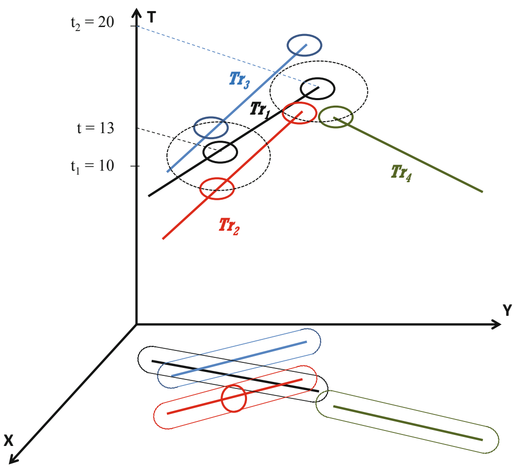


Fig. 3.14 NN Queries for Uncertain Trajectories: The Continuity Aspect of the Answer

$\boldsymbol{Tr}_3 = \{(150,100,10),(30,260,20)\}$
$\boldsymbol{Tr}_4 = \{(370,570,10),(270,330,20)\}$
corresponding to the scenario in Figure 3.14 without the uncertainty component, and consider the following query:
Q_NN: *Retrieve the nearest neighbor of the trajectory* $\boldsymbol{Tr}_1$ *between* $t_1 = 10$ *and* $t_2 = 20$.
Using the existing approaches [105, 139]), the answer $\mathbf{A}_{Q\_NN}$ to the query is the set $\{[\boldsymbol{Tr}_3,(10,15)],[\boldsymbol{Tr}_2,(15,20)]\}$, meaning that during the first 5 seconds of the time interval of interest, i.e., between $t_1 = 10$ and $t = 15$, the nearest neighbor of $\boldsymbol{Tr}_1$ is the trajectory $\boldsymbol{Tr}_3$ and during the last 5 seconds, i.e., between $t = 15$ and $t_2 = 20$, the nearest neighbor of $\boldsymbol{Tr}_1$ is the trajectory $\boldsymbol{Tr}_2$.

However, if we take the uncertainty into consideration (cf. Figure 3.14), assuming that at every time-instant, an object can be anywhere within a disk with a 30 meter radius, we observer, for example that at time $t = 13 (< 15)$, both $\boldsymbol{Tr}_3$ and $\boldsymbol{Tr}_2$ have a *non-zero probability* of being the NN-trajectory to $\boldsymbol{Tr}_1$. However, that is not the case for $\boldsymbol{Tr}_4$ which, at $t = 13$, cannot possibly be the nearest neighbor to $\boldsymbol{Tr}_1$. Moreover, at $t_2 = 20$, we note that $\boldsymbol{Tr}_4$—which was not part of the answer $\mathbf{A}_{Q\_NN}$ for crisp trajectories—also has a non-zero probability of being the NN-trajectory to $\boldsymbol{Tr}_1$. Hence, now some new important aspects emerge:

- *Syntax and Semantics of the Answer*: how can we capture the time-parameterized nature of the answer in a compact manner?
- *Ranking:* how can we establish the rank of a given trajectory's probability (e.g., highest or lowest) of being a nearest neighbor [133] at a particular time instant?
- *Continuity:* how can we efficiently detect the changes to the continuous ranking of the objects that qualify to be nearest neighbors (with non-zero probability) throughout the time-interval of interest?

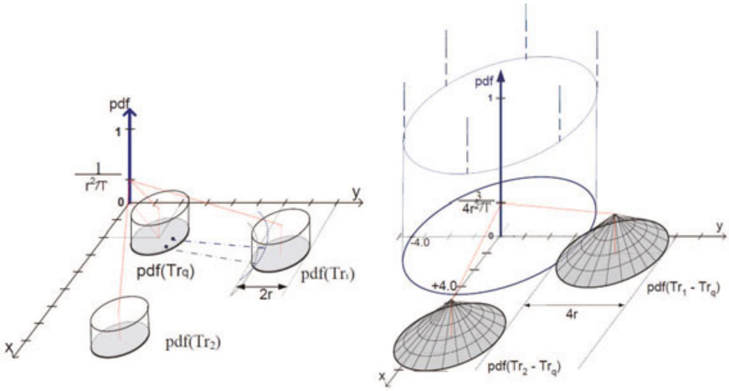


Fig. 3.15 Left: Uncountably many integrations needed to evaluate Within_Distance probability in "native" space; Right: After convolution, the querying trajectory is fixed and crisp

The first source of complication comes when the querying object is no longer crisp – how is the probability of being within distance evaluated in such cases? As shown in the left portion of Figure 3.15, we need to take infinitely many integrations over the entire disk of possible locations for the uncertain trajectory. However, since the relevant part for determining the nearest neighbor status between a given trajectory and the querying trajectory is their *distance*, it was observed in [148] that one may actually focus on a random variable specifying the difference between the two random variables: – one corresponding to a particular trajectory; – one corresponding to the querying trajectory. It is a consequence of the laws of probability theory that the difference-variable will have a *pdf* which is a *convolution* of the *pdf*s of the original variables and, in addition, as demonstrated in [148] – if the original *pdf*s have circular symmetry, so will their convolution. What is enabled by this observation is that one can "snap" the querying trajectory to the origin of the respective spatial coordinate system, and calculate the $P^{WD}_{i,q}$ using the results from [21], except the non-querying trajectories will be transformed by: – translation of the expected location; – modification of their location *pdf*. An illustration of this is provided in the right portion of Figure 3.15 – in effect, reducing an extra-level of an outer integration.

Most importantly, though, since the transformation described above is applicable to every time-instant, one can tackle the continuity aspect by using the, so called, *difference trajectories*. Specifically, instead of considering the original expected trajectories in the MOD to evaluate the expected distance from the querying trajectory, one can assume that the querying trajectory is "snap"-ed to the origin, and consider the modified trajectories to evaluate the change of the mutual distance. The main consequence of this, which is enabling the design of the efficient algorithm for calculating the answer to the continuous NN query for uncertain trajectories (again, assuming the location *pdf* has circular symmetry around the centroid) is that if the centroid of $\boldsymbol{Tr}_i^u - \boldsymbol{Tr}_q^u$ is closer to the coordinate-center than the centroid of $\boldsymbol{Tr}_j^u - \boldsymbol{Tr}_q^u$, then $\boldsymbol{Tr}_i^u$ has a higher probability of being the nearest neighbor of $\boldsymbol{Tr}_q^u$ than $\boldsymbol{Tr}_j^u$.

