Geoinformatica (2016) 20:179–212
DOI 10.1007/s10707-015-0238-6

Towards fusing uncertain location data from heterogeneous sources

Bing Zhang[1] · **Goce Trajcevski**[1] · **Liu Liu**[1]

Received: 3 February 2015 / Revised: 26 July 2015 /
Accepted: 29 September 2015 / Published online: 6 November 2015

Abstract Properly incorporating location-uncertainties – which is, fully considering their impact when processing queries of interest – is a paramount in any application dealing with spatio-temporal data. Typically, the location-uncertainty is a consequence of the fact that objects cannot be tracked continuously and the inherent imprecision of localization devices. Although there is a large body of works tackling various aspects of efficient management of uncertainty in spatio-temporal data – the settings consider homogeneous localization devices, e.g., either a Global Positioning System (GPS), or different sensors (roadside, indoor, etc.).In this work, we take a first step towards combining the uncertain location data – i.e., fusing the uncertainty of moving objects location – obtained from both GPS devices *and* roadside sensors. We develop a formal model for capturing the whereabouts in time in this setting and propose the *Fused Bead* (FB) model, extending the bead model based solely on GPS locations. We also present algorithms for answering traditional spatio-temporal range queries, as well as a special variant pertaining to objects locations with respect to lanes on road segments – augmenting the conventional graph based road network with the *width* attribute. In addition, pruning techniques are proposed in order to expedite the query processing. We evaluated the benefits of the proposed approach on both real (Beijing taxi) and synthetic (generated from a customized trajectory generator) data. Our experiments demonstrate that the proposed method of fusing the uncertainties may eliminate up to 26 % of the

Research Supported by the NSF grant III 1213038.

Research Supported by the NSF grants CNS 0910952 and III 1213038, and ONR grant N00014-14-10215.

✉ Bing Zhang
bing@u.northwestern.edu

Goce Trajcevski
goce@eecs.northwestern.edu

Liu Liu
leoliu@u.northwestern.edu

[1] Northwestern University, 2145 Sheridan Road, Evanston, IL 60201, USA

false positives in the Beijing taxi data, and up to 40 % of the false positives in the larger synthetic dataset, when compared to using the traditional bead uncertainty models.

Keywords Uncertainty fusion · Roadside sensors · Beads

1 Introduction

Many applications relying on some forms of Location Based Services (LBS) [39] depend on efficient techniques for storing, retrieving and querying data which describes the whereabouts-in-time of moving entities. Traditionally, such topics are studied in the field of Moving Objects Databases (MOD) [15], and the impacts of the effectiveness of those techniques are of an extreme importance in many applications of high societal relevance such as transportation and traffic management [6, 7, 12, 44], disaster remediation [23] and location-aware social networking [1]. Especially so since, due to the advances in networking and miniaturization of the various GPS-enabled devices, the volume of location-in-time data exceeds the order of Peta-Bytes per year just from smartphones [32].

Typically, the location of a given moving object at a particular time instant is obtained either by some GPS (Global Positioning System) based devices [41, 51], or by some type of a road-side sensor – e.g., lane level positioning [9, 21]. Such sensed location data may be further combined with data from different on-board sensing devices – e.g., U.S. Xpress gathers 900 to 970 data elements of various engine/component readings [29].

Due to the inherent imprecision of the sensing devices – be it on-board GPS or other – typically there is a degree of *uncertainty* associated with the measurements of the location of a given moving object at a particular time instant. The problem of capturing the impact of the location uncertainty into the spatio-temporal data models [27] as well adding proper syntactic constructs to capture its impact on the MOD queries and the respective processing algorithms has been recognized and tackled by several earlier works [8, 14, 15, 27, 36, 48, 49].

At the heart of the motivation for this work is the observation that the state of the art – to the best of our knowledge – has not provided any models and algorithmic approaches that would *combine (i.e., fuse) uncertain location data from two different sources*. Specifically, we take a first step towards fusing the uncertain location data from on-board GPS devices and road-side sensors. We demonstrate that properly considering the joint impact of the uncertainties from both sources can eliminate portion of the moving objects (trajectories) from the answer-set. In other words, what may have been considered an answer under the single (e.g., GPS) source, may become a false-positive after fusing the two location uncertainties. As an example, consider the following query:

Q1: *Retrieve all the vehicles which have crossed the lane in road segment RS1 when driving less than 50km/h and carrying less than 80% of the maximum load.*

Clearly, given the imprecision of the location measurements, **Q1** needs to be re-phrased so that it incorporates uncertainty:

Q1^u: *Retrieve all the vehicles which have had* $> \Theta$ *(*$0 < \Theta \leq 1$*) probability of crossing the lane in road segment RS1 when driving less than 50km/h and carrying less than 80% of the maximum load.*

The answers to such, so called, lane-crossing queries play an important role in applications related to efficient traffic management [5, 18, 41] for the purpose of regulating the regime of traffic lights [21, 31].

The main contribution of this work can be summarized as follows:

- We propose a novel model of spatio-temporal uncertainty for moving objects, which combines the location data obtained by GPS devices on-board moving objects and the location data obtained from road-side sensors. We also report our preliminary experimental observations, demonstrating the reduction of false positives from the answers to certain spatio-temporal queries.
- We discuss the semantic implications of the model, in terms of the basic *where_at* and *when_at* location-in-time (whereabouts) queries, and we present algorithms for processing *lane-crossing* queries (exemplified by $\mathbf{Q1}^u$ above) and basic *range* queries.
- We present experimental observations which quantify the benefits of fusing the two uncertainties for lane-crossing and range queries in terms of the percentage of trajectories which are pruned from the answer-sets when compared to using the traditional bead-model of uncertainty for GPS-based location data.

We note that an earlier version of a subset of the results in this work was presented in [56]. The present article extends [56] by providing a deeper analysis of the proposed model; presenting a new theorem regarding the relationship of the new fusion-based model in comparison to the existing models [27]; introducing the algorithms for processing the continuous queries over the new model; and an extended set of experimental results.

The rest of this article is structured as follows. In Section 2 we recollect some backgrounds in terms of modeling spatio-temporal uncertainty, and introduce the basic terminology used in the rest of the work. Section 3 presents the details of the new uncertainty model, along with the semantics of the basic whereabouts queries along with lane-crossing and range queries. Section 6 describes our experimental observations. In Section 7 we compare our work with related literature, and we summarize and outline directions for future work in Section 8.

2 Preliminaries

We now present an overview of some of the techniques for obtaining location data, which we assume and rely upon in this work. Specifically, we discuss the main features of road-side sensors and GPS devices. Subsequently, we proceed with introducing the basic terminology and notation used in the rest of the paper.

2.1 Road-side sensors

Starting in the 1920s, when the traffic signals were still manually controlled, several generations of sensor types have been developed and deployed along road segments in various states – all for the purpose of more efficient traffic management. The types of such sensors vary from the older pressure-sensitive ones introduced in 1931, to more modern laser-based sensors sensors [50] and quite a few different types have been commercialized and used in day-to-day practical settings. For example, the AMR sensor [18] developed by Honeywell is a type of magnetic sensor with low cost. The WiEye [10] is a passive infrared sensor that can be installed on top of motes to sense road condition. The variation of sensing technologies may affect the manner of how a motion is modeled, in order to capitalize on the capabilities of a particular type of sensor. In this paper, the data model for roadside sensor that we adopt is based on TruSense T-Series, manufactured by Laser Technology Inc. [28] – a kind of active infrared sensor with a very accuracy as well as a high sampling rate.

Table 1 Comparison among different types of sensors

Sensor technology	Count	Presence	Speed	Output Data	Classification	Multiple lane detection
Inductive loop	✓	✓	✓	✓	✓	x
Magnetometer (two axis fluxgate)	✓	✓	✓	✓	x	x
Magnetic induction coil	✓	✓	✓	✓	x	x
Microwave radar	✓	✓	✓	✓	✓	✓
Active infrared	✓	✓	✓	✓	✓	✓
Passive infrared	✓	✓	✓	✓	x	x
Ultrasonic	✓	✓	✓	✓	x	x
Acoustic array	✓	✓	✓	✓	x	✓
Video image processor	✓	✓	✓	✓	✓	✓

Table 1 provides a summary of features of several different types of roadside sensors [50]. As can be seen, all of the popular and commercially available types can detect the presence and speed of vehicles, as well as provide a count value for the number of vehicles that have been detected in their sensing range. However, very few types provide more detailed sensing capabilities, such as classification and multiple lanes detection. We note that, unlike the GPS-based data, the location-in-time information obtained from the roadside sensors has not been exploited extensively in MOD context.

2.2 GPS-based spatio-temporal uncertainty

As commonly done in the literature [15], in this paper the trajectory is defined as:

Definition 1 A *trajectory* Tr_i of a moving object with a unique identifier (*oID*) is a sequence of triplets $Tr_{oID} = [(L_1, t_1), (L_2, t_2), v_{max1}] \ldots, [(L_{n-1}, t_{n-1}), (L_n, t_n), v_{max\_(n-1)}]$ where each $L_i = (x_i, y_i)$ is a point in 2D space in a corresponding reference coordinate system, and t_i denotes the time instant at which the object was at location L_i. When it comes to the time-values, $i < j$ implies $t_i < t_j$, and v_{max_i} denotes the maximum speed of the object between samples at t_i and t_{i+1}.

Given the possibility of errors in the discrete location samples (e.g., due to the imprecision of the GPS devices), plus the fact that one attempts to model a continuous phenomenon (motion, in this case) with a discrete set – uncertainty becomes an inevitable component of the model. The problem of incorporating the location uncertainty into the syntax and the respective algorithms for calculating the queries answers has been treated from a couple perspective in the MOD literature [15, 49].

One approach for modeling spatio-temporal uncertainty of moving objects is the, so called, sheared cylinder model [49]. The main assumption is that at any time instant t_i, the object's location is inside a given disk with a fixed radius, centered at the expected location at t_i. For time values different from sampling ones, the expected location is obtained via linear interpolation [49]. This model assumes a fully-known trajectory is geared towards processing continuous queries over past/historic trajectories.

The implications of the fact that the object's motion was bound by some v_{max} in-between two consecutive location updates was analyzed in [35]. Based on the definition as a

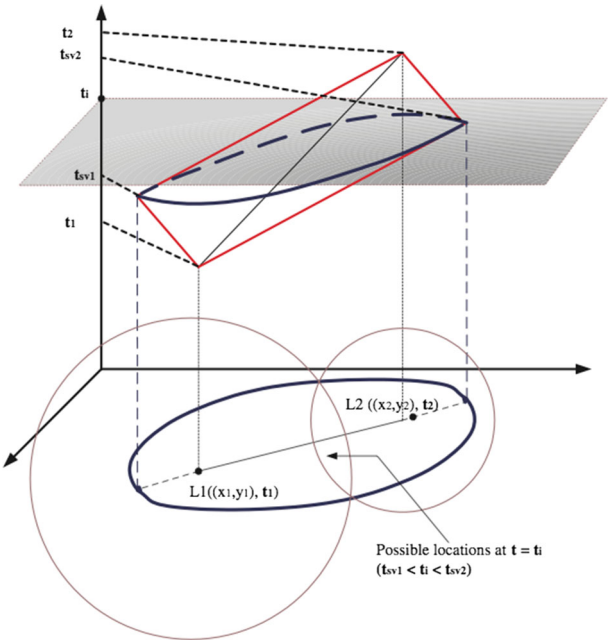


Fig. 1 Bead and ellipse model

geometric set of 2D points, it was demonstrated that the possible whereabouts are bound by an ellipse, with foci at the respective point-locations of the consecutive samples. Subsequently, [19] presented a spatio-temporal version of the model, naming the volume in-between two update points a *bead*, and the entire uncertain trajectory, a *necklace*. This model was actually introduced as a *space-time prism* in the geography literature [16]. However, the first work to present a formal analysis of the properties of the bead are [27]. An illustration is provided in Fig. 1. Letting $d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$ denote the distance between the starting location (at t_1) and ending location (at t_2), the equation of the projected ellipse (cf. [35]) is:

$$\frac{(2x - x_1 - x_2)^2}{v_{max}^2 (t_2 - t_1)^2} \frac{(2y - y_1 - y_2)^2}{v_{max}^2 (t_2 - t_1)^2 - (x_2 - x_1)^2 - (y_2 - y_1)^2} = 1 \tag{1}$$

The corresponding bead (equivalently, space-time prism) is specified with the following constraints:

$$\begin{cases} t_i \leq t \leq t_{i+1} \\ (x - x_i)^2 + (y - y_i)^2 \leq \left[(t - t_i) v_{max}^i\right]^2 \\ (x - x_{i+1})^2 + (y - y_{i+1})^2 \leq \left[(t_{i+1} - t) v_{max}^i\right]^2 \end{cases} \tag{2}$$

where v_{max} is the maximal speed that the object can take between t_i and t_{i+1}. We note that, what is commonly called *expected* speed in the case of crisp trajectories, now becomes *minimal* expected speed in-between the updates/samples. As shown in Fig. 1, at any time instant t between two consecutive samples, the possible locations of the objects are bound by the lens – i.e., intersection of two circles centered at the respective foci and with respective radii $v_{max}(t - t_1)$ and $v_{max}(t_2 - t)$.

