Karlsruhe Reports in Informatics 2018,3
Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Comparing Predictions of Object Movements

Saeed Taghizadeh, Martin Schäler, Klemens Böhm

2018

KIT – University of the State of Baden-Wuerttemberg and National Research Center of the Helmholtz Association

Comparing Predictions of Object Movements

Saeed Taghizadeh
Karlsruhe Institute of Technology
saeed.taghizadeh@kit.edu

Martin Schäler
Karlsruhe Institute of Technology
martin.schaeler@kit.edu

Klemens Böhm
Karlsruhe Institute of Technology
klemens.boehm@kit.edu

ABSTRACT

Estimating the future location of moving objects using different estimation models, such as linear or probabilistic models, has been investigated extensively. However, the location estimations of those models are generally not comparable. For instance, one model might return a position for some object, another one a Gaussian probability distribution, and a third one a uniform distribution. Similar issues arise for query answers. In this paper, we examine the question how estimations of different models can be compared. To do so, we propose a general model based on the central limit theorem. This allows handling different PDF-based approaches as well as models from the other groups (i.e., linear estimations) in a unified manner. Furthermore, we show how to inject privacy into the general model, a fundamental pre-requisite for user acceptance. Thus, we support well-known approaches like k-anonymity and spatial obfuscation. Based on our general model, we conduct a comprehensive experimental study considering a real-world road network; comparing models form different groups for the first time. Our results, for instance, reveal that estimation models based on individual velocity profiles are not necessarily better than models, which estimate the future location of objects only based on their direction. In more abstract terms, our general model allows comparison of estimation models that could not be compared before and gives way to build models that solve the privacy-accuracy challenge.

1. INTRODUCTION

Estimating future positions of moving objects is of great interest for location-based services (LBS) and moving object databases (MOD). The position of an object at any time is specified by its *initial location data*, usually $(\boldsymbol{x}, \boldsymbol{y})$ coordinates, and its *motion data*. Estimation accuracy depends on the precision of those data.

If the initial location of moving objects is available, the objects fall into **three** groups: The first one is objects which do not provide any data regarding their motion. For this group, approaches for position estimation must rely on assumptions, such as bounds on the velocity [1]. Second, there are objects which contribute *some* of their motion data, like their maximal speed [1, 2]. Objects in the third group provide all of their motion data at a fine-grained level. These approaches either (1) model motion as continuous function of time [3, 4], or (2) sample the velocity for fixed time intervals from a (learned) probability distribution, named ***velocity profile*** [5].

Having studied various estimation models, we observe that these estimation models are by themselves incomparable. The user of a system does not know which model best describes the motion of the object, and to which extent it is better than the other models. Such comparisons also depend on the parametrization. That incomparability exists at various levels, as we illustrate next.

Example 1. Model Accuracy. Think of an object starting its movement in Position $\boldsymbol{x} = \boldsymbol{0}$**,** and we estimate its position after $\boldsymbol{t} = \boldsymbol{100s}$ using different models, resulting in the estimations in **Figure 1**. To examine model accuracy, one needs to quantify how well each model estimates the real position. However, comparing model accuracy is difficult, as some approaches use different types of probability distributions, while others only return a point or yield lower and upper bounds without any probability. ■

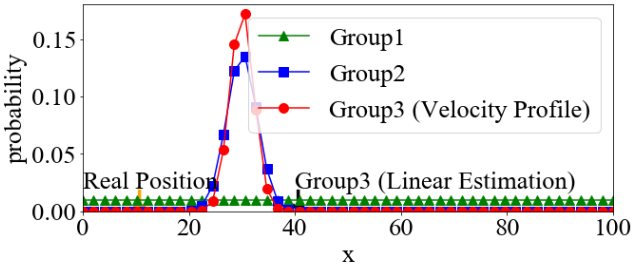

Figure 1. Uncertainty Distribution of Object

Evaluating the model accuracy is important as this is the foundation any LBS is built on. However, even if we can quantify the quality of the estimation models intrinsically, this does not mean that we can compare query results with high quality models:

Example 2 Query Result Accuracy: A user poses a range query $\boldsymbol{RQ}(\boldsymbol{t}, \boldsymbol{Query\ Area})$ against a MOD, retrieving all objects with their probability of being inside the query area at **time** $\boldsymbol{t}$**,** where $\boldsymbol{t}$ is some instant in the future. With different estimation models, she gets two answer sets with different uncertainty degrees for each object With $\boldsymbol{p_{ij}}$ the uncertainty degree of object $\boldsymbol{j}$ in answer set $\boldsymbol{i}$ we have:

$$AnswerSet1 = \{(o_1, p_{11}), (o_2, p_{12}), \dots, (o_n, p_{1n})\}\ \forall i\ p_{1i} \geq 0.9$$

$$AnswerSet2 = \{(o_1, p_{21}), (o_2, p_{22}), \dots, (o_n, p_{2n})\}\ \forall j\ p_{2j} \geq 0.7$$

The condition for each answer set means that all objects in the set fulfill it. If n is the number objects in each set, then one might expect at least $n \cdot 0.9$ objects to be in the range for the first set and $n \cdot$ 0.7 objects to be in the second set. Let us assume that $n \cdot$ 0.6 objects are in the query area in reality. Then, higher probabilities of objects do not give any information on how close these estimations are to reality. ■

Next, one needs a metric to compare different estimation models.

Example 3. Comparison Metrics. Suppose that two models, Model 1 and Model 2 as shown in

Figure **2**, have estimated the future location of an object. Both models estimate the object location as a normal distribution. However, the first model is far from the real position, but its uncertainty area is denser. The second model is closer to the real position of the object, but its uncertainty area is wider. Defining comparison metrics which measure the distances and compare results seems necessary**.** ■

There is another perspective on all these issues which makes them even more useful: Next to the perspective that model and query-result quality should be maximal, another perspective is that, to facilitate privacy, certain estimations must not be too accurate. If a certain extent of uncertainty shall be guaranteed, which data should be collected, and how should it be processed? Next, there exist different privacy-protection mechanisms, like k-anonymity, minimum region area privacy [6, 7, 8], differential privacy [9] and geo-indistinguishibility [10]. Differential privacy guarantees a certain privacy when aggregated data is released. In our case however, the data of individual moving objects is available during query processing. So while differential privacy does not play a role here, the other procedures mentioned report object locations as an area, possibly together with a probability distribution, and not as a point. MODs however tend to represent moving objects as points [11, 12, 13, 14, 15]. This calls for a generalization of existing approaches, to support privacy protection.