Given the observations above, along with the fact that the distance function between the centroids of the querying trajectory and an individual trajectory changes as a hyperbola [9, 119] over time[6] the *continuity* and *ranking* aspects can be handled based on the following properties:

- The nearest neighbor with highest probability will be the trajectory whose distance function determines the lower envelope of the collection of the distance function. The rank will change in the cusps of the lower envelope (i.e., whenever it becomes determined by the distance function of another trajectory).
- The trajectory with the second-highest probability of being a nearest neighbor in a given time-interval can be obtained if the one defining the lower envelope in that time-interval is removed (and recursively for the k-th highest probability ($k \geq 2$).
- Regardless of the particular *pdf*, for as long as the uncertainty zone of the objects' locations is bounded by a circle with radius r, every trajectory whose distance function is further than $4 \times r$ from the lower envelope can be pruned from consideration for a nearest neighbor with non-zero probability.

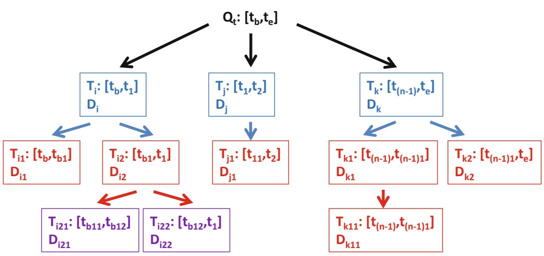


Fig. 3.16 Time-parameterization of the Answer to a Continuous Nearest Neighbor Query for Uncertain Trajectories (sheared cylinder model)

When it comes to the structure of the answer that is to be presented to the user [148] postulates that one compact structure can be obtained by splitting the time-interval of interest, say $[t_b, t_e]$, into sub-intervals $[t_b, t_1]$, $[t_1, t_2], \ldots, [t_{n-1}, t_e]$ so that the trajectory that has the highest probability of being the nearest neighbor of $\boldsymbol{Tr}_q^{\mathbf{u}}$ in each sub-interval is unique.

Subsequently, each such sub-interval can be further split into its own sub-intervals – e.g., $[t_{i-1}, t_i]$ is split into $[t_{i-1}, t_{(i-1),1}]$, $[t_{(i-1),1}, t_{(i-1),2}], \ldots, [t_{(i-1),(k-1)}, t_i]$. To each of this sub-intervals, again a unique trajectory is matched – representing the trajectory which would have been the actual highest-probability nearest neighbor of $\boldsymbol{Tr}_q^{\mathbf{u}}$, if the MOD did not contain $\boldsymbol{Tr}_{i-1}^{\mathbf{u}}$. Towards that, a tree-structure called IPAC-NN tree (Intervals of Possible Answers to Continuous-NN) was introduced, shown in Figure 3.16 with the following properties:

[6] Since squaring each distance function will not disturb the relative ordering, one may readily work with the corresponding parabolae.

- The root of the tree is node labelled with the description of the parameters of interest for the specification of the query, e.g., $\boldsymbol{Tr}_q^{\mathbf{u}}$, along with $[t_b, t_e]$.
- The root has one child for each sub-interval of $[t_b, t_e]$, throughout which there is a unique uncertain trajectory $\boldsymbol{Tr}_i^{\mathbf{u}}$ having the highest probability of being the nearest neighbor to $\boldsymbol{Tr}_q^{\mathbf{u}}$. Each child of the root is labelled with the corresponding trajectory (e.g., $\boldsymbol{Tr}_i^{\mathbf{u}}$) and the time-interval of its validity as the highest probability nearest neighbor (e.g., t_{i-1}, t_i) in Figure 3.16).
- After obtaining the respective labels from the respective parent-node, each child-node checks whether if it is removed from the MOD, there could still be some object with nonzero probability of being the nearest neighbor of $\boldsymbol{Tr}_q^{\mathbf{u}}$ in the time sub-interval of its label.
  - If so, then it is an internal node, and each internal node follows the principle of splitting its own (sub)interval like it has been done in the root, and uses the same labelling for its children.
  - If not, then that node is a leaf-node.

The construction of the IPAC-NN tree is based on the algorithm for constructing the collection of lower envelopes of the distance functions.

3.5.2.3 NN Query for Beads Uncertainty Model

Recall that at the heart of the space-time prisms (beads) is the assumption that the motion of the objects is constrained by some maximal velocity v_{max}, and the objects can take any speed $v \in [v_{min}, v_{max}]$ in-between two consecutive *(location,time)* updates.

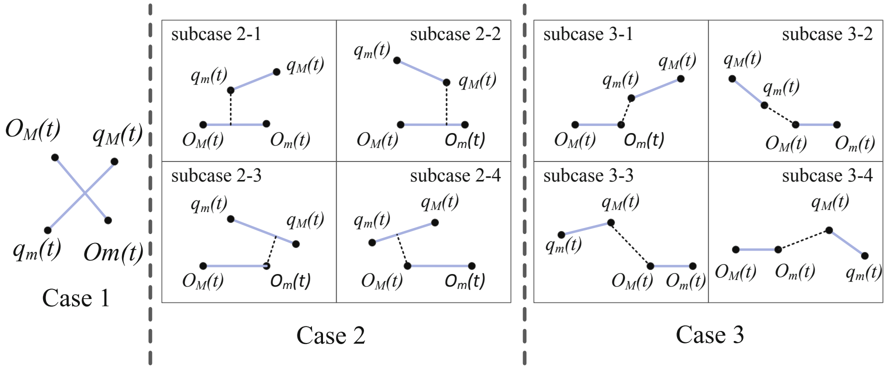


Fig. 3.17 Different Cases of Evaluating the Distance Between Two Objects with Uncertain Velocities (cf. [67], with permission)

The assumption for uncertain speed and crisp update points clearly affects how the minimum possible distance between two objects can vary in-between updates.

To capture the different variations, in [67] three basic cases of the minimum distance were identified (cf. Figure 3.17):

- The two objects are moving along paths that intersect.
- The two objects are on the expected segments that do not intersect, and the minimum distance is based on a perpendicular from a point on one segment to a point on the other.
- The two objects are on expected segments that do not intersect, however, their minimum distance happens when each of the two objects is located in some of the end-points of the expected segment of motion.

Based on the three cases for instantaneous distance values, when it comes to monitoring the distance between a given object and the querying object over a time-interval, the so called *function-switching time points* are determined. The key property is that in-between two consecutive function-switching time points the function describing the variation of the minimum-distance between the querying object q and a moving object o (denoted $d_{o,q}(t)$ in [67]) is one and the same function of time.

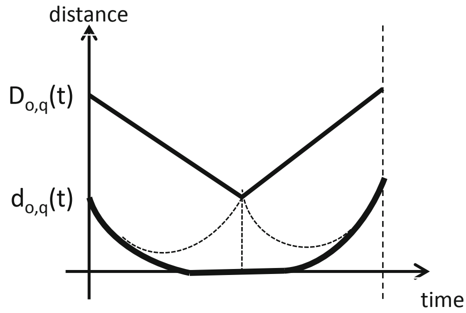


Fig. 3.18 Boundaries on the Possible Distances Between Two Objects (adapted from [67], with permission)

In addition to $d_{o,q}(t)$, a similar formalization of $D_{o,q}(t)$ was presented, where $D_{o,q}(t)$ describes the maximum-distance function between q and o over time. In effect, these two functions determine the boundaries for the possible distance between the two objects with uncertain speeds, q and o. An illustration of these boundaries is presented in Figure 3.18.