In similar spirit to [47, 49] we can define a *possible trajectory* to be any trajectory which has its starting point and its ending point coinciding with the foci, and is fully contained inside the given bead.

2.3 Trajectories and road networks

If the objects are constrained to move along a road network, then one would expect that the corresponding space-time prisms would somehow be restricted as volumes. Specifically, if the segments of the road network are assumed to be edges in a graph, then the prisms become restricted to 2D planar figures (c.f. [11]).

In this work, we define a road network as an *augmented graph* $G = (P, E_{RS})$ where $P = \{p_1, p_2, \ldots, p_n\}$ denotes a set of points (commonly corresponding to intersections) and $E_{RS} = \{r_{S1}, ..., r_{Sk}\}$ is a collection of triplets of the form $r_{Si} = (e_i, w_{ei}, v_{ei})$ where:

- $e_i = (p_{i1}, p_{i2})$ $(\in P \text{ X } P)$ is a "regular edge" (i.e., a link between two connected vertices)
- w_{ei} denotes the width of the road segment associated with the edge e_i.
- v_{ei} denotes the maximum speed associated with r_{Si}.

We assume that the maximum speed in-between two consecutive location samples along a particular road segment corresponds to the speed-limit of that segment. Geometrically speaking, the collection of all the r_{Si}'s is the boundary of the Minkowski sum of each "regular edge" e_i and a disk with diameter w_{ei}.

We also assume the existence of a collection of sensors $S = \{s_1, s_2, ..., s_m\}$, where each sensor s_j is located at a point along the outer boundary of some road segment r_{Si}. Each s_j detects when (i.e., the time instant at which) a moving object crosses the line segment going through its location and perpendicular to e_i. The concepts are illustrated in Fig. 2.

3 Modeling the uncertainties fusion

We now discuss the details of the new uncertainty model resulting from combining the GPS-based location data and the location data generated by road-side sensors.

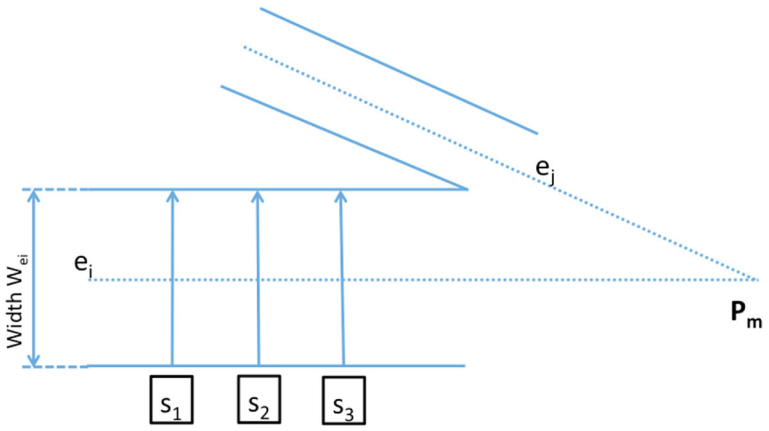


Fig. 2 Road segments and sensors

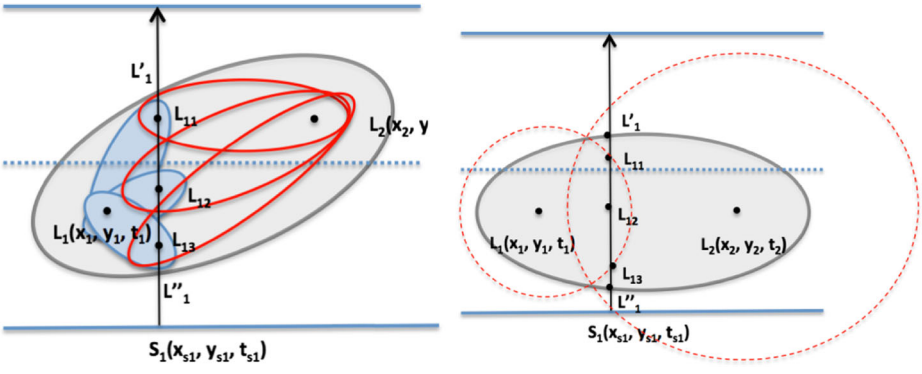
The main observation is that the road-side sensors provide additional constraints on the possible whereabouts in-between two consecutive GPS-based samples (and vice-versa). More specifically, recall that the "traditional" bead (i.e., space-time prism) was defined by the system of inequalities (2) (cf. Section 2). In addition to those inequalities, we now have the constraint that at a particular time instant t_{si}, the possible locations of a particular moving object detected by the roadside sensor are also known to be along a given line-segment determined by:

1. the location of the corresponding road-side sensor, and
2. the direction which is perpendicular to the (boundaries of the) road segment.

This can be formalized as:

$$\begin{cases} t_i \leqslant t \leqslant t_{i+1}, \\ (x - x_i)^2 + (y - y_i)^2 \leqslant (t - t_i)^2 v_{max}^2, \\ (x - x_{i+1})^2 + (y - y_{i+1})^2 \leqslant (t_{i+1} - t)^2 v_{max}^2, \\ y = m_i x + b_i, \text{ when } t = t_{si} \\ t_i \leqslant t_{si} \leqslant t_{i+1}. \end{cases} \tag{3}$$

An illustration of the system of constraints (3) is given in Fig. 3: Specifically, as shown in Fig. 3a, the original GPS-based locations L_1 and L_2 would yield a 2D projection which is an ellipse having them as foci (light-grey shaded shape in Fig. 3a) – denote it El_1. Due to the road-side sensor, the possible locations of the moving object at t_{s1} can only be along the *portion* of line segment originating in (x_{s1}, y_{s1}), perpendicular to the boundaries of the road segment, and intersecting the corresponding lens of El_1 – i.e., along the portion of the line segment $\overline{L_1' L_1''}$. Clearly, that intersection has an uncountably many points, and we show 3 such points in Fig. 3a: L_{11}, L_{12} and L_{13}. Each such point, in turn, can be used as a "generator" for two more space-time prisms: one originating in L_1, and the other terminating at L_2. The corresponding 2D projections (ellipses) are shown in Fig. 3a for L_{11}, L_{12} and L_{13}. The most important implication is that when combining the original ellipse El_1 with the uncountably infinite collection of the ellipses with one of the foci along the line segment due to the road-side sensors, the additional constraint induces a significant amount of a "dead-space" in El_1. A more detailed illustration of the valid range for selecting the points that will generate the infinite collection of (pairs of) new beads is given in Fig. 3b. Recall that at any t_{s1} between the sampling times t_1 and t_2, the object can be located inside of the



(a) GPS + Roadside sensors (b) Determining boundaries

Fig. 3 Fusing GPS and roadside sensors data

lens obtained as the intersection of the circles with radii $v_{max}(t_{s1} - t_1)$ and $v_{max}(t_2 - t_{s1})$. Hence, although the ray emanating from the roadside sensor s_1 would intersect the "global boundary" (i.e., the ellipse which is the projection of the bead) at L'_1 and L'_1, the only valid points to be considered as possible whereabouts are the ones along (and inside) the lens. As shown in Fig. 3b, those are the points along the line segment bounded by L_{11} and L_{13}.

We note that there is a complementary context of having a single uncertainty source – i.e., in contrast to having GPS-based points only. Namely, if there were only the roadside sensors available, then in between two detections by consecutive sensors (say, s_1 and s_2 from Fig. 2), the whereabouts of a given object is bounded by the infinite union $\cup(El_{si,sj})$ of uncountably many ellipses for which:

1. The first focus is some point L_{s1} located on the line-segment originating at the location of s_1.
2. The second focus is some point L_{s2} located on the line-segment originating at the location of s_2;
3. The distance between L_{s1} and L_{s2} is smaller than $v_{max}(t_{s2} - t_{s1})$ (i.e., the object could travel the distance within the time-interval $[t_{s1}, t_{s2}]$ for the given speed limit).

Incorporating the GPS-based bead in this context would either amount to the case where it intersects one (or more) of the line segments originating at the respective sensors locations, or it has no intersection with any of them. In the latter case, we have a scenario in which GPS sampling frequency is higher than the sampling frequency obtained by the roadside sensors. For such settings, the possible whereabouts will be reduced to the intersection of the $\cup(El_{si,sj})$ and the bead obtained from the GPS-based samples. In the former case, the model is a generalization of the one corresponding to the scenario illustrated in Fig. 3 – in the sense that it may be possible to have intersections of the GPS-based bead with > 1 sensor lines, as illustrated in Fig. 4. In the rest of this paper, we focus on detailed discussion of the scenarios in which a bead is intersected by a line segment emanating from a single roadside sensor.

We call the spatio-temporal structure induced by combining the two uncertainty sources – GPS and roadside sensors – a *Fused Bead* (FB), and it is a sixtuple $FB((x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1}), v_{max}, t_s, m, b)$ consisting of:

– The 2 GPS-based location-in-time samples (x_i, y_i, t_i), and $(x_{i+1}, y_{i+1}, t_{i+1})$ along with the v_{max} speed bound.
– The time instant of detection of the road-side sensor.

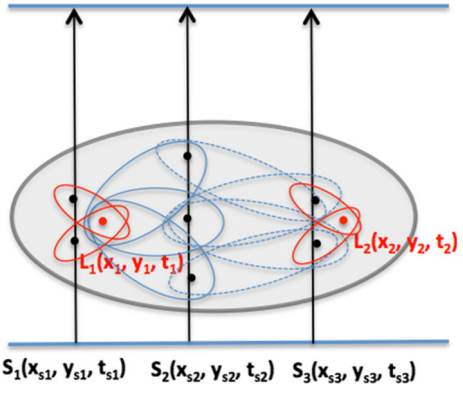


Fig. 4 Multiple roadside sensors intersecting a bead

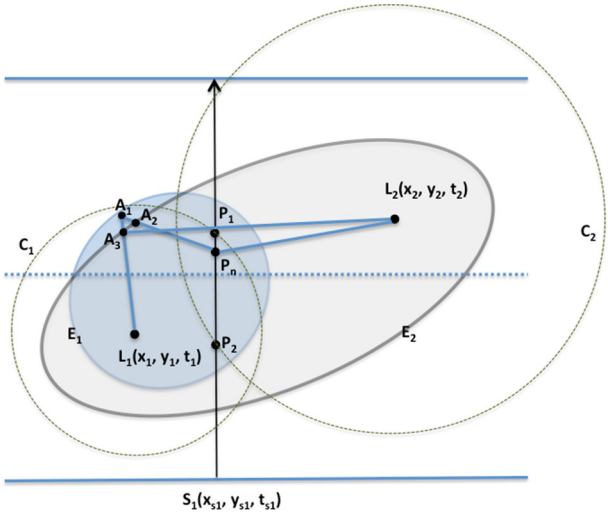


Fig. 5 Proof of fused bead containment

– The parameters of the equation $y = mx + b$ (in a given referent coordinate system) of the line specifying the corresponding line-segment emanating from the roadside sensor and specifying the locations of the possible new foci.

When it comes to bounding the possible whereabouts, an intuition may cause one expect that some of the points along the intersection of the line segment with the ellipse El_1 may yield possible focal points that would generate ellipses which are not fully contained inside El_1. However, the set of constraints in Eq. 3 will eliminate every portion which is outside the intersection of the original El_1.