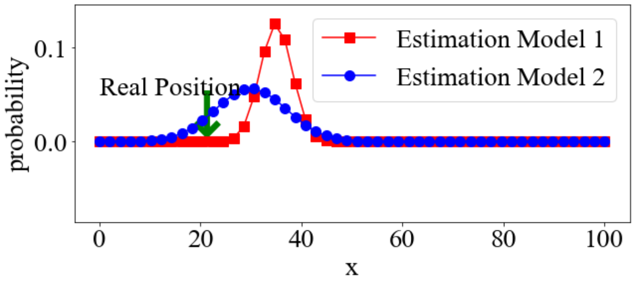

Figure 2. Incomparability of Estimation Models

Example 4 Query-Result Incomparability with Privacy Injected. Think of an object announcing its initial location as a rectangle. After a while, depending on the estimation model in use, the uncertainty area of the object is different, as shown in Figure 3. While the star marks the real location, two different models have estimated the location as two different shapes. The real location is somewhere between the two estimated areas. Two problems arise here: First, deriving the uncertainty area with its PDF is not trivial, as it depends on several factors like the initial area of the object, its PDF, or the estimation model in use. Second, injecting privacy causes the shapes to change from well-known distributions to some unknown ones, and comparisons get more complex. ■

The concern of this article is the design of a general model to compare model and query-result accuracy of estimation approaches and the effect of privacy injected. This will help choosing a good estimation model, depending on the accuracy of the data available or good values of privacy parameters. To illustrate, not all objects may use privacy protection. But the ratio of such objects affects both accuracy types.

Challenges

To our knowledge, it still is an open problem how to compare different estimation models from different perspectives (i.e., considering different queries or inherent accuracy) in a unified manner. We see the following challenges:

1) How to Design a Unified Estimation Model?

Designing a comprehensive model so that one can look at different models from one unified perspective is not trivial. Complexities arise when the models rely on different assumptions and employ different estimation methods. Next, estimates (predicted locations) are of various types (e.g., points or uncertainty regions), which makes the unified framework even more complex. Building comparison metrics for models also is nontrivial, cf. Example 3.

2) How to Compare Query Results?

Uncertainty in movements causes uncertainty in query results. A query result often is a set of objects with a degree of uncertainty assigned to each object. However, this is not applicable in case of queries where one expects one number. The time needed to compute a probability distribution on these objects grows exponentially with the number of objects. This is because of heavy computations needed to derive the uncertainty degree of the objects. On the other hand, comparing query results of different models is practical only when one can obtain the respective distributions in reasonable time. Next, different uncertainty models yield different answer sets for range queries. Comparing these answers to see which model operates more accurately is a challenging task, which the literature does not study well.

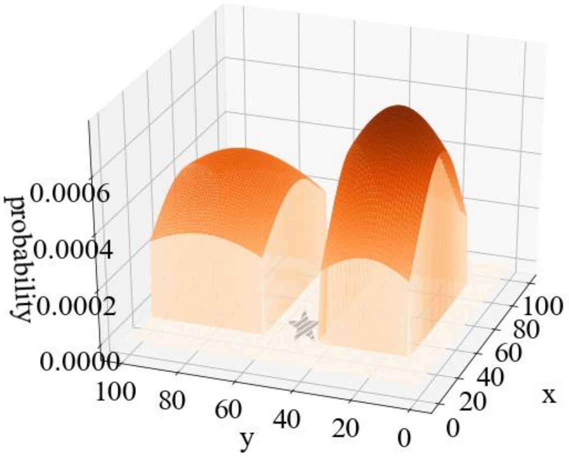

Figure 3. Uncertainty Distribution of Object with Privacy

3) Injecting Privacy

A third challenge deals with privacy: Namely, how to inject privacy-protection mechanisms into the unified model? In addition to uncertainty coming from motion data, our framework should also deal with location uncertainty coming from privacy-protection mechanisms. A user has various options to protect her location, leading to different forms of location uncertainty. Integrating them into our unified model and quantifying the effect of privacy mechanisms on query accuracy is not trivial.

Contributions

We address the three challenge groups as follows: First, we propose a general model to compare different estimation models, based on the central limit theorem. This allows us to handle different PDFs as well as the approaches from the other groups (i.e., linear estimations) in a unified manner and to compare intrinsic model accuracy. Second, regarding the query perspective, we define error metrics to compare result accuracy for range and count queries. For count queries, we employ different filtering mechanisms to obtain the respective distributions in a reasonable time. Third, we show how to inject privacy into the general model, for well-known approaches like k-anonymity and spatial obfuscation. Fourth, based on this framework, we conduct a comprehensive experimental study considering a real-world road network. Past studies [16, 1, 5] yield probabilistic query results, but do not *compare* them. Our results are insightful. For instance, estimation models based on individual velocity profiles are not necessarily better than models which estimate the future location of objects only based on their direction. This means that objects could enjoy some level of privacy with decent query-result quality at the same time.

2. Related Work

To our knowledge, previous work has not focused on the comparison of estimation models. So we only review work on estimation models. In all cases, the difference to our work is that this comparison does not play a role, and we will not explicitly say this another time anywhere in this section.

Early proposals anticipate the most possible path of the object based on linear functions of time [17, 18, 19], or they keep recent information on the object [20]. A drawback is that the uncertainty area (i.e., the area where an object can be) grows fast with time. Other methods [1, 5, 16, 21, 22, 23, 24] rely on the assumption that objects maintain their behavior unless there is an explicit notification. An object can be in any position within the uncertainty area with a probability defined by some probability density function (PDF). One approach is that the PDF is obtained from the velocity distribution of the object [5]. We adopt this approach in this paper.

In [1] the authors have studied time instant queries. In their model, there is an uncertainty region of the object O at time t. The authors have investigated on two types of queries called probabilistic range queries (PRQ) and probabilistic nearest neighbor queries (PNNQ).