Given the goal of the work – to determine the probabilistic answer for the continuous K- nearest neighbors for a querying object q, the solution proceeds in three main stages:

1. *Pruning*: in this stage, based on the boundaries of the possible-distance zones – i.e., $D_{o,q}(t)$ and $d_{o,q}(t)$, the objects for which it is impossible to be among the K closest ones to q during the time-interval of interest are eliminated from further consideration.
2. *Candidate-distilling*: during this stage, sub-intervals are identified, during which the set of possible K nearest neighbors consists of same objects. To determine the time-instants during which the change occurs, one needs to determine the

time t_c when $d_{o_i,q}(t_c = D_{o_j,q}(t_c)$ – i.e., the minimum-distance of the object o_i becomes equal to the maximum-distance of the object o_j with respect to the querying object q. In this case, if o_j was among the $PKNN(q)$ (i.e., possibly among the K nearest neighbors) before t_c, then it will be substituted by o_i at/after t_c.

3. *Ranking*: in this phase, a confidence value, based on a reasonable probability-model is determined for each object among the candidates.

In [68], an indexing structure was proposed – TPR^e tree – which can be applied to index trajectories with uncertain velocities, thereby avoiding unnecessary I/O operations from the secondary storage. Extending the paradigm from [67], scalable efficient techniques were presented to process probabilistic variants of the $K-NN$ query, along with a variant of the range query – a moving range (i.e., within a given distance from a moving object).

3.5.2.4 NN Query for Uncertain Trajectories on Road Networks

As discussed in Section 3.4.3, assuming that the periodic *(location,time)* samples do not contain any errors, the main source of uncertainty for the objects moving along road networks is the fluctuation of the speed between some minimal value v_{min} and maximal value v_{max}. Typically, in these settings, v_{max} corresponds to some speed-limits dictated for a particular type of road segment (e.g., highway portion vs. street in downtown area).

When it comes to processing NN query on road networks, the key aspect is the *distance function*. Contrary to the motion in a free 2D space, where the distance at a particular time-instant is the L_2 – Euclidian distance, as we mentioned in Section 3.4 in these settings, the distance can be evaluated only based on the existing network, i.e., the underlying graph representing it [67, 105, 129, 168]. Hence, the typical strategies rely on either the shortest path distance, or shortest travel-time distance.

A recent approach for tackling Continuous $K-NN$ queries for objects moving along segments of road networks with Uncertain speeds (CUkNN) has been presented in [91]. In a similar spirit to [67], for a given object o and a querying object q, two bounding distance-functions are presented:

1. $MaxD_{q,o,}(t)$ determining how the maximum distance between o and q varies over time.
2. $MinD_{q,o,}(t)$ determining how the minimum distance between o and q varies over time.

When calculating the distance functions, given the bounds v_{min} and v_{max}, for each of o and q at a given time-instant, the closest possible location and the furthest possible location from a vertex (e.g., an intersection in the graph-based network model) along the direction of movement in the current segment are obtained. The shortest path distance between the vertices of the graph incident to the edges along which o and q travel, together with the bounds for closest/furthest possible locations, is used to calculate the total value of $MinD_{q,o,}(t)$ and $MaxD_{q,o,}(t)$. The crux is that these two functions – which were hyperbolae for the motion in free 2D space (or, equivalently, parabolae) – now correspond to *line segments* in the *(distance,time)* space.

The typical CUkNN query processing proceeds with the standard three stages: *pruning* – eliminating the objects with zero probability of being one of the K nearest neighbors of q; *refinement* – where the possible candidates are ranked; and *probability evaluation* – the last phase where the actual probabilities for the objects from the refinement phase are calculated. We close this part with a note that a methodology for processing NN query in the settings in which a model based on location-uncertainty is coupled with a network distance function based on shortest travel-time has recently been presented in [147]

3.5.3 Potpourri: Some Miscellaneous Queries/Predicates for Uncertain Trajectories

The range and nearest neighbor queries have been identified as important categories even in traditional databases settings. However, there are certain predicates that are topological in nature [33, 76] which have recently been considered in the context of uncertain spatio-temporal data.

Given that the beads (or, space-time prisms) can be described by polynomial constraints (cf. Section 3.4), various queries that are well-suited for constrained database paradigm can be explored. For example, one can envision predicates like *inPrism(r,p,q,v)* specifying that the point r is inside the space-time prism defined by points p and q, with a maximum speed v, where p is preceding q in time.

An example query that is of interest to geographers [61] is the, so called, *alibi-query*. Given two space-time prisms, representing the uncertain motions of two individuals, the alibi-query asks whether those objects had a chance to meet – essentially, whether their corresponding space-time prisms intersect. It was observed that relying on the quantifier-elimination approaches for first order theories to process the alibi-query was computationally cost-prohibitive, and an analytic solution to this problem was presented in [82].

The *inside*-ness (cf. Section 3.5.1.2) can also be viewed as a topological property describing a possible relationship between an uncertain trajectory and a spatio-temporal range corresponding to the query-prism. This is but an example of the perspective taken in [94], where different topological predicates for uncertain trajectories under the, so called, *pendant* model are discussed. The pendant model is, in some sense, equivalent to the beads for moving points, however, the formalization in [94] addresses both uncertain moving points (*unmpoint*) and uncertain moving regions (*unmregion*). Extending the work in [34] presenting the *STP* framework for formalizing Spatio-Temporal Predicates, a collection of Spatio-Temporal Uncertain Predicates (*SUTP*s) is presented, based on the pendant model. Formally, a SUTP is a boolean expression containing:

- *Topological predicates*: disjoint, meet, overlap, covers, coveredBy, equal, inside, contains.
- *Traditional logic operators*: $\neg, \vee, \wedge, \exists, \forall$.
- *Set expressions*: $\cup, \cap, \in, \subset$.
- *Model-operations*: e.g., at_instance; temp_select (along with the distance function *dist*).

An interesting geometric approach towards the scalability aspect of the probabilistic NN query processing is presented in [4]. Namely, in a similar spirit to [22], the work introduces the concept of Probabilistic Voronoi Diagram that can be used to prune the objects that do not qualify to be in the answer set of the NN-query – i.e., ones whose Voronoi cells are not adjacent to the Voronoi cell of the querying object.