We now proceed with a formal analysis of an important property of the FB model, towards which we first recall some of the properties of the bead model presented in [27]. Let $B(x_i, y_i, t_i, x_{i+1}, y_{i+1}, t_{i+1}, v_{max})$ denote[1] the bead between two location-samples (x_i, y_i) and (x_{i+1}, y_{i+1}) at respective times t_i and t_{i+1}, during which the speed is bounded by v_{max}

Property 1 Given (x_i, y_i, t_i), and $(x_{i+1}, y_{i+1}, t_{i+1})$ with $t_i < t_{i+1}$ and $v_{max} > 0$, any trajectory from (x_i, y_i, t_i) to $(x_{i+1}, y_{i+1}, t_{i+1})$ for which the speed at any moment $t_i \leq t \leq t_{i+1}$ is less than v_{max} is located within the bead $B(x_i, y_i, t_i, x_{i+1}, y_{i+1}, t_{i+1}, v_{max})$ and the projection of such a trajectory on the (x, y)-plane is located within $\pi_{x,y}(B(x_i, y_i, t_e, x_{i+1}, y_{i+1}, t_{i+1}, v_{max}))$. Furthermore, for any point (x, y, t) in $B(x_i, y_i, t_i, x_{i+1}, y_{i+1}, t_{i+1}, v_{max})$, there exists a trajectory from (x_i, y_i, t_i) to $(x_{i+1}, y_{i+1}, t_{i+1})$ which passes through (x, y, t).

Property 1 explains the bounding relationship between trajectory and bead. Taking the constrain (3) into consideration, one can deduce to the following corollary:

[1]The original notation in [27] was $B(t_i, x_i, y_i, t_{i+1}, x_{i+1}, y_{i+1}, v_{max})$ and we slightly modified it for consistency with the rest of the notation in this article.

Corollary 1 *Any trajectory from x_i, y_i, t_i to $(x_{i+1}, y_{i+1}, t_{i+1})$ which passes through a point that lies on the boundary of the ellipse*

$$\frac{(2x - x_1 - x_2)^2}{v_{max}^2(t_2 - t_1)^2} + \frac{(2y - y_1 - y_2)^2}{v_{max}^2(t_2 - t_1)^2 - (x_2 - x_1)^2 - (y_2 - y_1)^2} = 1 \tag{4}$$

is the longest possible trajectory.

In a similar spirit, and based on these properties of the bead model, we now have the following property regarding the FB model:

Lemma 1 *Any bead generated by: (1) a focal point located in the GPS-based sample, and (2) a point from the line segment P_1P_2 representing possible locations obtained via a roadside sensor, is contained within the original bead.*

Proof We prove Lemma 1 by contradiction. Assume that P_n is a point on the line segment P_1P_2 and consider the ellipse El_2 with foci P_n and L_1. Let A_1 denote a point which lies within El_2 but outside the original bead El_1, defined by the original bead (i.e., foci L_1 and L_2, and vmax bounding speed). Using Fig. 5 as an illustration, we proceed with connecting the two line segments L_1A_1 and A_1P_n. They intersect El_1 at some points, denote them A_2 and A_3. According to Corollary 1, the polyline with two segments $L_1A_3L_2$ is the longest trajectory that the object could possibly move along from L_1 to L_2. However, by assumption, A_1 is bounded to be within El_1 which, in turn, implies that $L_1A_1P_n$ is a route of a valid trajectory from L_1 to P_n and, moreover, $L_1A_1P_nL_2$ is a route of a valid trajectory from L_1 to L_2. However, since, based on the triangular inequality, $\overline{A_3A_1} + \overline{A_1P_n} > \overline{A_3P_n}$ and $\overline{A_3P_n} + \overline{P_nL_2} > \overline{A_3L_2}$, we have $\overline{A_3A_1} + \overline{A1P_n} + \overline{P_nL_2} > \overline{A_3L_2}$. Based on the last inequality, we can conclude that $\overline{L_1A_3} + \overline{A_3A_1} + \overline{A_1P_n} + \overline{P_nL_2} > \overline{L_1A_3} + \overline{A_3L_2}$, which implies that the trajectory $\overline{L_1A_1P_nL_2}$ is longer than trajectory $\overline{L_1A_3L_2}$. This, however, is a contradiction to the Corollary 1 which states that no other valid trajectory is longer than $L_1A_3L_2$, and we could conclude that assumption on the existence of point A_1 is not valid. □

Lemma 1 demonstrates that whenever there is a location sampling from a roadside sensor in-between two GPS-based location samples, the possible locations by the FB model are contained within the set of possible locations bounded by original GPS-based bead. An illustration of a FB-based segment is shown in Fig. 6, and a visual comparison with the illustration of the full GPS-based bead (cf. Fig. 1) reveals one of the consequences of Lemma 1. Another important consequence of Lemma 1 is in the conclusion that the FB will not introduce any false positives – in comparison with the traditional bead – when determining an intersection of the possible whereabouts with other (spatial, or spatio-temporal) entities.

4 Possible locations at time instants

We now proceed with elaborating some basic calculations regarding the boundary of the possible locations of a given object at a specific time instant under the FB model, as well as the time-interval during which an object can be at a particular location. Subsequently, we also discuss the methodology for detecting whether the possible locations of a moving object are part of a given (spatial) range.

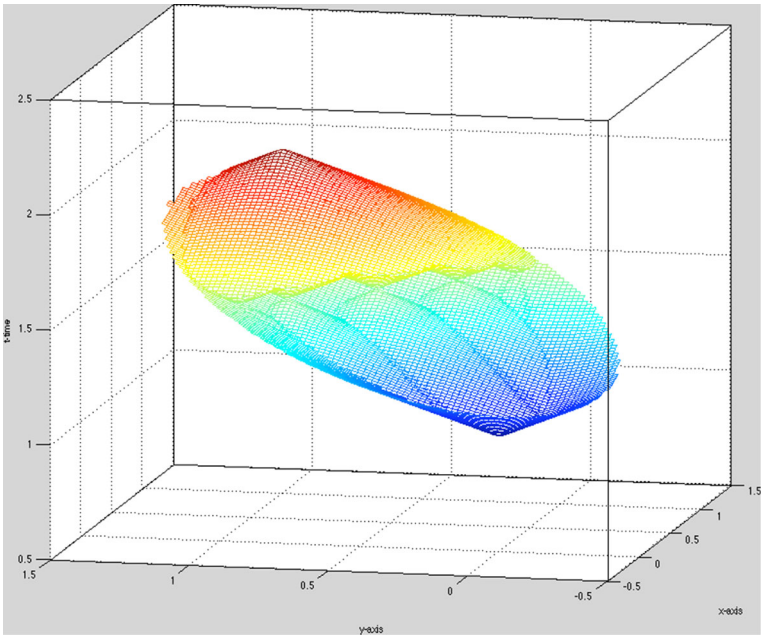


Fig. 6 Outer boundary of the fused uncertain locations

Recall that the FB model is based on the original bead obtained via GPS-based locations L_1 and L_2 (foci of a 2D ellipse El_1) and a road-side sensor providing possible locations along a line-segment perpendicular to a given road at a time instant t_s (cf. Fig. 3a).

When it comes to location whereabouts at certain time instant t_{s1}, the regular bead model has a boundary defined by a lens $L_e(t_{s1})$ which obtained as the intersection of the circles with radii $v_{max}(t_{s1} - t_1)$ and $v_{max}(t_2 - t_{s1})$, centered at L_1 and L_2 respectively (light blue

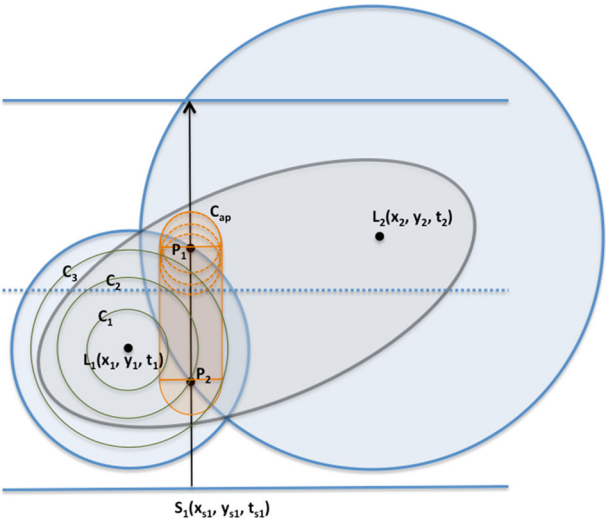


Fig. 7 Cross section of fused bead

shaded area in Fig. 7). If it happens that at that same time instant the object has been detected by a roadside sensor – then the object must be somewhere along the ray emanating from that sensors location and perpendicular to the road segment. However, because of the uncertainty boundary from the GPS-based location data, only the points along that ray which are inside the lens L_e are valid possible-locations – illustrated by the segment P_1P_2 in Fig. 7.

Let $\varepsilon \in [0, 1]$ denote a real variable. Any point $P(t_{s1}, \varepsilon) \in \overline{P_1P_2}$ which is a possible location of the object at t_{s1} has coordinates $x_{P(\varepsilon)} = \varepsilon x_{P1} + (1 - \varepsilon)x_{P2}$ and $y_{P(\varepsilon)} = \varepsilon y_{P1} + (1 - \varepsilon)y_{P2}$

With this in mind, given a time instant $t_i \in [t_1, t_{s1}]$, the possible locations of the moving object at t_i are bounded by the uncountable union of intersections between:

1. The disk centered at L_1 and with radius $v_{max}(t_i - t_1)$.
2. An infinite collection of disks, each centered at a point $P(t_{s1}, \varepsilon)$ along P_1P_2 and each with radius $v_{max}(t_s - t_i)$

In Fig. 7, the circles C_1, C_2 and C_3 are examples of the boundaries of the objects whereabouts at different time-values (t_i) due to the GPS-sample at location L_1. For a fixed value of t_i Fig. 7 also shows the boundary defined by the "envelope" of the union of the uncountably many disks centered along P_1P_2 – essentially, the sum of the line segment P_1P_2 and a disk with radius $v_{max}(t_s - t_i)$.

Depending on the time value and ε, there are five basic kinds of time-intervals during which shapes of the unions determining the object's whereabouts have distinct properties. We use the phrase *significant times* to denote the boundaries of those time-intervals.

1. $t \in [t_1, t_i^{l1})$ (*Occurrence of the first lens*): During this interval, the possible locations are inside a disk centered at L_1 – this is the case when t_i is very close to t_1 – meaning: regardless of the value of ε, each disk with radius $v_{max}(t_s - t_i)$ centered at any point along P_1, P_2, fully covers the disk centered at L_1 with radius $v_{max}(t_i - t_1)$. Let P_c and P_f denote points along P_1P_2 which is geometrically closest and farthest to L_1 respectively. Clearly, point P_f will be the one with the earliest change of this kind of containment with the disk – at some time instant t_i^{l1}, the intersection[2] will switch from a full-disk centered at L_1 into a lens defined by the intersection of the two disks: one centered at L_1 and one centered at P_f.
2. $t \in [t_i^{l1}, t_i^{l_A})$ (*from a single lens, until "lenses_All"*): During this time interval, depending on the values of ε, some of the disks centered along P_1P_2 (each with radius $v_{max}(t_{s1} - t_i)$) are still fully covering the disk centered at L_1 with radius $v_{max}(t_i - t_1)$. These are the ones whose centers are closer to P_1 (i.e., $P(t_{s1}, \varepsilon)$ with ε closer to 0).
3. $t \in [t_i^{l_A}, t_i^{d1})$ (*from lenses_All, until the first (full) disk appears*): This is the time-period during which each possible foci along P_1P_2 is a center of a disk with which yields a lens-shaped intersection with the disc centered at L_1. At the expiration of this time interval, the disk centered at P_c and with radius $v_{max}(t_s - t_i)$ is about to be fully covered by the disk centered at L_1 and with radius: $v_{max}(t_i - t_1)$
4. $t \in [t_i^{d1}, t_i^{d_A})$ (*from a single full disk appearance, until disks_All*): similarly to the 2nd case above, during this time interval some of the disks centered along P_1P_2 have a lens-shaped intersection with the disk centered at L_1, while some are fully contained inside of it.
5. $t \in [t_i^{d_A}, t_{s1})$ (*disks_All*): The last distinct time-interval for the part of the FB between the first GPS-based foci and the roadside sensor is similar to case "1" above, in the sense

[2]For clarity, we present the details of calculating t_i^{l1} and other significant times in the Appendix.

that every disk with radius $v_{max}(t_{s1} - t_i)$, regardless of where its center is located along P_1P_2, is fully contained inside the disk centered at L_1 and with radius $v_{max}(t_i - t_1)$.

We note that for time-values $t \in [t_{s1}, t_2]$, the cases are analogous (and in reverse order) from the ones specified above, in the sense that there are four significant time instants defining five distinct intervals.