Authors in [25] offer a probabilistic model of uncertain trajectories. They model the uncertainty of trajectories at each time instant using uniform distributions. Furthermore they consider the motions constrained by road networks. They have focused on a specific class of spatio-temporal queries called “Universal Range Queries”. These queries aim at finding objects which stay inside a region throughout the whole time interval.

[3] considers object movements without uncertainty. Obviously, the comparisons envisioned are trivial in this specific case because the uncertainty coming from different sources has not been covered by their model. [26] proposes a new operator called Transformed Minkowsky Sum (TMS), to determine whether a moving rectangle collides with a moving circular query region. With this new operator and traditional tree-traversal algorithms they have investigated on range queries and KNN queries. Finally, [4] proposes a Gaussian kernel-based local regression model to smoothen GPS feeds. The core contribution is a hybrid model for developing a semantic overlay, analyzing and transforming raw mobility data (GPS) to meaningful abstractions, e.g., semantic trajectories.

3. Abstract Estimation Model

In the following, we introduce our abstract model that allows us to systematically investigate the challenge groups and compare approaches and assumptions. We first give an intuition of the model and then introduce some foundations. Most assumptions are in line with related work. The primary difference is that we generalize existing approaches by using the central limit theorem to model moving objects.

3.1 Intuition

For moving objects, our key objective is to have a good estimation of their position at any time. This needs to hold also if the last reported position is rather old, or the position reporting is (intentionally) perturbed. Core assumptions in the literature [1, 2, 5, 16] are that (1) each dimension is independent, i.e., conceptually an object can move in any direction, and that (2) the velocities in different dimensions are independent, identically distributed (i.i.d.) random variables. We follow these assumptions throughout this paper.

To have a unified notation and to allow sampling the speed vector from any distribution, we now introduce our abstraction using a version of the central limit theorem. To our knowledge, it subsumes all estimation models from the literature we are aware of.

The abstract model is based on computing one probability density function (PDF) based on the velocity profile.

Definition 1 (*Velocity Profile*). The velocity profile of a moving object is the pair $(\mathbf{v_x}, \mathbf{v_y})$. $\mathbf{v_x}$ and $\mathbf{v_y}$ are two PDFs which are the velocity distributions in the $\boldsymbol{X}$ and $\boldsymbol{Y}$ direction respectively.

Example 5. We assume that $\boldsymbol{v_x}$ obeys a beta distribution with $\boldsymbol{a = 2}$ and $\boldsymbol{b = 2}$. $\boldsymbol{v_y}$ in turn follows a beta distribution with $\boldsymbol{a = 2}$ and $\boldsymbol{b = 5}$. **Figure 4** shows both distributions, where a value of 0 indicates the minimum speed of the object (stationary) and 1 is the maximum speed possible. The $\boldsymbol{y}$ dimension in each figure shows the probability distribution of respective velocities in different directions.

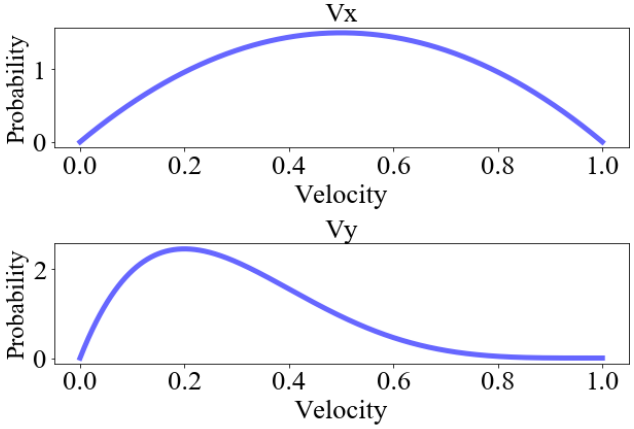

Figure 4. Velocity Profiles in X and Y Direction

3.2 Abstraction Using Central Limit Theorem

We now estimate the position of some object S_n at time t independently of the PDF of the components of the speed vector using an extension of the central limit theorem [27].

Let S_n be the position of a moving object after sampling the position in one dimension *n* times. That is, if the sampling rate is 10 seconds and *n = 10*, then *t = 100*. With the x-dimension as an illustration we have:

$$S_n = \Delta x = v_{1x} * \Delta t + v_{2x} * \Delta t + \cdots + v_{nx} * \Delta t$$

But a sampling rate of one time unit ($\Delta t = 1$) yields:

$$S_n = \Delta x = v_{1x} + v_{2x} + \cdots + v_{nx}$$

Then, according to the central limit theorem (CTL) [28], $\boldsymbol{S_n} \sim \boldsymbol{N}(\boldsymbol{n * \mu}, \boldsymbol{n * \sigma^2})$ holds.

The CLT does not impose any restriction that the random variables must follow a specific distributions, e.g., be uniform, only that they refer to the same distribution. When using the CTL, a fixed sampling rate for the data is required, a common assumption.

So all that is needed is to abstract the motion of each object to the mean and standard deviation of the velocity-probability distribution to get a random variable that corresponds to the estimated location. According to the CLT, this estimate follows a normal distribution. Consequently, we only need the velocity profiles with i.i.d. components per dimension. In this way we can also model 2D and 3D movements, since we consider each dimension separately. Finally, the central limit theorem has a small error when n is large ($n \geq 25$). In our experiments, n is much larger than this threshold (n is about 600) because we look at the traffic of Berlin of one working day, with 10 minutes for an average trip.

3.3 Query Evaluation under Abstract Model

We now say how to answer range queries.

Definition 2 (*RangeQuery*). A range query is a query that returns the probability of a moving object to be inside a given query rectangle at some point in time. We use the notation *range_query(query rectangle, time instant)*. More formally, $AnswerSet = \{(o_i, p_i) | 0 < p_i \leq 1\}$.

In this formula, o_i refers to object i, and p_i is the probability of o_i being in the range. So we generally have to compute the overlap of the query rectangle with the PDF, i.e., compute a bounded integral of the cumulative PDF (CDF). For ease of exposition, we first consider only one dimension and then the general case. Using the CLT, if query range is $[a, b]$, then the cumulative PDF of the standard normal distribution is calculated as follows:

Corollary 1 (Query Overlap in one dimension): For a large number of samples n $\boldsymbol{and}$ for query area $[\boldsymbol{a}, \boldsymbol{b}]$ with $\boldsymbol{a} < \boldsymbol{b}$,

$$P(estimated\ location\ of\ object\ after\ n\ time\ units) = \int_a^b \boldsymbol{S_n}\, dx$$

3.4 Two-Dimensional Abstract Model

For the two-dimensional case, we first need to obtain an estimation of the location of the object in 2D space. The second step is the processing of queries with respect to this estimation model.