3.6 Summary

In this Chapter, we presented an overview of the research trends addressing problems related to modelling and querying of uncertain trajectories data. After a selected historic overview of the treatment of uncertainty in a few disciplines which, in one way or another, have impacted the evolution of the thought related to the theme of this Chapter, we gave a more focused overview of the uncertainty in the fields of spatial and temporal databases. In the last two sections, we gave detailed discussions related to different models of trajectories that capture the uncertain data, and the issues that arise when processing queries over such data.

We now bring a few observations regarding the role and impact of trajectories uncertainty in a broader context and application settings/requirements[7]:

- In order to reduce the communication and bandwidth consumption, moving objects may have a ”contract” with the MOD server based, e.g., on the distance-based dead-reckoning policy [161, 162]. In these settings, each object will periodically transmit an update of the form *(location, time, velocity)* and will not transmit any other location update, for as long as the deviation between the actual location (as observed e.g., by the on-board GPS device) and the expected location in the MOD does not exceed certain threshold. Clearly, in-between updates, the MOD server cannot have any certain knowledge about the objects whereabouts. In the case that the objects are moving along a road-network, clearly, the location uncertainty can be reduced [45].
- In order to reduce the storage requirements for the large quantities of *(location,time)* data, sometimes MOD servers may apply data-reduction techniques [15]. Clearly, reducing the size of the data points will ultimately introduce an uncertainty, although the size vs. imprecision trade-offs can be managed (e.g., one can

[7] Note that Chapter 1 and Chapter 4 address in detail the topics of trajectory data reduction and privacy, respectively.

guarantee a particular error-bound). Clearly, this will affect the (im)precision in the answers to the spatio-temporal queries in such MOD [15]. To couple the management of the transmission cost and storage costs, sometimes one may delegate (part of) the responsibility of spatio-temporal data reduction to the moving objects themselves, asking them to periodically transmit a historic data of their trip after applying reduction techniques [87, 144]

- In wireless sensor networks scenarios, in the case that some imprecision in time/space is acceptable, some reduction in the *(location,time)* data-items locally generated by the tracking sensors can be spared from transmission to a dedicated sink. This can significantly contribute towards saving the scarce energy resources of the nodes [146, 165], and prolong the network lifetime.
- Another application domain in which the uncertainty arises as a requirement is the location-privacy [25, 101, 166]. One of the most popular techniques – spatial cloaking – actually blurs the user's exact location into a cloaked area, satisfying some "quality threshold" – e.g., the available location information contains an uncertainty disk with area larger than the desired threshold.

We believe that the spatio-temporal uncertainty will keep on playing an important role in many application domains in the future. One challenge is coming up with a unified model of location and time uncertainties, along with corresponding query constructs and processing strategies. Many aspects of trajectory data clustering [47, 75] and warehousing [153] will inevitably require a formal treatment of uncertainty. Similarly, many applications that rely on maintaining spatio-temporal variograms [96, 151] will need to incorporate some type of uncertainty. One field that could potentially benefit from proper adaptive use of uncertainty is visualization of mobile data, both in large center displays, as well as limited resolution displays on board moving vehicles [95].

Acknowledgments: Many thanks to Reynold Cheng, Ralf Guting, Yuan-Ko Huang, Bart Kuijpers, Ralph Lange, Markus Schneider, Egemen Tanin, Yufei Tao and Stephan Winter (along with their collaborators) for generously permitting the use of some of the figures appearing in their works throughout this Chapter.

Special thanks to Hans-Peter Kriegel for his constructive suggestions.
As stated earlier, this Chapter was written with an intention of providing an overview of the topic of uncertainty in location trajectories in a manner that will balance the breadth of the coverage vs. depth of exposing the main issues in a few selected problems and solutions, as well as a tutorial-like source for both non-experts and researchers in the field. Any omissions, with sincere apologies, are sole responsibility of the author.

References

1. Abiteboul, S., Kanellakis, P.C., Grahne, G.: On the representation and querying of sets of possible worlds. In: SIGMOD Conference, pp. 34–48 (1987)
2. Aggarwal, C.C., Yu, P.S.: A survey of uncertain data algorithms and applications. IEEE Trans. Knowl. Data Eng. **21**(5), 609–623 (2009)
3. Alchourrón, C., Gärdenfors, P., Makinson, D.: On the logic of theory change. Journal of Symbolic Logic **50**, 510–530 (1985)
4. Ali, M.E., Tanin, E., Zhang, R., Kotagiri, R.: Probabilistic voronoi diagrams for processing moving nearest neighbor queries (2011). (personal communication, manuscript under revision)
5. de Almeida, V.T., Güting, R.H.: Supporting uncertainty in moving objects in network databases. In: GIS, pp. 31–40 (2005)
6. Antova, L., Koch, C., Olteanu, D.: $10^{(10^6)}$ worlds and beyond: efficient representation and processing of incomplete information. VLDB Journal **18**(5), 1021–1040 (2009)
7. Apt, K., Blair, H.: Arithmetic classification of perfect models of stratified programs. In: R. Kowalski, K. Bowen (eds.) Proc. 5^{th} International Conference and Symposium on Logic Programming, pp. 765–779. Seattle, Washington (August 15-19, 1988)
8. Beckmann, N., Kriegel, H., Schneider, R., Seeger, B.: The r* tree: An efficient and robust access method for points and rectangles. In: ACM SIGMOD (1990)
9. Benetis, R., Jensen, C.S., Karciauskas, G., Saltenis, S.: Nearest and reverse nearest neighbor queries for moving objects. VLDB Journal **15**(3), 229–249 (2006)
10. Bernecker, T., Emrich, T., Kriegel, H.P., Mamoulis, N., Renz, M., Züfle, A.: A novel probabilistic pruning approach to speed up similarity queries in uncertain databases. In: ICDE, pp. 339–350 (2011)
11. Bernecker, T., Kriegel, H.P., Mamoulis, N., Renz, M., Züfle, A.: Scalable probabilistic similarity ranking in uncertain databases. IEEE Trans. Knowl. Data Eng. **22**(9), 1234–1246 (2010)
12. Biazzo, V., Giugno, R., Lukasiewicz, T., Subrahmanian, V.S.: Temporal probabilistic object bases. IEEE Trans. Knowl. Data Eng. **15**(4), 921–939 (2003)
13. Buh, I.: Epistemic Logic in the Middle Ages. Routledge (1993)
14. Cai, Y., Hua, K., Cao, G.: Processing range-monitoring queries on heterogeneous mobile objects. In: International conference on Mobile Data Management (MDM) (2004)
15. Cao, H., Wolfson, O., Trajcevski, G.: Spatio-temporal data reduction with deterministic error bounds. VLDB Journal **15**(3) (2006)
16. Cavallo, R., Pittarelli, M.: The theory of probabilistic databases. In: VLDB, pp. 71–81 (1987)
17. Chellas, B.: Modal Logic: An Introduction. Cambridge University Press (1980)
18. Chen, J., Cheng, R., Mokbel, M.F., Chow, C.Y.: Scalable processing of snapshot and continuous nearest-neighbor queries over one-dimensional uncertain data. VLDB Journal **18**(5), 1219–1240 (2009)
19. Chen, T., Schneider, M.: Data structures and intersection algorithms for 3d spatial data types. In: GIS (2009)
20. Cheng, R., Chen, J., Mokbel, M.F., Chow, C.Y.: Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. In: ICDE (2008)
21. Cheng, R., Kalashnikov, D.V., Prabhakar, S.: Querying imprecise data in moving objects environments. IEEE-Trans. Knowl. Data Eng. **16**(9) (2004)
22. Cheng, R., Xie, X., Yiu, M.L., Chen, J., Sun, L.: Uv-diagram: A voronoi diagram for uncertain data. In: ICDE, pp. 796–807 (2010)
23. Chomicki, J.: Temporal query languages: A survey. In: ICTL, pp. 506–534 (1994)
24. Chon, H.D., Agrawal, D., Abbadi, A.E.: Range and knn query processing for moving objects in grid model. Mobile Networks and Applications **8** (2003)
25. Chow, C.Y., Mokbel, M.F., Liu, X.: Spatial cloaking for anonymous location-based services in mobile peer-to-peer environments. GeoInformatica **15**(2), 351–380 (2011)