Let $D_1(t)$ denote the disk centered at L_1 and with radius $v_{max}(t - t_1)$. Also, let $D_P(t, \varepsilon)$ denote the disk centered at the point $P(t_{s1}, \varepsilon)$ with radius $v_{max}(t_{s1} - t)$. For a given 2D shape S, let $A(S)$ denote its area. Assuming a uniform distribution in each time-interval between two consecutive significant times,[3] we obtain that the corresponding *pdf*s (probability density functions) are:

1. $t \in [t_1, t_i^{l1})$:
$$f(x, y, t) = \begin{cases} \frac{1}{\pi(v_{max}(t-t_1))^2} & \text{if}(x, y) \in D_1(t) \\ 0 & \text{otherwise} \end{cases}$$
2. $t \in [t_i^{l1}, t_i^{l_A})$
$$f(x, y, t) = \frac{1}{\pi(v_{max}(t - t_1))^2 + A(\cup_{\varepsilon > \delta_1(t)}(D_1(t) \cap D_P(t, \varepsilon)))}$$
where $\delta_1(t)$ is the smallest value of ε at a given t for which $D_P(t, \varepsilon) \not\subseteq D_1(t)$.
3. $t \in [t_i^{l_A}, t_i^{d1})$
$$f(x, y, t) = \frac{1}{A(\cup_{\varepsilon}(D_1(t) \cap D_P(t, \varepsilon)))}$$
4. $t \in [t_i^{d1}, t_i^{d_A})$
$$f(x, y, t) = \frac{1}{\pi(v_{max}(t_{s1} - t))^2 + A(\cup_{\varepsilon > \delta_2(t)}(D_1(t) \cap D_P(t, \varepsilon)))}$$
where $\delta_2(t)$ is the smallest value of ε at the given t for which $D_1(t) \not\subseteq D_P(t, \varepsilon)$.
5. $t \in [t_i^{d_A}, t_{s1})$
$$f(x, y, t) = \begin{cases} \frac{1}{\pi(v_{max}(t_{s1}-t))^2 + \overline{P_1P_2} \cdot (v_{max}(t_{s1}-t))} & \text{if}(\forall \varepsilon) D_P(t, \varepsilon) \subseteq D_1(t) \\ 0 & \text{otherwise} \end{cases}$$

When calculating the probability that a given moving object whose motion is modelled as an FB is inside a given spatial range at a given time instant, we need the area of the intersection. However, given the complexity of the boundary of the objects whereabouts, the calculation of overlapping area may necessitate relying on numerical integration methods.

4.1 Numerical method for complex area calculation

Selecting an approximate evaluation method, i.e., numerical method, depends on the task at hand. If we aim at calculating the intersection of two curves, the Newton-Raphson Method is the most widely used one, whereas calculating the area bounded by a given curve may rely upon Trapezoid Rule, Gaussian Quadrature Method or Monte Carlo Integration [13].

As an example, in a GPS-based bead, the location whereabouts given time instant are relatively straightforward to compute since they are either a circular disk or a lens formed by

[3]Throughout this work, we assume independence between location-values in successive location samples (cf. [4, 11]).

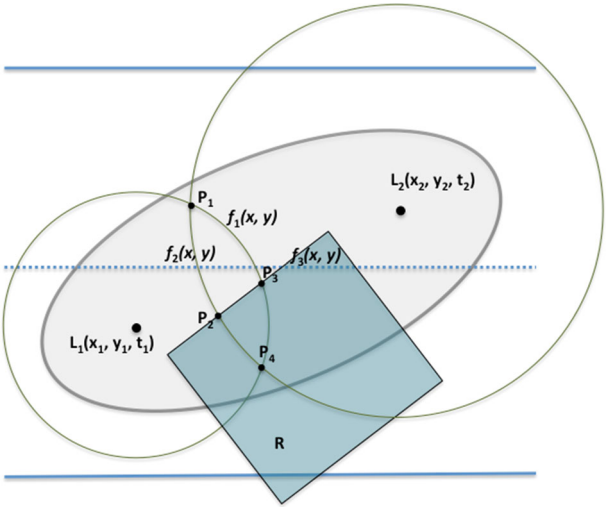


Fig. 8 Area calculation in GPS-based bead

intersection of two circles. Moreover, finding the points where the boundary of the object's whereabouts intersect a given polygon is still achievable analytically, since the possibilities amount to calculate an intersection between a circle and a line(segment) is limited, as shown with P_1 and P_4 in Fig. 8. However, even in such cases, one may need to use numerical methods for calculating the area of the intersection.

Given the complexity of the FB structure at a particular time instant, in this work we resort to approximate computations based on a spatial grid, as shown in Fig. 9. Clearly, the

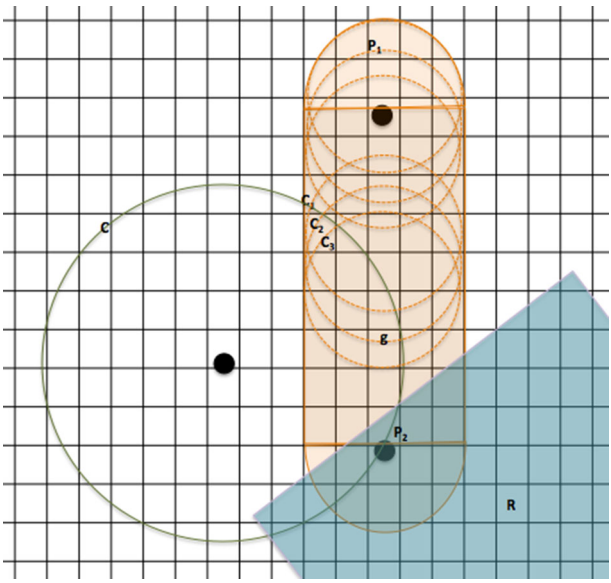


Fig. 9 Grid based numerical approximation

size of the grid cell will affect the running time of the (execution of the) corresponding algorithms. However, there is another aspect to consider – the (im)precision. By the very definition of the FB, it is a union of uncountably many (subsets of) disks. Hence, we need to discretize the number of such disks, for which a basic unit Δd is introduced, specifying the locations of the centers of the disks that will be accounted for when calculating a particular area. These impacts are analyzed in Section 6.

5 Query processing

We now turn our attention to processing spatio-temporal queries under the FB model. We start with the basic *where_at* and *when_at* location-in-time queries, followed by a range query and lane-crossing query. Lastly, we discuss the possibility of speeding up the query processing via pruning.

Without loss of generality, the presentation will use the setting of a single fused bead. However, when necessary, the issues that may arise due to considering the entire necklace will be explicitly addressed.

5.1 Basic queries

Similarly to the GPS-based bead, in order to determine the whereabouts at a given time instant t for a fused bead, we need to obtain the intersection of FB with the horizontal plane $Time = t$. The corresponding illustration of the volume in 2D space + Time, along with the 2D projection, is shown in Fig. 10. The boundary of the 2D projection is obtained as the "envelope" of the union of two collections of uncountably many intersections of disks centered along the line-segment originating at the roadside sensor, with the disk centered at L_1. The details were elaborated in Section 4.

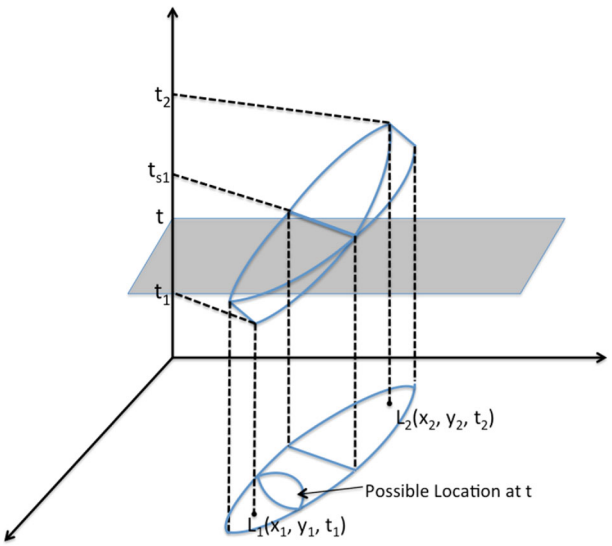


Fig. 10 Whereabouts at time instant

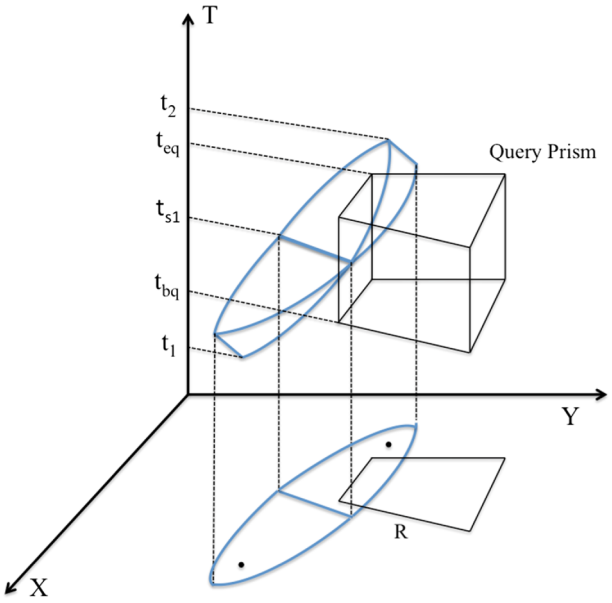


Fig. 11 Range query for a given prism

The GPS-based bead (e.g., L_1) and the other centered at a point along the intersection chord (cf. $\overline{L_{11}L_{13}}$ in Fig. 3) resulting from secant due to the roadside sensor and the arc from the lens of the GPS-based bead. Thus, one of the boundaries is always a circular arc originating at the focal point of the "original" GPS-based bead, centered at focus of the GPS-based bead (say, L_1) and with radius $v_{max}(t - t_1)$. The boundary is actually the boundary of the union of uncountably many disks with radii $v_{max}(t_{s1} - t)$, with centers along the intersection-chord (Fig. 11).

The complementary query, *when_at(oID, L)* returns the times during which it is possible for the object oID to be at the location $L(x_L, y_L)$, i.e., a time-interval $[t_{L1}, t_{L2}]$. The time-interval can be defined as the two intersections between the boundary of the fused bead FB and the vertical line (i.e., ray) emanating from L. To calculate the values, we have the following observations:

1. t_{L2} is the latest time that a circle located at the GPS-based focus from the sample at t_1 will "reach" L – hence, it can be obtained as a solution to the equation:
   $\overline{L_1 L} = v_{max}(t_{L2} - t_1)$
2. t_{L1}, on the other hand, is the earliest time that any circle with the center on the intersection chord(P_1P_2 in Fig. 7) and radius $v_{max}(t_s - t_{L1})$ would pass through L.

5.2 Range query processing

A typical spatial range query aims at retrieving the spatial (static) objects which have a particular topological relationship (e.g., inside, intersect, etc...) with a given range, which

is, an entity with spatial extent [42]. A distinct feature of spatio-temporal range queries is that they are continuous – i.e., the answer may change with time: for example, an object that was inside a given query region may subsequently exit it, and vice versa. In our settings, the key observation is that we need to take into account the uncertainty of the object's location at a given time instant when formulating the syntactic variants of the range query [43, 49].

In this work, we assume that the spatial region of interest for the range query is bounded by a simple polygon R (Fig. 11) and we also assume $[tbq, teq]$ values indicating the bounds of interest in the temporal dimension. We denote the set $\{\forall(x, y, t)|(x, y \in R$ and $t \in [t_{bq}, t_{eq}])\}$ for QP_R (query prism). Earlier works [47, 49] have provided qualitative variants regarding the domains of space and time in the sense of uncertain object being inside R: (1) *sometimes* or *always* throughout the time-interval of interest; and (2) *possibly* or *definitely* so.

For a given uncertain trajectory represented as a sequence of FBs, $Tr = [FB_1, FB_2, \ldots, FB_n]$, where each $FB_i = ((x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1}), v_{max}, t_s, m, b)$, we are interested in answering the following type of a range query:

$\mathbf{Q}^u_R$: *Does the moving object have a probability* $\geq \Theta$ *of being inside* R *at least* ϕ *of the time-interval* $[t_{bq}, t_{eq}]$.

We use the generic notation *Inside* $(FB, R, t_{bq}, t_{eq}, \theta, \phi)$ to denote the (parameterized version of the) queries like $\mathbf{Q}^u_R$, with the intended meaning $\exists\phi$ – a sum of time-intervals (not necessarily contiguous) during which the ratio of the intersection of the FB and QP_R is greater than Θ. We note that [47] proposed analytical solutions for answering existential/universal variants by verifying intersecting conditions between ellipses and circles in the traditional bead model. Thus, for example, one could verify whether *Sometime Inside* (FB, R, t_{bq}, t_{eq}) based on an existence of a time instant at which the intersection between R and FB is not empty – which corresponds to any $\Theta > 0$ in the current context. Similarly, the predicate *Always Inside* (FB, R, t_{bq}, t_{eq}) would amount to $\Theta = 1$ throughout the entire time-interval of interest for the query.