3.4.1 Obtaining a Two-Dimensional Estimation Model Using the Velocity Profile

Think of an object starting to announce its location from $t = 0$. We want to see if the object is inside a query rectangle at some time $t >= 0$. The object samples at $times = 0, 1, 2, \ldots, t$ from its corresponding velocity profile. Because we are talking about the future time, the exact velocity for each instant of time is unknown. However, each future velocity sample could be represented as a probability random variable. So the movement vector is as follows:

$$d = (\Delta x, \Delta y)$$

But we have assumed that the object samples from its velocity profile every time unit. So we will have:

$$\Delta x = v_{1x} + v_{2x} + \cdots v_{tx}$$
$$\Delta y = v_{1y} + v_{2y} + \cdots + v_{ty}$$

For simplicity we refer to $\Delta \boldsymbol{x}$ as $\boldsymbol{X}$ and $\Delta \boldsymbol{y}$ as $\boldsymbol{Y}$ from now on.

As stated, the sum of arbitrary identically distributed random variables tends to a normal distribution. Therefore, in our case, according to CLT, X and Y random variables could be approximated by Gaussian random variables. Suppose that the expected value and variance of two velocity variables are as follows:

$$v_{ix}: \mu_x\ and\ var_x$$
$$v_{iy}: \mu_y\ and\ var_y$$

To estimate the location of the object after t time units for X and Y we derive the following:

$$X \sim N(t \cdot \mu_x, t \cdot var_x)$$
$$Y \sim N(t \cdot \mu_y, t \cdot var_y)$$

3.4.2 Two Dimensional Query Evaluation

In 2D space we have two variables. So for the joint distribution we need to define:

$$P(X = x, Y = y)$$

The two variables are independent. So we have the following:

$$P(X = x, Y = y) = P(X = x) \cdot P(Y = y)$$

To compute the probability of being inside the query rectangle, we take the integral over all points x and y inside the rectangle.

$$A = being\ inside\ query\ rectangle\ at\ time\ t$$

$$Pr(A) = \iint_{(x,y) \in QR} P(X = x, Y = y) dy dx$$
$$= \int_{x \in QRx} P(X = x) dx \,. \int_{y \in QRy} P(Y = y) dy$$

We know that X and Y are normal random variables. If we name their respective cumulative density function (CDF) F and G, we have the following:

$$Pr(A) = \big(F(right) - F(left)\big) * (G(up) - G(down))$$

With the help of above formula, one can evaluate the uncertainty degree of each object with respect to query boundaries in two dimensional space.

4. Model Extensions

So far we have derived a general two-dimensional estimation model and have described the corresponding query processing. But we have not yet addressed location privacy for the initial location announcement, a topic of this section. Next, while we have discussed range queries for a single point of time. we now also cover range queries over a time period, i.e., objects will be within a specific range during the entire time interval. After this, we will indeed compare different estimation models, the main concern of this paper.

4.1 Range Queries over Time Periods

We compute the probability that an object always is in the rectangle during a time interval $[t1, t2]$. To do so, we need to take the integral over all points x and y which are inside the query rectangle over the entire time period $\Delta t = t2 - t1$.

$$A = being\ inside\ query\ rectangle\ during\ the\ interval\ [t1, t2]$$

$$Pr(A) = \frac{1}{\Delta t} \int_{t1}^{t2} \iint_{(x,y) \in QR} P(X = x, Y = y, T = t) dy dx\ dt$$

4.2 Adding Privacy Mechanisms

We now extend the model so that, instead of reporting the exact location, a moving object might report an area it is in. One very common privacy mechanism is spatial cloaking [7, 6, 29, 30, 31], which we will use as well. Spatial cloaking hides the true location of the user in a region contingent on different policies like saving k-anonymity in the spatial region (i.e., the cloaked spatial region contains at least k users) or the minimum region area privacy requirements [6, 7, 8]. One way to fulfil these privacy policies is to have some user-defined *r*. Then the moving object reports some sphere (x_c, y_c, r), so that its real position (inside the sphere) and all points inside the sphere have the same probability to be the real position. For good privacy protection, the actual position of the object is not necessarily the center of the sphere. We have the following theorem which allows one to inject privacy protection mechanism into the estimation model in a unified manner. Also note that the user announces her initial location as a sphere around her instead of revealing her exact location. To allow for reporting a sphere of radius R instead of a point as location, we extend the estimation model so that we compute the bounded integral for all points inside the sphere and the query rectangle using polar coordinates, as follows:

Theorem 1 Let Event A be defined as follows:

$$A \;=\; being\ inside\ query\ rectangle\ at\ time\ t$$

$$starting\ from\ any\ point\ inside\ the\ circle\ (x_c, y_c)\ with\ radius\ r$$

Then the probability of Event $\Pr(A)$ is as follows:

$$Pr(A) = \frac{1}{\pi * R^2}\int_0^R \int_0^{2\pi} \left[\left(F\left(\frac{x_2 - \ m_x - R * cos\theta - x_c}{\sqrt{2} * var_x}\right) - \ F\left(\frac{x_1 - \ m_x - R * cos\theta - \ x_c}{\sqrt{2} * var_x}\right)\right) . \left(G\left(\frac{y_2 - \ m_y - R * sin\theta - \ y_c}{\sqrt{2} * var_y}\right) - \ G\left(\frac{y_2 - \ m_y - R * sin\theta - \ y_c}{\sqrt{2} * var_y}\right)\right)\right] d\theta dr$$

Here, F and G are cumulative distribution functions (CDF) for x and y respectively.■

Proof. To prove this, we can assume the movement has the ability to be started from any point within the circle. But because no distribution has been defined on the points inside the circle, the weight will be the same value for all the points. It means that the starting point of the movement is uniformly distributed among all points inside the circle. To overcome the infinite number of the points, we need to use differential calculus. We need two new random variables which are defined as below:

$$X' = X + \ InitDevX + \ \ x_c$$

$$Y' = Y + \ InitDevY + \ \ y_c$$

$InitDevX\ and\ \ InitDevY$ are two variables which show the deviation of the selected point inside the circle from its center.