26. Civilis, A., Jensen, C.S., Pakalnis, S.: Techniques for efficient road-network-based tracking of moving objects. IEEE Trans. Knowl. Data Eng. **17**(5) (2005)
27. of Defense, U.S.D.: Navstar gps: Global positioning system standard (2008)
28. Demiryurek, U., Pan, B., Kashani, F.B., Shahabi, C.: Towards modeling the traffic data on road networks. In: GIS-IWCTS (2009)
29. Detlovs, V., Podnieks, K.: Introduction to Mathematical Logic. University of Latvia (2011). Hyper-textbook for students
30. Ding, Z.: Utr-tree: An index structure for the full uncertain trajectories of network-constrained moving objects. In: MDM, pp. 33–40 (2008)
31. Ding, Z., Güting, R.H.: Uncertainty management for network constrained moving objects. In: DEXA, pp. 411–421 (2004)
32. Egenhofer, M., Franzosa, R.: Point set topological relations. International Journal of Geographical Information Systems **5** (1991)
33. Egenhofer, M.J., Dube, M.P.: Topological relations from metric refinements. In: GIS (2009)
34. Erwig, M., Schneider, M.: Spatio-temporal predicates. IEEE Trans. Knowl. Data Eng. **14**(4), 881–901 (2002)
35. Etzion, O., Jajodia, S., (eds.), S.S.: Temporal Databases: Research and Practice. Springer, LNCS (1998)
36. Fagin, R., Ullman, J., Vardi, M.: On the semantics of updates in databases. In: Proc. ACM PODS, pp. 352–365 (1983)
37. Gedik, B., Liu, L.: Mobieyes: A distributed location monitoring service using moving location queries. IEEE Transactions on Mobile Computing **5**(10) (2006)
38. George, B., Kim, S., Shekhar, S.: Spatio-temporal network databases and routing algorithms: A summary of results. In: SSTD (2007)
39. Ghica, O., Trajcevski, G., Zhou, F., Tamassia, R., Scheuermann, P.: Selecting tracking principals with epoch-awareness. In: Proceedings of the 18th ACM SIGSPATIAL GIS Conference, pp. 222–231 (2010)
40. Ginsberg, M.L., Smith, D.E.: Reasoning about action i: A possible worlds approach. Artif. Intell. **35**(2), 165–195 (1988)
41. Goguen, J.: The logic of inexact concepts. Synthese **19**, 325–373 (1969)
42. Goldblatt, R.: Mathematical modal logic: A history of its evolution (2006). Http://homepages.mcs.vuw.ac.nz/ rob/papers/modalhist.pdf
43. Goodchild, M.F., Zhang, J., Kyriakidis, P.C.: Discriminant models of uncertainty in nominal fields. T. GIS **13**(1), 7–23 (2009)
44. von Gottfried Wilhelm Leibniz, F.: Theodicy: Essays on the Goodness of God, the Freedom of Man, and the Origin of Evil. Open Court (1998)
45. Gowrisankar, N., Nittel, S.: Reducing uncertainty in location prediction of moving objects in road networks. In: GIScience (2002)
46. Grant, J., Horty, J., Lobo, J., Minker, J.: View updates in stratified disjunctive deductive databases. Journal of Automated Reasoning **11**, 249–267 (1993)
47. Gudmundsson, J., van Kreveld, M.J.: Computing longest duration flocks in trajectory data. In: GIS (2006)
48. Güting, R.H.: Gral: An extensible relational database system for geometric applications. In: VLDB (1989)
49. Güting, R.H.: An introduction to spatial database systems. VLDB J. **3**(4), 357–399 (1994)
50. Güting, R.H., de Almeida, V.T., Ding, Z.: Modeling and querying moving objects in networks. VLDB Journal **15**(2) (2006)
51. Güting, R.H., Behr, T., Düntgen, C.: Secondo: A platform for moving objects database research and for publishing and integrating research implementations. IEEE Data Eng. Bull. **33**(2), 56–63 (2010)
52. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N., Nardeli, E., Schneider, M., Viqueira, J.R.R.: Spatio-temporal models and languages: An approach based on data types. In: Spatio-Temporal Databases – the Chorochronos Approach (2003)

53. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N., Schneider, M., Vazirgiannis, M.: A foundation for representing and queirying moving objects. ACM TODS (2000)
54. Güting, R.H., Böhlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M.: A foundation for representing and querying moving objects. ACM Trans. Database Syst. **25**(1) (2000)
55. Güting, R.H., Schneider, M.: Realm-based spatial data types: The rose algebra. VLDB J. **4**(2), 243–286 (1995)
56. Güting, R.H., Schneider, M.: Moving Objects Databases. Morgan Kaufmann (2005)
57. Guttman, A.: R-trees: A dynamic index structure for spatial searching. In: SIGMOD Conference, pp. 47–57 (1984)
58. Haas, P.J., Suciu, D.: Special issue on uncertain and probabilistic databases. VLDB Journal **18**(5), 987–988 (2009)
59. Hadjielefteriou, M., Kollios, G., Bakalov, P., Tsotras, V.: Complex spatio-temporal pattern queries. In: VLDB (2005)
60. Hadjieleftheriou, M., Kollios, G., Tsotras, V.J., Gunopulos, D.: Efficient indexing of spatiotemporal objects. In: EDBT (2002)
61. Hägerstrand, T.: What about people in regional science? Papers of the Regional Science Association **24**, 7–21 (1970)
62. He, G., Hou, J.C.: Tracking targets with quality in wireless sensor networks. In: 13th IEEE International Conference on Network Protocols (ICNP) (2005)
63. Herrick, P.: The Many Worlds of Logic. Oxford University Press (1999)
64. Hintikka, J.: Knowledge and Belief: An Introduction to the Logic of the Two Notions. Cornell University Press (1962)
65. Hjaltason, G.R., Samet, H.: Distance browsing in spatial databases. ACM Trans. Database Syst. **24**(2), 265–318 (1999)
66. Hornsby, K., Egenhofer, M.J.: Modeling moving objects over multiple granularities. Ann. Math. Artif. Intell. **36**(1-2), 177–194 (2002)
67. Huang, Y.K., Chen, C.C., Lee, C.: Continuous k -nearest neighbor query for moving objects with uncertain velocity. GeoInformatica **13**(1) (2009)
68. Huang, Y.K., Lee, C.: Efficient evaluation of continuous spatio-temporal queries on moving objects with uncertain velocity. GeoInformatica **14**(2), 163–200 (2010)
69. Ilarri, S., Mena, E., Illarramendi, A.: Location-dependent query processing: Where we are and where we are heading. ACM Comput. Surv. **42**(3) (2010)
70. online issue, F.: (2006). Http://www.forbes.com/home/digitalentertainment/2006/04/13/google-aol-yahoo-cx_rr_0417maps.html
71. Iwerks, G.S., Samet, H., Smith, K.P.: Maintenance of k-nn and spatial join queries on continuously moving points. ACM Trans. Database Syst. **31**(2) (2006)
72. Jensen, C.S., Snodgrass, R.T.: Temporal data management. IEEE Trans. Knowl. Data Eng. **11**(1), 36–44 (1999)
73. Jeong, J., Guo, S., He, T., Du, D.: Apl: Autonomous passive localization for wireless sensors deployed in road networks. In: INFOCOM (2008)
74. Jeung, H., Liu, Q., Shen, H.T., Zhou, X.: A hybrid prediction model for moving objects. In: ICDE, pp. 70–79 (2008)
75. Jeung, H., Yiu, M.L., Zhou, X., Jensen, C.S., Shen, H.T.: Discovery of convoys in trajectory databases. PVLDB **1**(1) (2008)
76. Kainz, W., Egenhofer, M., Greasley, I.: Modeling spatial relations and operations with partially ordered sets. International Journal of Geographical Information Systems **7**(3) (1993)
77. Kanjilal, V., Liu, H., Schneider, M.: Plateau regions: An implementation concept for fuzzy regions in spatial databases and gis. In: IPMU, pp. 624–633 (2010)
78. Keogh, E.J.: A decade of progress in indexing and mining large time series databases. In: VLDB (2006)
79. Keogh, E.J., Chakrabarti, K., Mehrotra, S., Pazzani, M.J.: Locally adaptive dimensionality reduction for indexing large time series databases. In: SIGMOD Conference (2001)

80. Koubarakis, M., Sellis, T., Frank, A., Grumbach, S., Güting, R., Jensen, C., Lorentzos, N., Manolopoulos, Y., Nardelli, E., Pernici, B., Scheck, H.J., Scholl, M., Theodoulidis, B., Tryfona, N. (eds.): Spatio-Temporal Databases – the CHOROCHRONOS Approach. Springer-Verlag (2003)
81. Kripke, S.: Semantical considerations on modal logic. Acta Philosophica Fennica **16** (1963)
82. Kuijpers, B., Grimson, R., Othman, W.: An analytic solution to the alibi query in the space-time prisms model for moving object data. International Journal of Geographical Information Science **25**(2), 293–322 (2011)
83. Kuijpers, B., Miller, H.J., Neutens, T., Othman, W.: Anchor uncertainty and space-time prisms on road networks. International Journal of Geographical Information Science **24**(8), 1223–1248 (2010)
84. Kuijpers, B., Moelans, B., Othman, W., Vaisman, A.A.: Analyzing trajectories using uncertainty and background information. In: SSTD, pp. 135–152 (2009)
85. Kuijpers, B., Othman, W.: Modelling uncertainty on road networks via space-time prisms. Int.l Journal on GIS **23**(9) (2009)
86. Kuijpers, B., Othman, W.: Trajectory databases: data models, uncertainty and complete query languages. Journal of Computer and System Sciences (2009). Doi:10.1016/j.jcss.2009.10.002
87. Lange, R., Farrell, T., Dürr, F., Rothermel, K.: Remote real-time trajectory simplification. In: PerCom (2009)
88. Lange, R., Weinschrott, H., Geiger, L., Blessing, A., Dürr, F., Rothermel, K., Schütze, H.: On a generic uncertainty model for position information. In: QuaCon, pp. 76–87 (2009)
89. Lee, D.L., Zhu, M., Hu, H.: When location-based services meet databases. Mobile Information Systems **1**(2), 81–90 (2005)
90. Lema, J.A., Forlizzi, L., Güting, R.H., Nardelli, E., Schneider, M.: Algorithms for moving objects databases. Computing Journal **46**(6) (2003)
91. Li, G., Li, Y., Shu, L., Fan, P.: Cknn query processing over moving objects with uncertain speeds in road networks. In: APWeb, pp. 65–76 (2011)
92. Lian, X., 0002, L.C.: Probabilistic ranked queries in uncertain databases. In: EDBT, pp. 511–522 (2008)
93. Lin, D., Cui, B., Yang, D.: Optimizing moving queries over moving object data streams. In: DASFAA, pp. 563–575 (2007)
94. Liu, H., Schneider, M.: Querying moving objects with uncertainty in spatio-temporal databases. In: DASFAA (1), pp. 357–371 (2011)
95. Lynch, J.D., Chen, X., Hui, R.B.: A multimedia approach to visualize and interact with large scale mobile lidar data. In: ACM Multimedia, pp. 1689–1692 (2010)
96. Ma, C.: Spatio-temporal variograms and covariance models. Advances in Applied Probability **37**(3), 706–725 (2005)
97. Mamoulis, N., Cao, H., Kollios, G., Hadjieleftheriou, M., Tao, Y., Cheung, D.W.: Mining, indexing, and querying historical spatiotemporal data. In: International Conference on Knowledge Discovery and Data Mining, pp. 236–245 (2004)
98. Minker, J., Lobo, J., Rajasekar, A.: Circumscription and disjunctive logic programming. In: V. Lifschitz (ed.) Artificial Intelligence and Mathematical Theory of Computation, pp. 281–304. Academic Press (1991)
99. Mokbel, M.F., Aref, W.G.: SOLE: Scalable on-line execution of continuous queries on spatio-temporal data streams. VLDB Journal **17**(5), 971–985 (2008)
100. Mokbel, M.F., Aref, W.G.: SOLE: scalable on-line execution of continuous queries on spatio-temporal data streams. VLDB Journal **17**(5), 971–995 (2008)
101. Mokbel, M.F., Chow, C.Y., Aref, W.G.: The new casper: Query processing for location services without compromising privacy. In: VLDB (2006)
102. Mokbel, M.F., Xing, X., Hammad, M., Aref, W.G.: Continuous query processing of spatio-temoral data streams in place. The GeoInformatica Journal **9**(4) (2005)
103. Mokhtar, H., Su, J.: Questo: A query language for uncertain and exact spatio-temporal objects. In: ADBIS, pp. 184–198 (2008)