Following our discussions in Section 4, the probability *Prob(X,Y,T)* (i.e., the probability that the object is inside a region bounded by implicit curves "X", "Y" throughout a time-interval "T") is defined as triple integral on 2D+time:

$$Prob(X, Y, T) = \int_T \int_Y \int_X f(x, y, t)$$

The grid based numerical method provides an estimation regarding areas of location whereabouts at a certain time instant. Assuming a uniform *pdf* at any time instant, we have:

$$\begin{aligned} Prob(X, Y, T) &= \frac{\int_T \text{Overlapping Area between FB and } R\ (t)}{\int_T \text{Possible FB Whereabouts (t)}} \\ &= \frac{\text{Overlapping Volume between FB and } QP_R}{\text{Overall volume of FB}} \end{aligned}$$

Let $A_{FB}(t)$ denote the area of the possible whereabouts of the object at time t (i.e., the area of the region corresponding to the answer of *where_at(t)* query) and let $A(R)$ denote the area of the query region R. We have the following algorithm:

Algorithm 1 Inside $(Tr_{FB}, R, t_{bq}, t_{eq}, \theta, \phi)$

```
1: float T = 0, t_total = 0;
2: int k = 0;
3: while (t_q + k · △) < t_eq do
4:     T = T + △;
5:     if ((A_F B(T) ∩ A(R))/ A_FB)(T) ≥ θ then
6:         t_total = t_total + △
7:         if t_total ≥ φ then
8:             The trajectory satisfies the predicate;
9:             Exit;
10:        end if
11:    end if
12:    k + +;
13: end while
14: Trajectory satisfies the predicate only t_total of the [t_bq, t_eq];
```

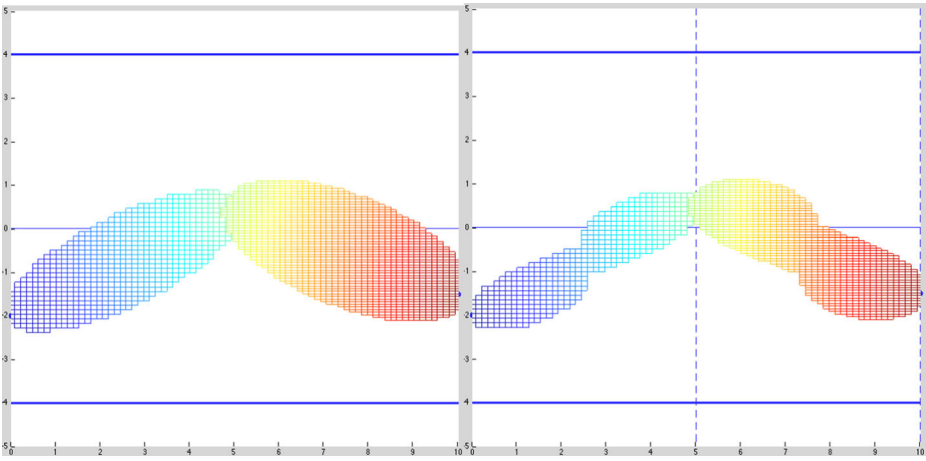
Algorithm 1, without checking the value of the "time-accumulator" and with a minor addition to sum up the values $((A_F B(T) \cap A(R))/(A_{FB})(T)) \times \triangle$, can be used to calculate the ratio of the volume (i.e., the corresponding probability) of the object being inside R. We note that, if one simply wants to calculate the probability of an object being inside QP_R, without any concerns about Θ or ϕ (i.e., overloading the argument-signature), then the "If()" test in Algorithm 1 can be eliminated, and the corresponding approximations summed up.

As our experimental results in Section 6 will illustrate, applying Algorithm 1 to process range queries over uncertain trajectories modeled with FB consistently yields fewer false positives, in comparison to the case of applying it to a collection of uncertain trajectories represented via regular beads.

5.3 Lane-crossing query processing

Lane-crossing query can be perceived as a special case of a range query where the query prism is degenerated from a polygon into a half plane. Figure 12, illustrates the lane-crossing query for the regular bead and FB models. As mentioned in Section 1, the lane-crossing query is important in applications related to fleet management and efficient traffic management. We reiterate the statement explaining such queries:
$\mathbf{Q}^{\mu}_{LC}$: *Given a fused bead* $FB(((x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1}), v_{max}, t_s, m, b))$, *does the moving object have* $> \Theta$ $(0 < \Theta \leq 1)$ *probability of crossing the lane and entering half-plane* R.

We use the generic notation *Lane-Cross* $(FB, L, t_{bq}, t_{eq}, \theta)$ to specify the corresponding predicate expressing the fact that an uncertain moving object represented via *FB* has crossed the lane L on a given road segment (cf. Section 2) with a probability $\geq \Theta$, sometime between $[t_{bq}, t_{eq}]$. If we wish to calculate the total probability of an object crossing the lane L throughout the entire time-interval of interest of the query, then we can obtain an approximate value by applying similar ideas as in Algorithm 1 – i.e., summing up the products of the intersection area with $\triangle$.



(a) Bead model (b) Fused bead model

Fig. 12 Beads and lane-crossing query

5.4 Pruning techniques

Typically, spatio-temporal query processing proceeds in three "stages" [15]:

(1) *Filtering*, where an index is used to eliminate those data items that are guaranteed not to satisfy the query [45]; followed by:
(2) *Pruning*, where some properties might be used to further reduce the set of the possible candidates for the answer, portion without introducing any false negatives;
(3) *Refinement*, where algorithmic checks and calculations are used to eliminate false positives that were not eliminated during the previous stage(s).

While the problem of efficient and effective indexing structures for processing spatio-temporal queries over the FB model is outside the scope of this work, we note that for the specific queries discussed here, there may be pruning approaches that can speed up the overall execution of the spatio-temporal queries on FB model. In the sequel we discuss few such strategies:

A. Definitely Outside – Individual Fused Bead Bounds (IBb)

Proposed in [47], this pruning strategy is designed for GPS-based bead. It approximates each GPS-based bead with its minimum bounding vertical cylinder. According to the Lemma 1, FB is bounded by GPS-based bead, which justifies its application to FB as well. In effect, the ellipse – which is the projection of a bead, formed by two GPS points belongs to FB, on (X,Y) plane, becomes a circle centered in the center of the respective ellipse, as shown in Fig. 13. The radius of the approximation-disk Ad_i is: $r(Ad_i) = 1/2(v^i_{max})(t_{i+1} - t_i)$.

B. Definitely Inside – GPS points pre-screening

This pruning technique is specially designed for lane-crossing query, where the predicate determines if it is possible for a lane-cross to occur. The technique is based on the following observation: if two consecutive GPS points are located on two different sides of the central

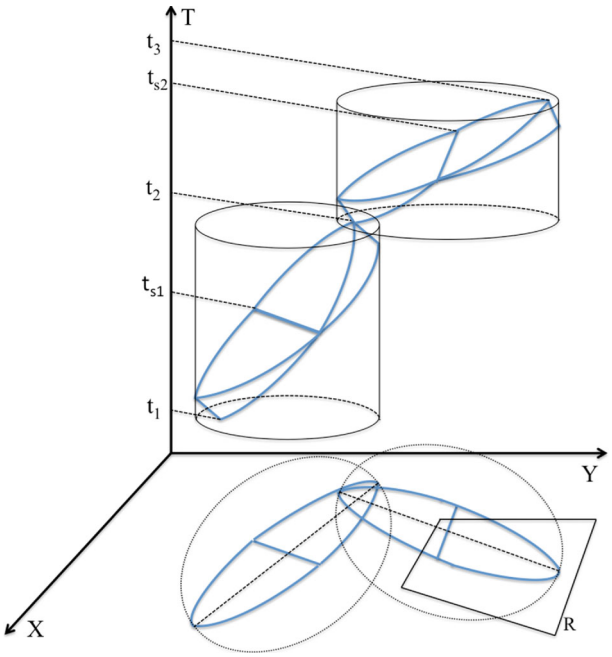


Fig. 13 Cylinder-based pruning approximation

line, there must be at least one time instant at which the moving object crosses the road, in which case we are able to prune the FB and direct return true.

C. Sometime Inside – fine grained dead-space removal

We are interested in finding the time instant(s) when uncertain trajectories enter/exit the query region R – call them *critical points*. By doing so, we eliminate some redundant time-intervals with respect to the time-bounds of a particular query. The general case for time $t \in [t_i, t_{i+1}]$ being a critical point occurs when the intersection of the uncertain region at t with a query rectangle is a single point. In the time interval $[t_i, t_s]$, the single-point-intersection between disk centered at the first GPS point and query region stands for the entering moment. Similarly, in the time interval $[t_s, t_{i+1}]$, the single-point-intersection represents exiting moment. Since the query region is represented as polygon in the (X, Y) plane, each edge of the polygon is defined as a segment of 2D line $y = ax + b$. The calculation of the critical times is presented in the Appendix.

6 Experimental observations

We now present the experimental observations regarding the traded-offs between the benefits of the FB model in terms of reducing the number of false positives in the queries' answers vs. the computational costs. More specifically, we implemented the proposed approach and tested it for lane-crossing query and range query, comparing the beads obtained using only GPS data against the FB model, and ran comprehensive experimental comparisons based on correctness, robustness and efficiency. In addition, we present two

types of pruning techniques which we applied as part of the query processing and discuss their impact.

6.1 Dataset description

In our experiments we used both synthetic and real-life datasets.

Synthetic Data The synthetic data was generated by a modified version of Brinkhoff network-based generator, representing vehicles' movements on road network. GPS points are generated on the map of Oldenburg, which are available at the Brinkhoff generator official website (http://iapg.jade-hs.de/personen/brinkhoff/generator/).

Real-life Data The real world dataset we used in our experiments is based on Beijing taxi data from the T-Drive project [54, 55]. Essentially, the Beijing road network is built based on OpenStreetMap data, containing 140207 vertices and 155997 road segments. GPS points are map-matched to road network using point-to-curve matching approach [38]. Figure 14 illustrates the map matching process, where green dots correspond to the raw GPS points, and the blue dots are the points obtained after map-matching process. To minimize the impact of the measurement errors, we filtered out the low speed GPS points (i.e., ones with speed less than 1m/s).

Following is the description of the setups that were applied in order to run the experiments for each of the queries:

1. Vehicles are allowed to move along the road network with a speed $\leq$ 50km/h.
2. We add a *width* parameter to the road network, the value of which is set to be $\leq$ 4m [40].
3. At each time instant of the object's motion along the road, its width location is generated by a python-based random generator within a given random interval based on the width parameter used in that location. The values are selected such that 0 represents the center of the road; negative values represent left lane; and positive values represent the right lane – with respect to the direction of the object's motion.

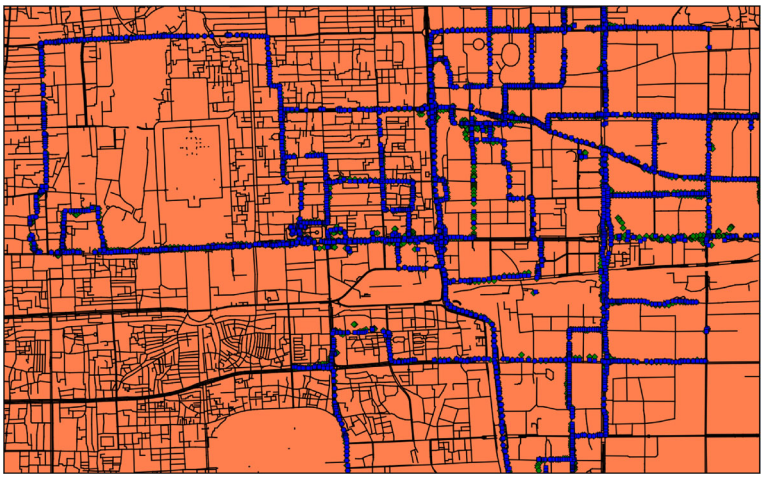


Fig. 14 Beijing road network and taxi dataset

4. We apply additional post-processing to the trajectories by adding roadside sensor data. As mentioned, for a given location on the right (i.e., in the direction of object's motion) side of the road, we generate a ray perpendicular to the road's boundary.
5. To cater to the variations of the speed, we vary the actual time at which the moving object crosses the ray corresponding to a particular roadside sensor. Given a bead $B(x_i, y_i, t_i, x_{i+1}, y_{i+1}, v_{max})$ and the ray $y = mx + b$ from a given roadside sensor, we calculate the time interval during which a moving object can cross the (ray generated at the) location of the sensor as:

$$[T_{smin}, T_{smax}] = \left[t_i + \frac{\text{distance from } (x_i, y_i) \text{ to sensor}}{v_{max}}, t_{i+1} - \frac{\text{distance from } (x_{i+1}, y_{i+1}) \text{ to sensor}}{v_{max}} \right]$$

Then we calculate the sensor time t_s following normal distribution between T_{smin} and T_{smax} with average $\mu = (T_{smin} + T_{smax})/2$ and standard deviation $\sigma = ((T_{smax} + T_{smin})/2 - T_{smin})/2.5$.