The calculations will be as follow:

$$A \;=\; being\ inside\ query\ rectangle\ at\ time\ t$$
$$=\; t1\ starting\ from\ any\ point\ inside\ the\ circle\ (x_c, y_c)\ with\ radius\ R$$

$$Pr(A) = \frac{1}{\pi * R^2}\int_0^R \int_0^{2\pi} \iint_{(x,y)\in QR} P(X' = x\ , Y' = y) dydx \ \ d\theta\ dr$$

$$= \frac{1}{\pi * R^2}\int_0^R \int_0^{2\pi} \iint_{(x,y)\in QR} P(X' = x) * P(\ Y' = y) dydx \ \ d\theta\ dr$$

$$= \frac{1}{\pi * R^2}\int_0^R \int_0^{2\pi} \left(\int_{x\in QRx} P(X' = x)dx * \int_{y\in QRy} P(\ Y' = y)dy\right) d\theta\ dr$$

To continue the calculations we need to find out the probability distribution over random variables X' and Y' with respect to their definition.

We know that normal distribution is closed under linear operations. So we can calculate the expected value and variance for the new variables as follow:

$$m_{x,} = \ m_x + C_1$$

$$var_{x,} = \ var_x$$

$$m_{y,} = \ m_y + C_2$$

$$var_{y,} = \ var_y$$

So X' and Y' are normal variables with above characteristics.

$$\int_{x\in QRx} P(X' = x)dx \ = \left(F\left(\frac{x_2 - \ m_{x,}}{\sqrt{2} * var_{x,}}\right) - \ F\left(\frac{x_1 - \ m_{x,}}{\sqrt{2} * var_{x,}}\right)\right)$$
$$= \left(F\left(\frac{x_2 - \ m_x - R * cos\theta - x_c}{\sqrt{2} * var_x}\right) - \ F\left(\frac{x_1 - \ m_x - R * cos\theta - \ x_c}{\sqrt{2} * var_x}\right)\right)$$

The same procedure is applicable to second integral part as follow:

$$\int_{y\in QRy} P(Y' = y)dy = \left(G\left(\frac{y_2 - \ m_{y,}}{\sqrt{2} * var_{y,}}\right) - \ G\left(\frac{y_1 - \ m_{y,}}{\sqrt{2} * var_{y,}}\right)\right)$$
$$= \left(G\left(\frac{y_2 - \ m_y - R * sin\theta - \ y_c}{\sqrt{2} * var_y}\right) - \ G\left(\frac{y_2 - \ m_y - R * sin\theta - \ y_c}{\sqrt{2} * var_y}\right)\right)$$

Hence we will have the following formula:

$$Pr(A) = \frac{1}{\pi*R^2}\int_0^R \int_0^{2\pi} \left[\left(F\left(\frac{x_2 - m_x - R*cos\theta - x_c}{\sqrt{2}*var_x}\right) - \ F\left(\frac{x_1 - m_x - R*cos\theta - \ x_c}{\sqrt{2}*var_x}\right)\right) * \left(G\left(\frac{y_2 - m_y - R*sin\theta - \ y_c}{\sqrt{2}*var_y}\right) - \ G\left(\frac{y_2 - m_y - R*sin\theta - \ y_c}{\sqrt{2}*var_y}\right)\right)\right] d\theta dr■$$

The theorem lets us compute the probability of an object being inside a query rectangle in the presence of spatial noise. So we can now quantify the effect of noise on accuracy. In our experiments, we use this formula to obtain the uncertainty degrees of objects which hide their initial location.

5. Evaluation Criteria

As discussed before, the evaluation of estimation models depends on the perspective. In this section we define measures for the inherent accuracy of estimation models. Next, an important issue in comparing estimation models is the problem of heavy computations. We will show in the following that under some mild conditions we can decrease these heavy calculations significantly.

A common way to compare the effectiveness of models is to compute the deviation of the estimated path from the corresponding real one. To do so, we calculate the distance of the exact location of an object to an object which is obeying the velocity distribution. To this end, we calculate the weighted distance between the point and every point in the distribution:

$$Distance(Point(x_0,y_0),Distribution) = \iint_{(x,y)\in R^2} f_{XY}(x,y) \cdot distance\big(x,y,Point(x_0,y_0)\big)dxdy$$

We will use this metric in our experiments.

Calculating the integral may be time consuming. If m is the number of intervals for the first integral and n the number for the second integral, the calculation is in $O(m\cdot n)$. In some cases like when $var_x = var_y$ we can resort to the following theorem. It reduces the problem from a two-dimensional integral to a one-dimensional integral calculation.

Theorem 2 If $\boldsymbol{var_x = var_y = \sigma}$ then we have the following:

$$Distance(Point(x_0,y_0),Distribution) = 2\pi\cdot d^2\cdot \int_{r[0,\infty)} r\cdot e^{-\frac{r^2}{2\sigma^2}}dr + 2\pi \cdot \int_{r[0,\infty)} r^3\cdot e^{-\frac{r^2}{2\sigma^2}}dr$$

d is the Euclidean distance between $Point(x_0,y_0)$ and the center of the distribution.■

Proof. We use polar coordinates to calculate the distance between a point and a distribution as shown in Figure 5.

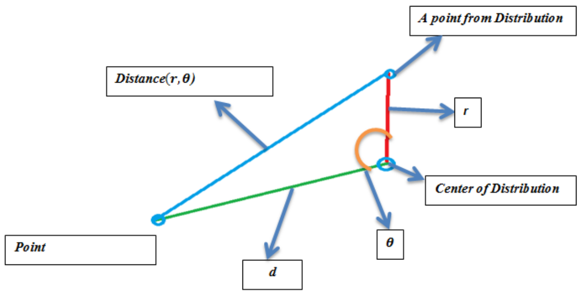

Figure 5. Distance between Point and Distribution

$$Distance\ (Point\ ,Distribution) = \iint_{r[0,\infty),\theta[0,2\pi]} distance(r,\theta)^2 * weight(r,\theta)d\theta dr$$

$$= \iint_{r[0,\infty),\theta[0,2\pi]} (k^2 + r^2 - 2*r*d*\cos\theta) * \left(r * e^{-\frac{r^2}{2\sigma^2}}\right) d\theta dr$$