104. Mokhtar, H., Su, J., Ibarra, O.: On moving objects queries. In: PODS (2002)
105. Mouratidis, K., Yiu, M.L., Papadias, D., Mamoulis, N.: Continuous nearest neighbor monitoring in road networks. In: VLDB, pp. 43–54 (2006)
106. du Mouza, C., Rigaux, R.: Multi-scale classification of moving objects trajectories. In: SSDBM (2004)
107. O'Rourke, J.: Computational Geometry in C. Cambridge University Press (2000)
108. Othman, W.: Uncertainty management in trajectory databases. Phd thesis, Hasselt University (Belgium) (2009)
109. Pasquale, A.D., Forlizzi, L., Jensen, C.S., Manolopoulos, Y., Nardelli, E., Pfoser, D., Proietti, G., Saltenis, S., Theodoridis, Y., Tzouramanis, T.: Access methods and query processing techniques. In: Spatio-Temporal Databases: the Chorochronos Approach (2003)
110. Pattem, S., Poduri, S., Krishnamachari, B.: Energy-quality tradeoffs for target tracking in wireless sensor networks. In: IPSN (2003)
111. Pei, J., Hua, M., Tao, Y., Lin, X.: Query answering techniques on uncertain and probabilistic data: tutorial summary. In: ACM SIGMOD (2008)
112. Pelanis, M., Saltenis, S., Jensen, C.S.: Indexing the past, present, and anticipated future positions of moving objects. ACM Trans. Database Syst. **31**(1) (2006)
113. Pfoser, D., Jensen, C.S.: Capturing the uncertainty of moving objects representation. In: SSD (1999)
114. Pfoser, D., Tryfona, N.: Capturing fuzziness and uncertainty of spatiotemporal objects. In: ADBIS, pp. 112–126 (2001)
115. Pfoser, D., Tryfona, N., Jensen, C.S.: Indeterminacy and spatiotemporal data: Basic definitions and case study. GeoInformatica **9**(3) (2005)
116. Popivanov, I., Miller, R.J.: Similarity search over time-series data using wavelets. In: ICDE (2002)
117. Prabhakar, S., Xia, Y., Khalashnikov, D., Aref, W., Hambrusch, S.: Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects. IEEE - TRANS. KNOWL. DATA ENG. **51**(10) (2002)
118. Qi, Y., Jain, R., Singh, S., Prabhakar, S.: Threshold query optimization for uncertain data. In: SIGMOD Conference, pp. 315–326 (2010)
119. Raptopoulou, K., Papadopoulos, A., Manolopoulos, Y.: Fast nearest-neighbor query processing in moving-object databases. GeoInformatica **7**(2), 113–137 (2003)
120. Renz, M., Cheng, R., Kriegel, H.P., Züfle, A., Bernecker, T.: Similarity search and mining in uncertain databases (tutorial). PVLDB **3**(2), 1653–1654 (2010)
121. Reynolds, A., Ioacono, G.L.: On the simulation of particle trajectories in turbulent flows. Physics of Fluids **16**(12) (2004)
122. Rigaux, P., Scholl, M., Voisard, A.: Introduction to Spatial Databases: Applications to GIS. Morgan Kauffmann (2000)
123. Roberts, F.: Tolerance geometry. Notre Dame Journal of Formal Logic **14**(1), 68–76 (1973)
124. Roussopoulos, N., Kelley, S., Vincent, F.: Nearest neighbor queries. In: SIGMOD Conference, pp. 71–79 (1995)
125. Sarma, A.D., Benjelloun, O., Halevy, A.Y., Nabar, S.U., Widom, J.: Representing uncertain data: models, properties, and algorithms. VLDB Journal **18**(5), 989–1019 (2009)
126. Schiller, J., (editors), A.V.: Location-Based Services. Morgan Kaufmann (2004)
127. Schneider, M.: Uncertainty management for spatial data in databases: Fuzzy spatial data types. In: SSD, pp. 330–351 (1999)
128. Schneider, M.: Fuzzy spatial data types for spatial uncertainty management in databases. In: Handbook of Research on Fuzzy Information Processing in Databases, pp. 490–515 (2008)
129. Shahabi, C., Kolahdouzan, M.R., Sharifzadeh, M.: A road network embedding technique for k-nearest neighbor search in moving object databases. GeoInformatica **7**(3), 255–273 (2003)
130. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice Hall (2003)
131. Sistla, A.P., Wolfson, O., Chamberlain, S., Dao, S.: Modeling and querying moving objects. In: 13th Int'l Conf. on Data Engineering (ICDE) (1997)
132. Snodgrass, R.T., Böhlen, M.H., Jensen, C.S., Steiner, A.: Transitioning temporal support in tsql2 to sql3. In: Temporal Databases, Dagstuhl, pp. 150–194 (1997)