Experiments are conducted based on the synthesized dataset described above, and the executional environment was a 64 bit jdk running on a Linux system with 4-core i7-3770 CPU with 3.40GHz, and 8GB of memory.

6.2 Granularity of the numerical solution

As mentioned in Section 4, a grid based numerical method is used to measure the cross section area of FB given a certain time instant. Since the cell size significantly influences the area estimation accuracy, we measured the number of grids between two GPS points as a metric to determine the level of granularity. In the experiment, a traditional bead is formed by two GPS sample points with a maximum speed. We pick an arbitrary query time and calculate the location, which acts as a true value A_t. The estimated area is denoted as A_e, and the relative error is defined as: $\delta A = \frac{|A_e - A_t|}{A_t}$ as the ratio between residual and true value.

When determining the grid size, a 1 % tolerance was chosen as a threshold and multiple runs of the experiment were performed for beads with different distance between two GPS sample points.

As expected, we observe in Fig. 15 that the estimation errors decrease as the grid size increases, and the numerical estimations for different size of beads – with distance ranging from 1m to 1000m – have identical errors given the same grid size. Hence, to reach 1 % error tolerance, we choose grid size to be such that there are 100 units between two successive GPS points.

6.3 Lane-crossing query experiment

We assume road networks are composed of two-lane roads. A sequence of trajectories with different lengths are generated and the data sets we used in experiments are not correlated - that is, we generate each dataset separately.

Correctness improvement:

Figure 16 illustrates the number of false positives when lane-crossing query are applied to trajectories, under the bead and FB model. As we can see, the FB eliminates around 40 % of the false positives from GPS-based bead model, due to its reduction of "dead-space".

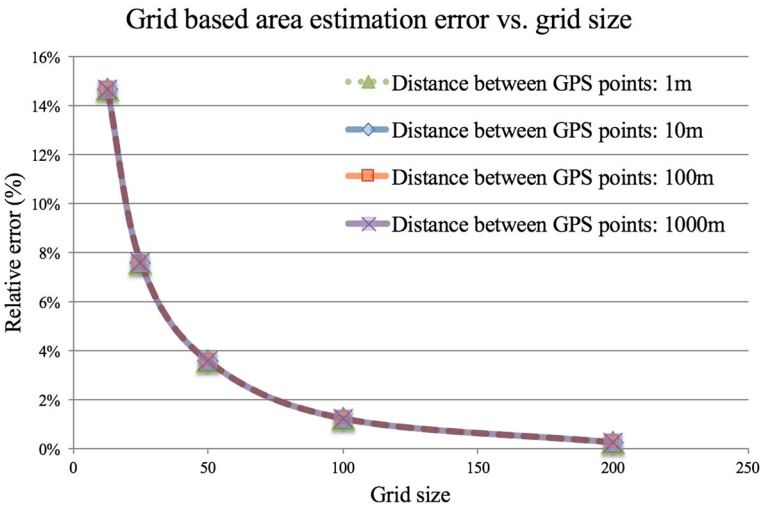


Fig. 15 Impact of grid size on the estimation accuracy

As a follow-up experiment, we reduced the percentages of FBs contained in trajectories to five levels (0 %, 25 %, 50 %, 75 % and 100 %), in order to mimic real situations when roadside sensors are not fully and densely deployed on a given road network. Figure 17 reveals the relationship between percentages of FBs and number of false positives. Clearly, the more FBs contained as components of a trajectory, the smaller the overall number of false positives.

When we apply the same experiment to Beijing Taxi data, its outcome indicates the same effect, where 26.6 % false positives are reduced, as shown in Fig. 18. The real life

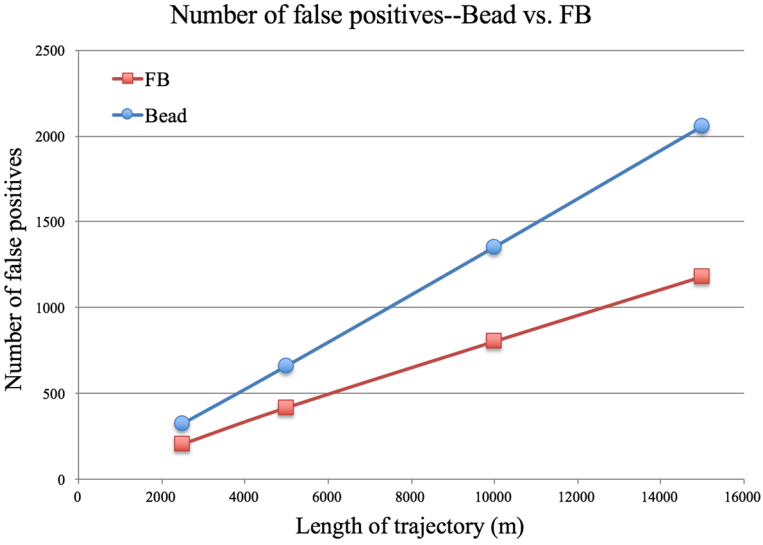


Fig. 16 Lane-crossing query — FB reduces number of false positives

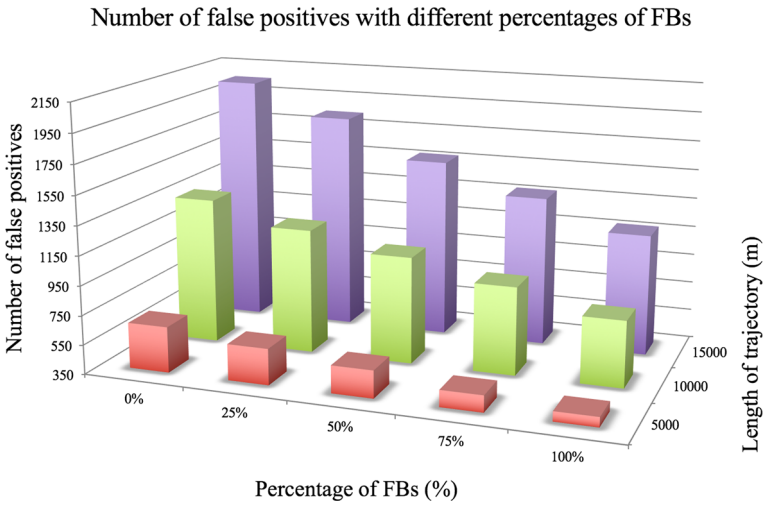


Fig. 17 Percentage of roadside sensor deployed influence the number of false positives

scenarios contained in Beijing taxi data is highly complicated, with continuously changing speed, compared with Brinkhoff trajectory generator where vehicles drive under a constant speed within each road segment. Despite the varieties and complications in real life data, our algorithm is adaptive and effective in reducing location uncertainties.

Sensor deployment in real world applications is largely constrained by factors such as budget, terrain, infrastructure, etc. To add to the realistic aspects of the experiments, we examined the influence of sensor deployment density. The effects of executing the lane-crossing query for the same dataset but for different sensor densities are illustrated in

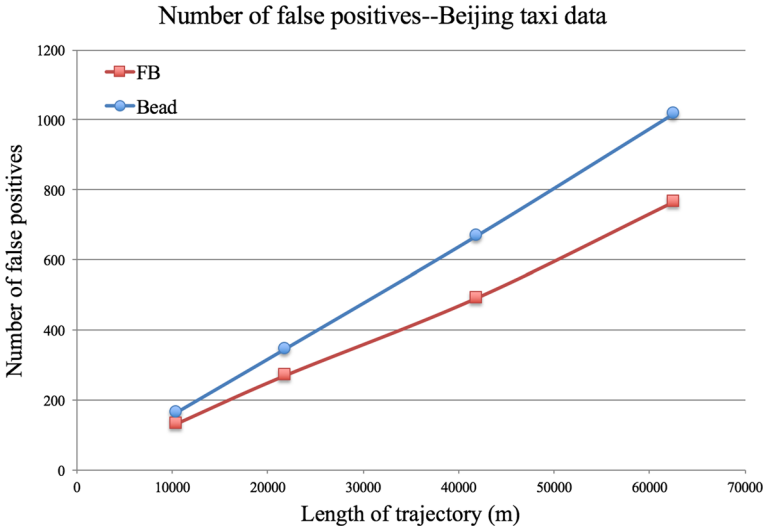


Fig. 18 Lane-crossing query on Beijing taxi data set

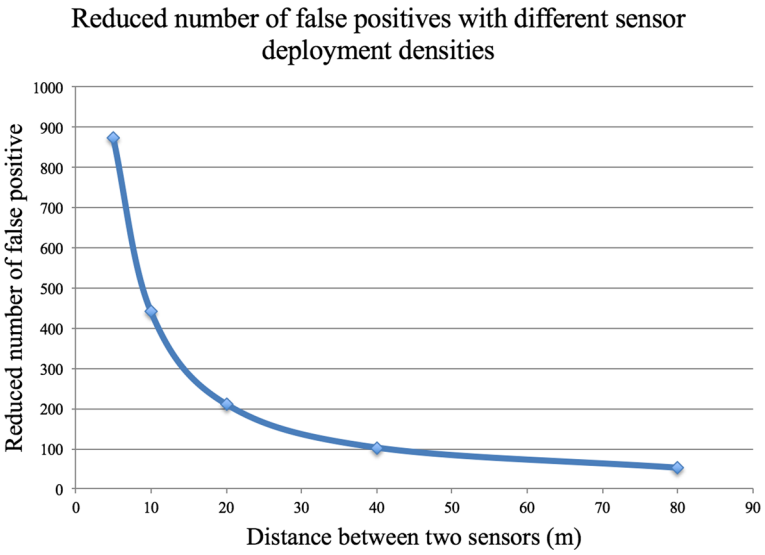


Fig. 19 Road-side sensor deployment density influences the number of false positives

Fig. 19. As shown, the higher the sensor deployment density is, the more false positive we are able to reduce with the FB model.

Efficiency:

Next, we compare the performance of GPS-based bead and FB in terms of the respective execution times. An important parameter affecting the execution time is the number of points chosen to approximate the line segment from roadside sensor (intersecting with lens), which are used to construct sub-beads. Recall from Section 3, that the FB is the union of

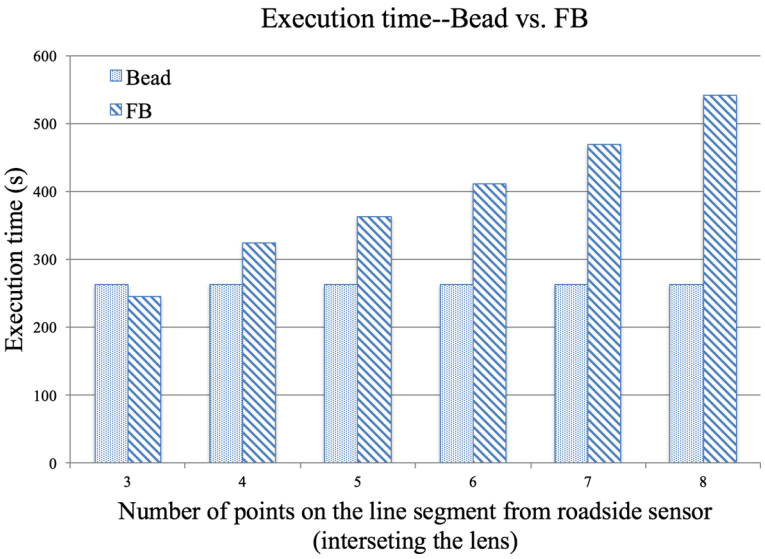


Fig. 20 Execution time comparison

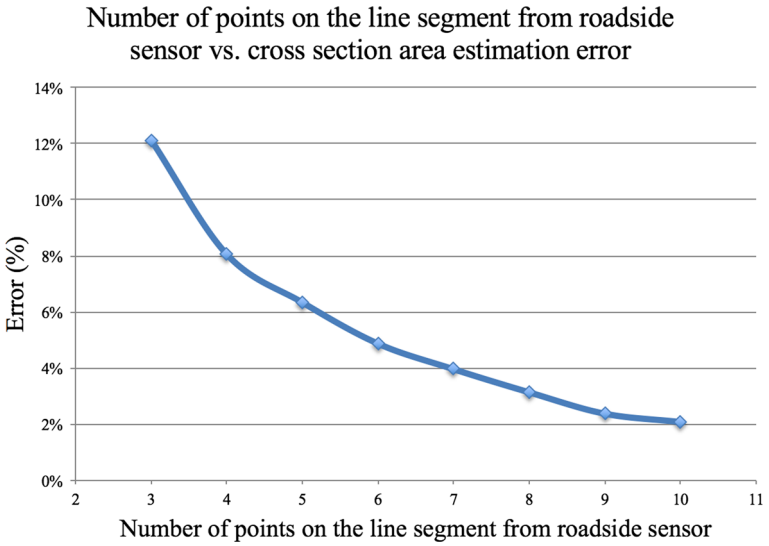


Fig. 21 Number of points on the line segment from roadside sensor affects estimation accuracy

uncountably many sub-beads, with one of the foci located along the ray emanating from the roadside sensor, within the width of the road. When approximating the uncountably many sub-beads, the trade-off is the precision (i.e., using as many points as possible to have an accurate approximation) vs. the computational overheads. Two experiments were performed to compare both execution efficiency and approximation accuracy, executing lane-crossing queries on trajectories with lenght $\leq$ 5000m, varying the number of points chosen on the roadside sensor line segment. The total execution times for each of the GPS-based bead model and FB model are displayed in Fig. 20. On a complementary note, taking maximum number of points allowed in a given grid setting as true value, their corresponding approximation errors regarding location whereabouts areas are shown in Fig. 21.