$$= \iint_{r[0,\infty),\theta[0,2\pi]} [\left(d^2 * r * e^{-\frac{r^2}{2\sigma^2}}\right) + \left(r^3 * e^{-\frac{r^2}{2\sigma^2}}\right) - \left(2*r^2*d*e^{-\frac{r^2}{2\sigma^2}}*\cos\theta\right)]d\theta dr$$

$$= \int_{r[0,\infty)} \left[\left(d^2 * r * e^{-\frac{r^2}{2\sigma^2}}\right)*2\pi + \left(r^3 * e^{-\frac{r^2}{2\sigma^2}}\right)*2\pi\right] dr$$

$$= 2\pi * d^2 * \int_{r[0,\infty)} r * e^{-\frac{r^2}{2\sigma^2}}dr + 2\pi * \int_{r[0,\infty)} r^3 * e^{-\frac{r^2}{2\sigma^2}}dr$$■

An immediate consequence of Theorem 2 is:

Corollary 2 *(Complexity Order).* If $\boldsymbol{var_x = var_y}$**,** calculating $\boldsymbol{Distance(Point(x_0,y_0),Distribution)}$ is in $\boldsymbol{O(m)}$ where $\boldsymbol{m}$ is the number of intervals for $\boldsymbol{r}$ variable.

We also need a distance metric which takes into account estimated model of spatial noise discussed before.

If there is spatial noise caused by a privacy-protection mechanism, any point inside the sphere could be the starting location of the moving object. This leads to the following definition:

Definition 3 (*Circular Distribution***).** The estimated uncertainty area with its corresponding PDF for an object with a circular initial location is called *circular distribution.*

As before, we need a metric to compare models with different parameters. Because circular distribution is a distribution the same distance metric proposed before remains unchanged. However, in some special cases like when $var_x = var_y$ we can prove the following theorem which helps to compare models with different parameters in the presence of spatial noise.

Theorem 3 If $\boldsymbol{var_x = var_y = \sigma}$ with spatial noise due to a sphere with radius $\boldsymbol{R}$ we have the following:

$$Distance(Point(x_0,y_0),CircularDistribution) = 2\pi\cdot A\cdot\frac{R^3}{3} + 2\pi R\cdot\ (A\cdot d^2 + B)$$

d is the Euclidean distance between $Point(x_0,y_0)$ and the center of the circular distribution, and A and B are as follow:

$$A = 2\pi \int_{r[0,\infty)} r * e^{-\frac{r^2}{2\sigma^2}}$$

$$B = 2\pi * \int_{r[0,\infty)} r^3 * e^{-\frac{r^2}{2\sigma^2}}$$

■

Proof. As shown in Figure 6 an arbitrary point is chosen in the plane. The center of circular distribution and an arbitrary Gaussian distribution from this circular distribution are marked. The distance between point and Gaussian distribution (k') is a function of r' *and* α. Therefore we have the following:

$$Distance\ (Point\ ,CircularDist) = \int_0^R \int_0^{2\pi} (\ A*\ k'(\boldsymbol{r}',\alpha)^2 +\ B)d\alpha d\boldsymbol{r}'$$

$$= 2\pi * A * \frac{R^3}{3} + 2\pi R *\ (A*d^2 + B)$$

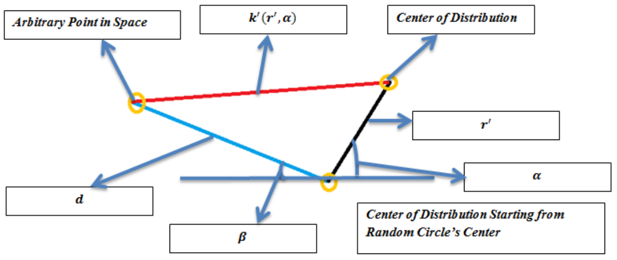

Figure 6. Distance between Point and Circular Distribution

6. Count Queries

In this section, we apply our approach described so far to queries that yield aggregate values; we focus on count queries as a special case of range queries.

6.1 Difficulties

As mentioned before, uncertainty regarding object movements causes uncertain query results. The result of a count query is a number; the result structure so far however is a set of objects with a degree of uncertainty assigned to each of them. In order to compare the query result in the current case with the real situation, the result will again be a probability distribution, and we come up with an appropriate error metric.

6.2 Computing Count Query Results

Think of a count query proposed to count the number of cars inside some region R at time instant t. First, a set of pairs (o_i, p_i), where o$_i$ refers to i-th object and p$_i$ indicates the corresponding uncertainty degree, is calculated like the result of a range query and looks as follows:

$$AnswerSet = \{(o_1, p_1), (o_2, p_2), \dots, (o_n, p_n)\}$$

To generate a probability distribution, we first fix the minimum zero and maximum n , e.g., number of cars in the database, for the interval. We model this as a Poisson binomial distribution. This distribution is the discrete probability distribution of a sum of independent Bernoulli trials that are not necessarily identically distributed. In the real scenario of moving objects, the existence of a certain object is independent of the one of other objects.

To obtain the corresponding Poisson distribution, we consider p_i as probability of object o_i. The probability of having k moving objects out of a total of n in the query result is the sum

$$\Pr(Count = k) = \sum_{A \in F_k} \prod_{i \in A} p_i \prod_{i \notin A} (1 - p_i)$$

F_k is the set of all subsets of k objects out of n objects.

F_k contains $\frac{n!}{k!(n-k)!}$ elements, therefore calculating the sum over the set F_k is impossible in practice unless the total number n of objects is small. To overcome this problem, we employ two different filtering mechanisms to prune irrelevant objects. By irrelevant object, we mean the objects which should not be part of the query result; however their estimated location falls into query region. This is because we deal with uncertainty area of the objects and therefore always there is a possibility for the uncertainty area of the objects to overlap with query area.

With the *minimum threshold filtering mechanism*, there is a predefined threshold th for the uncertainty degree of objects. Any object whose uncertainty degree is less than th is left aside.

With the *maximum number of objects filtering mechanism*, an exogenous parameter specifies the maximum number of objects participating in the query result. In this filtering mechanism, we only let n objects with highest probability to remain in the answer set and the other objects with less probability will be pruned out.