133. Soliman, M.A., Ilyas, I.F., Chang, K.C.C.: Top-k query processing in uncertain databases. In: ICDE (2007)
134. Suciu, D., Dalvi, N.N.: Foundations of probabilistic answers to queries. In: ACM SIGMOD (2005). Tutorial
135. Sundararaman, B., Buy, U., Kshemkalyani, A.D.: Clock synchronization for wireless sensor networks: a survey. Ad Hoc Networks **3**(3), 281–323 (2005)
136. Szewczyk, R., Mainwaring, A.M., Polastre, J., Anderson, J., Culler, D.E.: An analysis of a large scale habitat monitoring application. In: SenSys, pp. 214–226 (2004)
137. Tanin, E., Chen, S., Tatemura, J., Hsiung, W.P.: Monitoring moving objects using low frequency snapshots in sensor networks. In: MDM (2008)
138. Tao, Y., Faloutsos, C., Papadias, D., 0002, B.L.: Prediction and indexing of moving objects with unknown motion patterns. In: SIGMOD Conference, pp. 611–622 (2004)
139. Tao, Y., Papadias, D.: Spatial queries in dynamic environments. ACM Trans. Database Syst. **28**(2) (2003)
140. Tao, Y., Papadias, D., Shen, Q.: Continuous nearest neighbor search. In: VLDB (2002)
141. Tao, Y., Papadias, D., Sun, J.: The tpr∗-tree: An optimized spatio-temporal access method for predictive queries. In: VLDB (2003)
142. Tao, Y., Xiao, X., Cheng, R.: Range search on multidimensional uncertain data. ACM Trans. Database Syst. **32**(3), 15 (2007)
143. Tøssebro, E., Nygård, M.: Uncertainty in spatiotemporal databases. In: ADVIS, pp. 43–53 (2002)
144. Trajcevski, G., Cao, H., Scheuermann, P., Wolfson, O., Vaccaro, D.: On-line data reduction and the quality of history in moving objects databases. In: MobiDE, pp. 19–26 (2006)
145. Trajcevski, G., Choudhary, A., Wolfson, O., Li, Y., Li, G.: Uncertain range queries for necklaces. In: MDM (2010)
146. Trajcevski, G., Ghica, O.C., Scheuermann, P.: Tracking-based trajectory data reduction in wireless sensor networks. In: SUTC/UMC, pp. 99–106 (2010)
147. Trajcevski, G., Tamassia, R., Cruz, I., Scheuermann, P., Hartglass, D., Zamierowski, C.: Ranking continuous nearest neighbors for uncertain trajectories. VLDB Journal (2011). (accepted, to appear)
148. Trajcevski, G., Tamassia, R., Ding, H., Scheuermann, P., Cruz, I.F.: Continuous probabilistic nearest-neighbor queries for uncertain trajectories. In: EDBT (2009)
149. Trajcevski, G., Wolfson, O., Hinrichs, K., Chamberlain, S.: Managing uncertainty in moving objects databases. ACM Trans. Database Syst. **29**(3) (2004)
150. Trajcevski, G., Wolfson, O., Zhang, F., Chamberlain, S.: The geometry of uncertainty in moving objects databases. In: International Conference on Extending Database Technology (EDBT) (2002)
151. Umer, M., Kulik, L., Tanin, E.: Spatial interpolation in wireless sensor networks: localized algorithms for variogram modeling and kriging. GeoInformatica **14**(1), 101–134 (2010)
152. UNISYS: 2009 hurrican/tropical data for atlantic. http://weather.unisys.com/hurricane/atlantic/2009/index.htm (2009)
153. Vaisman, A.A., Zimányi, E.: What is spatio-temporal data warehousing? In: DaWaK, pp. 9–23 (2009)
154. Vazirgiannis, M., Wolfson, O.: A spatiotemporal model and language for moving objects on road networks. In: SSTD (2001)
155. Vlachos, M., Hadjielefteriou, M., Gunopulos, D., Keogh, E.: Indexing multidimensional time-series. VLDB Journal **15**(1) (2006)
156. Weibel, R.: Generalization of spatial data: Principles and selected algorithms. In: Algorithmic Foundations of Geographic Information Systems. LNCS Springer Verlag (1996)
157. Wilke, G., Frank, A.F.: Tolerance geometry – euclid's first postulate for points and lines with extension. In: Proceedings of the 18th ACM SIGSPATIAL GIS Conference, pp. 162–171 (2010)
158. Winslett, M.: Reasoning about action using a possible model approach. In: Proc. of the Seventh National Conference on Artificial Intelligence, pp. 89–93 (1988)

159. Winter, S., Yin, Z.C.: Directed movements in probabilistic time geography. International Journal of Geographical Information Science **24**(9) (2010)
160. Winter, S., Yin, Z.C.: The elements of probabilistic time geography. GeoInformatica **15**(3), 417–434 (2011)
161. Wolfson, O., Chamberlain, S., Dao, S., Jiang, L., Mendez, G.: Cost and imprecision in modeling the position of moving objects. In: 14 Int'l Conf. on Data Engineering (ICDE) (1998)
162. Wolfson, O., Sistla, A.P., Chamberlain, S., Yesha, Y.: Updating and querying databases that track mobile units. Distributed and Parallel Databases **7** (1999)
163. Wolfson, O., Xu, B., Chamberlain, S., Jiang, L.: Moving objects databases: Issues and solutions. In: SSDB (1999)
164. Xing, X., Mokbel, M.F., Aref, W.G., Hambrusch, S.E., Prabhakar, S.: Scalable spatio-temporal continuous query processing for location-aware services. In: International Conference on Scientific and Statistical Database Management (SSDBM) (2004)
165. Xu, Y., Lee, W.C.: Compressing moving object trajectory in wireless sensor networks. IJDSN **3**(2) (2007)
166. Yang, J., Hu, M.: Trajpattern: Mining sequential patterns from imprecise trajectories of mobile objects. In: International Conference on Extending Database Technology, pp. 664–681 (2006)
167. Yi, K., Li, F., Kollios, G., Srivastava, D.: Efficient processing of top-k queries in uncertain databases. In: ICDE, pp. 1406–1408 (2008)
168. Yiu, M.L., Mamoulis, N., Papadias, D.: Aggregate nearest neighbor queries in road networks. IEEE Trans. Knowl. Data Eng. **17**(6) (2005)
169. Yoon, H., Shahabi, C.: Robust time-referenced segmentation of moving object trajectories. In: ICDM (2008)
170. Yu, B., Kim, S.H., Alkobaisi, S., Bae, W.D., Bailey, T.: The tornado model: Uncertainty model for continuously changing data. In: DASFAA, pp. 624–636 (2007)
171. Yu, X., Mehrotra, S.: Capturing uncertainty in spatial queries over imprecise data. In: DEXA, pp. 192–201 (2003)
172. Yu, X., Pu, K.Q., Koudas, N.: Monitoring k-nearest neighbor queries over moving objects. In: ICDE, pp. 631–642 (2005)
173. Zadeh, L.A.: Commonsense knowledge representation based on fuzzy logic. IEEE Computer **16**(10), 61–65 (1983)
174. Zadeh, L.A.: Granular computing - computing with uncertain, imprecise and partially true data. In: Int.l Symp. on Spatial Data Quality (2007)
175. Zhang, Q., Sobelman, G.E., He, T.: Gradient-based target localization in robotic sensor networks. Pervasive and Mobile Computing **5**(1) (2009)
176. Zhang, Y., Cheng, R., Chen, J.: Evaluating continuous probabilistic queries over imprecise sensor data. In: DASFAA (1), pp. 535–549 (2010)
177. Zhao, Y., Aggarwal, C.C., Yu, P.S.: On wavelet decomposition of uncertain time series data sets. In: CIKM, pp. 129–138 (2010)
178. Zheng, K., Trajcevski, G., Zhou, X., Scheuermann, P.: Probabilistic range queries for uncertain trajectories on road networks. In: EDBT, pp. 283–294 (2011)
179. Zhong, Z., Zhu, T., Wang, D., He, T.: Tracking with unreliable node sequences. In: INFOCOM, pp. 1215–1223 (2009)
180. Zhu, X., Sarkar, R., Gao, J., Mitchell, J.S.B.: Light-weight contour tracking in wireless sensor networks. In: INFOCOM, pp. 1175–1183 (2008)