To discuss one specific setting explicitly: in the case when six points were used on the sensor line segment as a parameter for constructing the FB, in comparison with GPS-based bead model: (1) we have an overhead of a 30 % latency in time to complete the query; (2) however, in return, we reduce more than 26 % false positives and gain a much more narrow possible locations boundary (modulo the accuracy of the area approximation).

6.4 Range-query experiment

The datasets used in range query are generated in the same fashion as the ones for the lane-crossing query, and the query regions we used were squares and disks with respective areas covering 12.5 % of the total map area. As shown in Fig. 22, similarly to the results for the lane-crossing query, FB outperforms GPS-based bead model by reducing 30 % false positives.

6.5 Impact of pruning

As discussed in Section 5, we postulated that pruning techniques can significantly improve the efficiency of the queries processing. In addition, during the early stages of our experiments, we also observed that the fine grained dead-space removal method did not provide

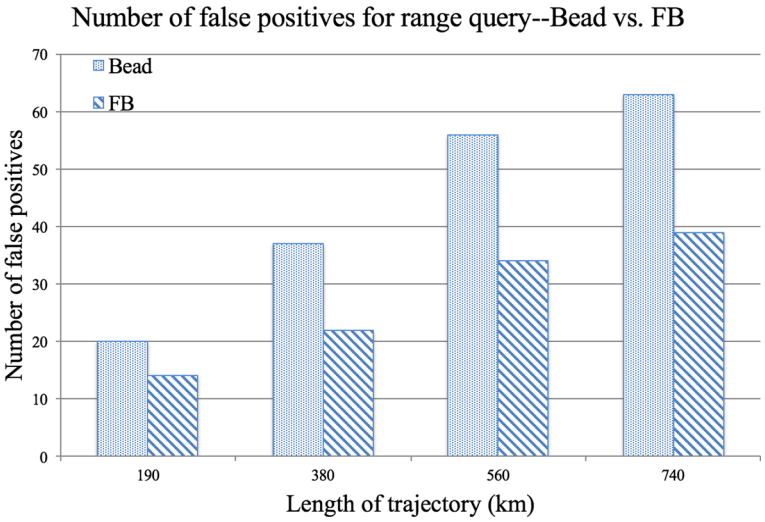
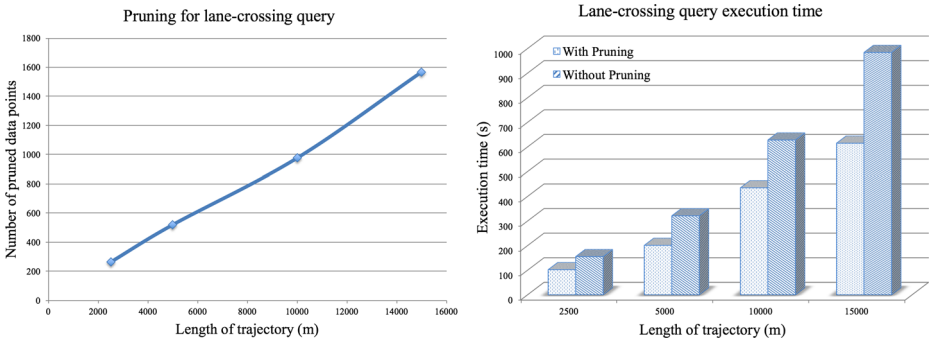


Fig. 22 Range query — FB reduces number of false positives

any significant performance boost. Thus, we introduced the Individual Fused Bead boundary (IBb) as a volume corresponding to a cylinder in 2D + time space, since in Algorithm 1 the most computationally expensive part is to aggregate the total volume in in the respective 2D + time space. Even though the redundant time removal method eliminates some time-intervals with no intersections between FB and query prism from the refinement, we note that one still needs to calculate the volume for those parts.

Figure 23 shows the effects of pre-processing based pruning. As the Fig. 23b indicates, around 30 % of other total time have been saved via pruning. Figure 23a shows that as the trajectory length increases, the number of FB that can be pruned increase accordingly.

Our experiment for the effects of pruning in the case of range query were conducted with a data set containing trajectories with a total length of 55km. IBb rules out large amount of data points that definitely have no intersections with query prism. In Fig. 24, x-axis represents the ratio of the areas of the query region and the entire map. It indicates the



(a) Number of data pruned in lane-crossing query (b) Execution time for lane-crossing query

Fig. 23 Pruning techniques applied to lane-crossing query

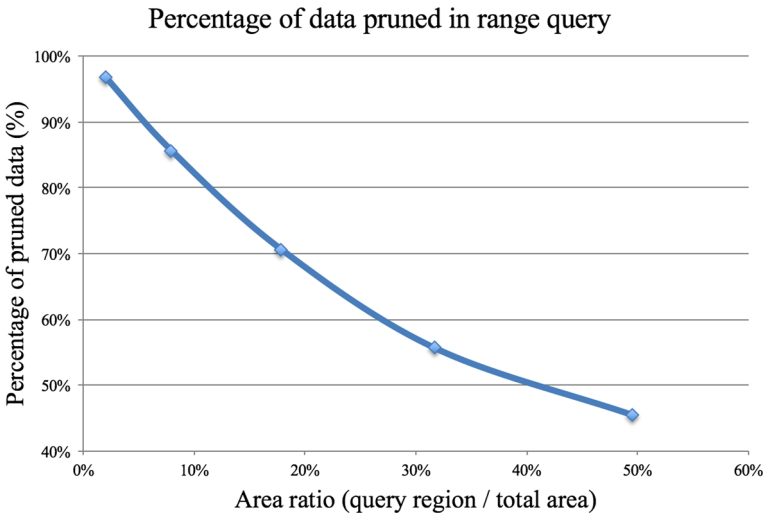


Fig. 24 Pruning — Individual fused bead bounds

relationship between the pruning effect and the size of query region. As we expected, when the query area decrease, the IBb prunes more data points.

7 Related work

There are three main bodies of research literature that are related to the work presented in this article and, in one way or another, were used as foundation for our work.

The first body of works consists of the results from the GIS, MOD and spatio-temporal databases communities, where the problem of capturing the uncertainty of motion has been studied extensively. Starting with [16], and more recently [53], the issue of uncertain whereabouts from the perspective of probabilistic time geography has been tackled by a model of emanating cones-in-time, with a vertex at the last location sample. The 2D boundary of the possible locations of moving objects with bounded speed was formalized by an ellipse in [35], and its 2D+time version – beads – was presented in [19]. Subsequently, [26, 27] provided a full formalization of the beads model and also provided extensions to capture the impact of road networks [24]. A plethora of the works dealing with uncertainty (either in free-space motion or road networks constrained) from MOD and spatio-temporal databases community have also addressed the efficient processing of popular spatio-temporal queries (range, (k)NN, reverse-NN) under various models of uncertainty [4, 15].

Unlike these works, we focused on fusing the uncertain location data from two sources – GPS and the roadside sensors. In addition, to illustrate some of the features of the new model and its use in query processing, unlike the traditional MOD-based works we considered the road network which has a width as a parameter, instead of graph edges.

The second body of works originates in the transportation and traffic management communities. Substantial efforts have been made to tackle the lane-crossing query and several

works have focused on building novel system to overcome the shortcoming of single GPS receivers which yields unstable measurements with large uncertainty [5, 9]. Complementary attempts have been made to acquire location data using commercially available smartphones [41], but nearly 50 % of the data failed to fall within the road network region. Other efforts include the use of integrated sensor like gyroscope to fill the unknown values between two GPS sample updates [46]. However, the works did not consider the uncertainty in-between consecutive GPS-based updates and sensor-based location detections.

Some of the works [9, 46], use map matching algorithms to determine which lane the vehicle belongs to and, subsequently, try to revise the measurement error using post-processing. However, the bead (or, space-time prism) model has not been exploited.

The third body of works originates from the Wireless Sensor Networks (WSN) community. Tracking of moving objects is considered to be a canonical research problem in WSN settings. Various facets of the problem have been investigated: from the trade-off between energy consumption and the accuracy of the tracking process, to routing protocols for conveying location-in-time information to a given sink (see, e.g., [2, 3, 20, 34, 52]). Typically, the location of a given object is determined by some form of collaborative trilateration among the tracking sensors equipped with different distance-estimation devices (e.g., vibrations, audio-strength, etc.). However, to the best of our knowledge, there have been no results on fusing the location data from heterogeneous sources.

8 Concluding remarks and future works

We proposed a formal model – FB (fused bead) – for capturing the possible whereabouts of a moving object whose location data is obtained at discrete time-instants, either by a GPS-device, or by a roadside sensor. Each of them entails a specific kind of uncertainty for the location-in-time data, however, we demonstrated that when combining the values from the two sources it turns out that "two uncertainties are better than one. In other words, integrating/fusing the data from both sources narrows the possible whereabouts when compared to each individual location data source. We analyzed the details of the FB model, and its impact on the lane-crossing query and range query, and conducted a collection of experiments to compare the overall performance between GPS-based bead model and FB model. We demonstrated that, by accepting a small amount overhead in the processing time, our FB model reduces the number of false positives.

There are a few directions that we plan to pursue in the near future. Firstly, we would like to investigate the impact of incorporating other types of sensors and location sources – e.g., the ones obtained via cellular networks [17] or indoor-localization – and develop a formal model capable of multisensor fusion [22]. A particular challenge in these settings is that different data sources may have different horizons of spatial and temporal validity. Our second extension is to consider the processing of other popular spatio-temporal queries (e.g., Nearest Neighbor) under the new model of location uncertainty. Yet another avenue is to extend the model/formalism so that it captures the uncertainty/imprecision in the very samples [37] as well as the possibility of accelerated motion [25]. Lastly, we are also planning to investigate the impact that incorporating other kinds of semantic information in the model – both in the representation of the trajectories [33], e.g., type of a vehicle (for fuel consumption, size, etc.), as well as the type of roads [30] – can have on the processing of spatio-temporal queries.

Appendix A: Significant times in instantaneous possible location query

In Section 4 we analyze the boundary of the possible locations at a given time instant under the FB model. The detailed significant times calculation will be presented here. Let d_{min} and d_{max} denote the shortest and longest distance from L_1 to any point $P(t_{s1}, \varepsilon) \in \overline{P_1 P_2}$.

$$
\begin{aligned}
t_i^{l1} &= \frac{t_1 + t_s}{2} - \frac{d_{max}}{2v_{max}} \\
t_i^{lA} &= \frac{t_1 + t_s}{2} - \frac{d_{min}}{2v_{max}} \\
t_i^{d1} &= \frac{t_1 + t_s}{2} + \frac{d_{min}}{2v_{max}} \\
t_i^{dA} &= \frac{t_1 + t_s}{2} + \frac{d_{max}}{2v_{max}}
\end{aligned}
$$

Appendix B: Enter/exit time calculation for range query

The general case for time $t \in [t_i, t_{i+1}]$ being a critical point occurs when the intersection of the uncertain region at t with a query rectangle is a single point. In the time interval $[t_i, t_s]$, the single-point-intersection between disk centered at the first GPS point and query region stands for the entering moment. Similarly, in the time interval $[t_s, t_{i+1}]$, the single-point-intersection represents exiting moment. Since the query region is represented as polygon in the (X, Y) plane, each edge of the polygon is defined as a segment of 2D line $y = ax + b$.