6.3 Evaluation of Results of Count Queries

To compare the effectiveness of estimation models, the solution is to define error metrics which quantify the difference between reality and the estimated number of objects. We use the mean value of the derived distribution to represent it. Based on this, we have the following error metric:

$$ErrMetric1 = |RealNumberofObjects - E(ResultDistribution)|$$

$ResultDistribution$ represents the query result, a discrete distribution over the estimated number of objects.

The mean value does not include the effect of individual elements of the distribution, in some contrast to the following error metric, which computes a weighted average.

$$ErrMetric2 = \sum_{i=1}^{n} |i - RealNumberofObjects| \cdot p_i$$

We will use these metrics to compare estimation models.

7. Experimental Studies

Our general model allows comparing models from different groups as well as models within a group. Using our general model, different models are represented by different parameters of a velocity profile. In the following, we investigate how different parameters, representing different models, affect inherent model accuracy and query result accuracy. In addition, we study the tradeoff between uncertainty in data and quality of query results. In summary, the results show that our general model is well-suited to investigate questions like whether one model is better than another one, and how injection of privacy affects accuracy of query results.

7.1 Experiment Setup

As data set we employ the BerlinMOD [32] data generator to simulate one working day in the city of Berlin, serving as ground truth. The generated data contains 100,000 Home-Work-Home trips between 8:00h and 9:00h a.m. – the time with most trips. Each moving object updates its location at least every 2 seconds, and average trip time in Berlin is about 10 minutes. The data contains 500,000 location updates. Therefore, we choose the trips made during this time interval as the target of our experiments. This choice is crucial, since response time for queries is a major issue due to numerous (two-dimensional) integral computations per moving object. Thus, performance mainly depends on the number of moving objects. If we can handle times when this number is maximal, we can handle so many objects in any other time period as well.

Uncertainty Model Selection. We examine three different models during our experiments, one model from each of the three groups of approaches. See Table 1. The general model will result in a Gaussian distribution, which employs one velocity profile per moving object to estimate its future location. The velocity profiles for each object, i.e., the respective expected value and variance, are learned initially by sampling the movement of an object. The first basic model considers a uniform distribution over an area bounded by maximum velocity of the objects. The second basic model assumes the total direction of movement is available, and based on this, it defines a uniform distribution over the velocity of the object to estimate its future movement.

Table 1. Uncertainty Models

Group	Uncertainty Model	Velocity PDF	Direction
3	General Model (Prop)	Any PDF	Known
1	First Basic Model (Basic1)	None, region bounded	Unknown
2	Second Basic Model (Basic2)	Uniform, maximum speed bounded	Known

Experiments at a Glance. We conduct a series of four experiments with different objectives. The first experiment quantifies inherent model accuracy. The second one examines query result accuracy. Experiment 3 and 4 quantify how the results of the second experiment change in case one injects different forms of privacy.

All the experiments have been implemented in Java, and we have run them on a Linux server with 16 Processors (each one 1400 MHz) and 132 GB main memory.

7.2 Experiment 1: Inherent Model Accuracy

Experiment Objective. We examine the inherent model accuracy for each of the three models in the absence of queries. To this end, for each model, we calculate the deviation of the predicted location and the real location of every moving object and compare them by using the distance metric introduced in Section 5. Generally, the results are relevant as one would expect that more inherently accurate models also deliver higher query-result accuracy – the actual target of the models. However, one must keep in mind that higher inherent accuracy also means less privacy.

Parameters and Procedure. We select 1,000 of the 100,000 objects randomly from the dataset. We assume these objects to announce their initial location and then estimate their location for the next 10 minutes using either model. Then we calculate for each model the average and maximum deviation for each point in time.

Observations. As shown in Figure 7, the average deviation for three models is increasing almost linearly with time, and after 10 minutes the proposed model shows an average deviation of about 4.5 km. Interestingly, the second basic model shows less deviation (about 3.5 km) from the actual path of the object. By contrast, the first basic model shows the largest deviation (about 5.5 km), and hence it is not a good model to estimate the position of the object.

By looking at maximum deviation errors in Figure 8, we see that there is little difference between the proposed model and the second basic model. In our case, the maximum deviation for the proposed model and the second basic model grows linearly with time, and it does not exceed 10 km.

Interpretation. The results indicate that the second basic model features the least deviation from the actual path. So this model is inherently more accurate than the other two models.

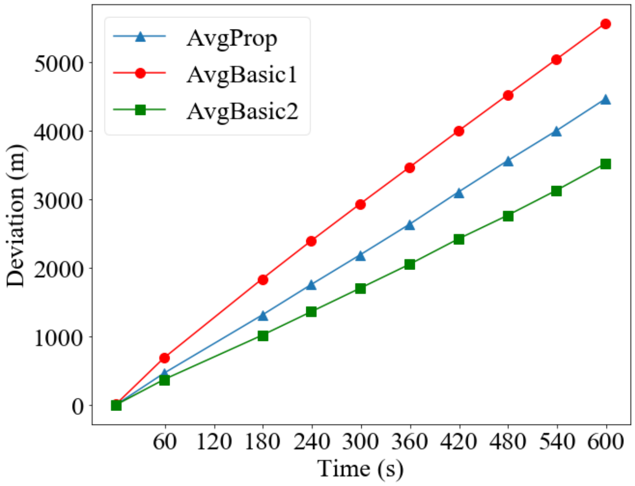

Figure 7. Average Deviation

7.3 Experiment 2: Query-Result Accuracy

Experiment Objective. We are interested whether one can generally expect good query-result accuracy using either model. To this end, we analyze query-result accuracy for all models employing the metrics from Section 6.3. In addition, we study how performance improvements i.e., pruning of very unlikely objects, affects performance and accuracy of the queries. Ultimately, we target at a model which is able to deliver good quality accuracy in short time. In addition, the results from this section are the baseline to study the effects of injecting different forms of privacy in Experiments 3 and 4.

Parameters and Procedure. This experiment consists of 2 sub-experiments. The first one named Experiment 2.1 examines how the ratio of objects not reporting their motion data, but only their initial location and start of movement, named *inactive objects*, affects result accuracy. The parameter of this sub-experiment is the percentage of inactive objects. This is important as we deem it implausible that all objects, e.g., all cars, are willing to participate in a monitoring system collecting fine-grained motion data. So the question examined here is: How many objects need to participate so that one can expect good accuracy. To better understand these numbers, we also investigate how many of the inactive objects would have been part of the query result.