The entry boundary of FB is:

$$(x - x_i)^2 + (y - y_i)^2 = (t - t_i)^2 v_{max}^2$$

Substituting for y for the equation of the line, we have:

$$(x - x_i)^2 + (ax + b - y_i)^2 = (t - t_i)^2 v_{max}^2$$

This yields an equation in x and t:

$$A * x^2 + x * (B + C * t) + D * t^2 + E = 0$$

Where A, B, C, D, E are constant. Solving for x, as a function of t, we have:

$$x_{1,2} = \frac{-(B + C * t) \pm \sqrt{(B + C * t)^2 - 4 * A * (D * t^2 + E)}}{2 * A}$$

To be noted that, we need to check the solution for x against the boundaries of the respect edge of the query region. To find the time for critical point, we set the discriminant to be zero:

$$\sqrt{(B + C * t)^2 - 4 * A * (D * t^2 + E)} = 0$$

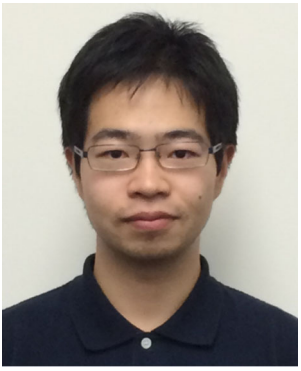
The real root t_{in} is the time instant when the uncertain trajectory start to enter the query prism.

In the time interval $[t_s, t_{i+1}]$, we can use the similar method to find the exiting time t_{out}.

References

1. Brugere I, Gunturi VMV, Shekhar S (2014) Modeling and analysis of spatiotemporal social networks. In: Encyclopedia of Social Network Analysis and Mining, pp 950–960
2. Cao Q, Yan T, Stankovic J, Abdelzaher T (2005) Analysis of target detection performance for wireless sensor networks. In: DCOSS, pp 276–292
3. Chen W, Hou J, Sha L (2003) Dynamic clustering for acoustic target tracking in wireless sensor networks. In: IEEE International Conference on Network Protocols (ICNP'03)
4. Cheng R, Emrich T, Kriegel HP, Mamoulis N, Renz M, Trajcevski G, Züfle A (2014) Managing uncertainty in spatial and spatio-temporal data. In: ICDE, pp 1302–1305
5. Dao TS, Leung KY, Clark CM, Huissoon JP (2007) Markov-based lane positioning using intervehicle communication. Trans. Intell. Transport. Sys. 8(4):641–650. doi:10.1109/TITS.2007.908574
6. Department of Transportation (2014) U.S.: Travel monitoring and traffic volume. http://www.fhwa.dot.gov/policyinformation/travelmonitoring.cfm. Office of Highway Policy Information
7. Ding Z, Güting RH (2004) Managing moving objects on dynamic transportation networks. In: SSDBM
8. Ding Z, Güting RH (2004) Uncertainty management for network constrained moving objects. In: DEXA, 411–421
9. Du J, Barth M (2008) Next-generation automated vehicle location systems: Positioning at the lane level. IEEE Trans. Intell. Transp. Syst. 9(1):48–57. doi:10.1109/TITS.2007.908141
10. EasySen LLC (2008) Wieye - sensor board for wireless surveillance applications. 401 North Coquillard Dr., South Bend, IN 46617
11. Emrich T, Kriegel HP, Mamoulis N, Renz M, Züfle A (2012) Querying uncertain spatio-temporal data. In: ICDE, 354–365
12. George B, Shekhar S (2008) SP-TAG: a routing algorithm in non-stationary transportation networks. In: 5th Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services, MobiQuitous 2008, Dublin, Ireland
13. Gilat A (2010) Numerical methods for engineers and scientists. Wiley
14. Gowrisankar N, Nittel S (2002) Reducing uncertainty in location prediction of moving objects in road networks. In: GIScience
15. Güting RH, Schneider M (2005) Moving objects databases. Morgan Kaufmann
16. Hägerstrand T (1970) What about people in regional science Papers of the Regional Science Association 24:7–21
17. Hellebrandt M, Mathar R (1998) Location tracking of mobiles in cellular radio networks 45(5):1558–1562
18. Honeywell International Inc (2005) Vehicle detection using amr sensors. Tech. rep., Defense and Space Electronics Systems, 12001 Highway 55, Plymouth, MN 55441
19. Hornsby K, Egenhofer MJ (2002) Modeling moving objects over multiple granularities. Ann Math Artif Intell 36(1–2):177–194
20. Jeong J, Hwang T, He T, Du DHC (2007) Mcta: Target tracking algorithm based on minimal contour in wireless sensor networks. In: INFOCOM
21. Kelly R (2007) M42 active traffic management scheme. Road Traffic Technology. S.M. Limited Editor. Birmingham, United Kingdom
22. Khaleghi B, Khamis AM, Karray F, Razavi SN (2013) Multisensor data fusion: A review of the state-of-the-art. Information Fusion 14(1):28–44. doi:10.1016/j.inffus.2011.08.001
23. Kim S, Shekhar S (2005) Contraflow network reconfiguration for evacuation planning: a summary of results. In: 13th ACM International Workshop on Geographic Information Systems, ACM-GIS 2005, November 4–5, 2005, Bremen, Germany, Proceedings, pp 250–259
24. Kuijpers B, Miller HJ, Neutens T, Othman W (2010) Anchor uncertainty and space-time prisms on road networks. Int J Geogr Inf Sci 24(8):1223–1248
25. Kuijpers B, Miller HJ, Othman W (2011) Kinetic space-time prisms. GIS '11, pp 162–170. ACM, Chicago, IL, USA. doi:10.1145/2093973.2093996
26. Kuijpers B, Othman W (2009) Modelling uncertainty on road networks via space-time prisms. Int'l Journal on GIS 23(9)
27. Kuijpers B, Othman W (2009) Trajectory databases: data models, uncertainty and complete query languages. Comput Syst Sci. doi:10.1016/j.jcss.2009.10.002
28. Laser Technology Inc (2013) Trusense t-series. Online; accessed May 22, 2014
29. Leonard T (2012) Delivering deeper insights with big data and real-time analytics
30. Li G, Li Y, Shu L, Fan P (2011) Cknn query processing over moving objects with uncertain speeds in road networks. In: APWeb, pp 65–76

31. Liu W, Zheng Y, Chawla S, Yuan J, Xing X (2011) Discovering spatio-temporal causal interactions in traffic data streams. In: KDD, pp 1010–1018
32. Mckinsey Global Institute (2011) Big data: The next frontier for innovation, competition, and productivity
33. Parent C, Spaccapietra S, Renso C, Andrienko G, Andrienko N, Bogorny V, Damiani ML, Gkoulalas-Divanis A, Macedo J, Pelekis N, Theodoridis Y, Yan Z (2013) Semantic trajectories modeling and analysis. ACM Comput Surv 45(4):42:1–42:32
34. Pattem S, Poduri S, Krishnamachari B (2003) Energy-quality tradeoffs for target tracking in wireless sensor networks. In: IPSN
35. Pfoser D, Jensen CS (1999) Capturing the uncertainty of moving objects representation. In: SSD
36. Pfoser D, Tryfona N (2001) Capturing fuzziness and uncertainty of spatiotemporal objects. In: ADBIS, pp 112–126
37. Pfoser D, Tryfona N, Jensen CS (2005) Indeterminacy and spatiotemporal data: Basic definitions and case study. GeoInformatica 9:3
38. Quddus MA, Ochieng WY, Noland RB (2007) Current map-matching algorithms for transport applications: State-of-the art and future research directions. Transportation Research Part C: Emerging Technologies 15(5):312–328. doi:10.1016/j.trc.2007.05.002. http://www.sciencedirect.com/science/article/pii/S0968090X07000265
39. Schiller J (ed) (2004) A.V.: Location-Based Services. Morgan Kaufmann
40. Schramm AJ, Rakotonirainy A (2009) The effect of road lane width on cyclist safety in urban areas. In: Proceedings of the 2009 Australasian Road Safety Research, Policing and Education Conference: Smarter, Safer Directions. Roads and Traffic Authority of New South Wales, Australia
41. Sekimoto Y, Matsubayashi Y, Yamada H, Imai R, Usui T, Kanasugi H (2012) Lightweight lane positioning of vehicles using a smartphone gps by monitoring the distance from the center line. In: ITSC 2012, pp 1561–1565. doi:10.1109/ITSC.2012.6338737
42. Shekhar S, Chawla S (2003) Spatial Databases: A Tour. Prentice Hall
43. Sistla A, Wolfson P, Chamberlain S, Dao S (1999) Querying the uncertain positions of moving objects. In: Etzion O, Jajodia S, Sripada S (eds) Temporal Databases, Research and Practice
44. Southwest Research Institute Advanced traffic management systems. http://www.swri.org/4ORG/d10/its/atms/default.htm
45. Tao Y, Xiao X, Cheng R (2007) Range search on multidimensional uncertain data. ACM Trans Database Syst 32(3):15
46. Toledo-Moreo R, Betaille D, Peyret F (2010) Lane-level integrity provision for navigation and map matching with gnss, dead reckoning, and enhanced maps. IEEE Trans Intell Transp Syst 11(1):100–112. doi:10.1109/TITS.2009.2031625
47. Trajcevski G, Choudhary A, Wolfson O, Li Y, Li G (2010) Uncertain range queries for necklaces. In: MDM
48. Trajcevski G, Tamassia R, Cruz I, Scheuermann P, Hartglass D, Zamierowski C (2011) Ranking continuous nearest neighbors for uncertain trajectories. VLDB J 20(5):767–791
49. Trajcevski G, Wolfson O, Hinrichs K, Chamberlain S (2004) Managing uncertainty in moving objects databases. ACM Trans Database Syst 29(3)
50. Turner-Fairbank Highway Research Center Traffic Detector Handbook. U.S. Department of transportation
51. United States Department of Defense (2008) Navstar gps: Global positioning system standard
52. Wang H, Yao K, Estrin D (2005) Information-theoretic approaches for sensor selection and placement for target localization and tracking in sensor networks. JCN 7(4):438–449
53. Winter S, Yin ZC (2010) Directed movements in probabilistic time geography. Int J Geogr Inf Sci 24(9)
54. Yuan J, Zheng Y, Xie X, Sun G (2011) Driving with knowledge from the physical world. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, pp 316–324. ACM, New York, NY, USA. doi:10.1145/2020408.2020462
55. Yuan J, Zheng Y, Xie X, Sun G (2013) T-drive: Enhancing driving directions with taxi drivers' intelligence. IEEE Trans Knowl Data Eng 25(1):220–232. doi:10.1109/TKDE.2011.200
56. Zhang B, Trajcevski G (2014) The tale of (fusing) two uncertainties. In: Proc. International Conference ACM SIGSPATIAL GIS



Bing Zhang is a Ph.D. candidate in Computer Science at Northwestern University. He received his B.S.E. degree in Computer Engineering from the University of MichiganAnn Arbor. His research interests are Moving Objects Databases (MOD), Data Mining and Mobile Data Management.

Goce Trajcevski received his B.Sc. degree from the University of Sts. Kiril i Metodij, and his MS and PhD degrees from the Dept. of Computer Science at the University of Illinois at Chicago. His main research interests are in the areas of spatio-temporal data management, routing and data management in wireless sensor networks, and reactive behavior in dynamic systems. He has published over 90 papers in refereed conferences and journals and received a Best Paper Award at the CoopIS conference (2000), Best Paper Award at the IEEE MDM conference (2010) and Best Short Paper Award at ACM MSWiM conference (2013). His research has been funded by BEA, Northrop Grumman Corp., NSF and ONR. He has served as an associate editor at ACM DiSC, and is presently an associate editor of GeoInformatica and ACM Transactions on Spatial Algorithms and Systems (TSAS). He has served on program and organizing committees in numerous conferences and workshops. Currently, he is an Assistant Chairman with the Department of Electrical Engineering and Computer Science at the Northwestern University.

Liu Liu is a Ph.D. student in Computer Science at Northwestern University (NU). He received his Bachelor of Science degree from the University of Arizona, and his Masters from Northwestern University, all in computer science. His research focuses on big temporal data streaming and analysis with distributed systems.