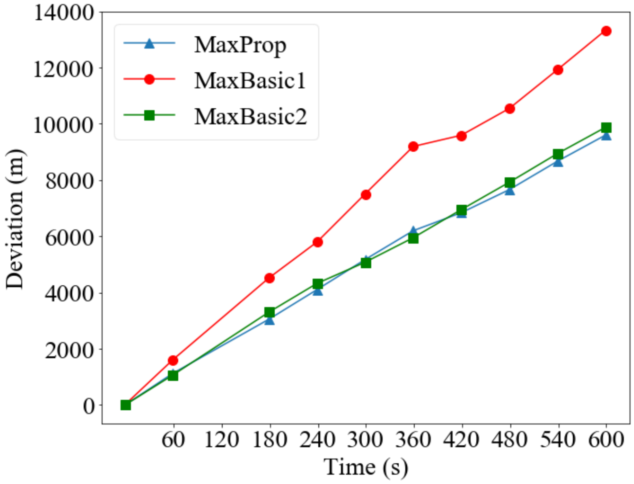

Figure 8. Maximum Deviation

The second sub-experiment, Experiment 2.2, investigates how different filtering mechanisms affect query-response time (performance) and accuracy considering different ratios of inactive objects. The filter thresholds are selected such that interactive query-response times, a major concern in moving object databases, are achieved. That is, all response times are below one second, and the largest response time is close to one second. To this end, we select the minimum threshold to be 10%. Thus, all objects having a probability of less than 10% of being inside the query are eliminated. Our intention is to study what accuracy drop is to be expected compared to Experiment 2.1.

There are additional parameters for both sub-experiments:

- *Query Rectangle.* The width and length are chosen according to the following normal distribution $N((max + min) / 2, (max + min) / 4)$. The rationale is that the objects in a city mostly tend to move towards the center of city.
- *Number of Queries.* We pose one query per minute locating the query rectangle as defined above.
- *Ratio of Inactive Objects:* In each experiment, we start by setting this parameter to 10% of the objects, and we increase it by steps of 10% until covering all objects.

Observation Experiment 2.1: Ratio of Inactive Objects. Figure 9 features the results of this experiment. It graphs the deviation of the number of objects from the real one in relation to the ratio of inactive objects for each model. For example, a number of 2 on the y-axis states that there are, on average, two objects too short or too much for a count-query result. Note that we do not depict the results of the first basic model, in order to ease the presentation. The model has the worst results far away from the other two. It yields about 52 with respect to error metric 1 and 57 with respect to the error metric 2 only when 10% of objects are inactive. To put this into perspective, the other models have a deviation of 2. The deviation grows to 417 and 437 respectively when all objects are inactive. These poor results are because of the number of objects that could be in the area when we have no information regarding their direction and only an upper bound on the velocity. So the resulting uniform distribution covers a wide range of the entire Berlin area for each object.

In contrast, the other two models give almost similar estimations of the number of objects located inside the query in all cases. This shows that the velocity-profile quality of the objects, and the resulting inherent model accuracy, is not the decisive factor. Generally, even when only having access to the direction information of the object, query results can be good. As shown in Figure 9, the deviation from the real value is less than 15 for *all* cases. In addition, the results of the two error metrics are quite close to each other in our experiments. This similarity is due to the settings in our experiments. Hence, from now on we only consider the average query-result accuracy ratio, (cf. Figure 10) for the remaining experiments, for better readability.

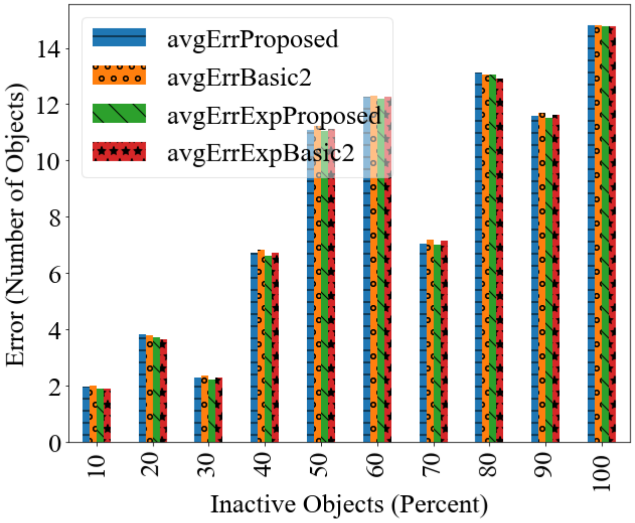

Figure 9. Average Deviation for Count Queries

In order to better understand the numbers presented above, Figure 11 tells us the real number of objects for each experiment. For example, with 10% inactive objects, with private routes there are on average 5 objects inside the query result. It is interesting to observe that the query accuracy in Figure 10 almost has an inverted behavior. This indicates that indeed the inactive objects cause the query inaccuracy. Moreover, even with, say, 60% inactive objects, the result accuracy is almost 80% for the proposed model and the second basic model. We find this remarkable. This indicates that one can generally expect good query accuracy for these two models.

Observation Experiment 2.2: Trade-off between accuracy and performance. Our objective is to achieve interactive query response times, i.e., response time below one second. To achieve this, we use filters which prune very unlikely objects. However, the filters may filter out relevant objects as well. In order to be able to evaluate the filters, we evaluate the ratio of relevant objects that are not filtered out. Obviously, the higher this parameter value, the better is the performance of the filtering mechanism. We have illustrated the results for the first and the second filtering in Figure 12 and Figure 13. Figure 12 indicates the percentage of the relevant objects passing Filter 1. Figure 13 shows the percentage of the relevant objects remaining after *additionally* applying Filter 2. The resulting expected accuracy is the product of the share of remaining objects and the average query result ratio. This ratio is the share of the estimated number of objects over the total number of objects in the query result from Figure 10. As shown in Figure 12, 47% of the relevant objects remain in the best case. In the worst case, only 15% of the relevant objects remain. The resulting accuracy (multiplication of the accuracy with the share of remaining objects) therefore decreases significantly. Still, having an approximate number in short time that gives an intuition is an important use case.

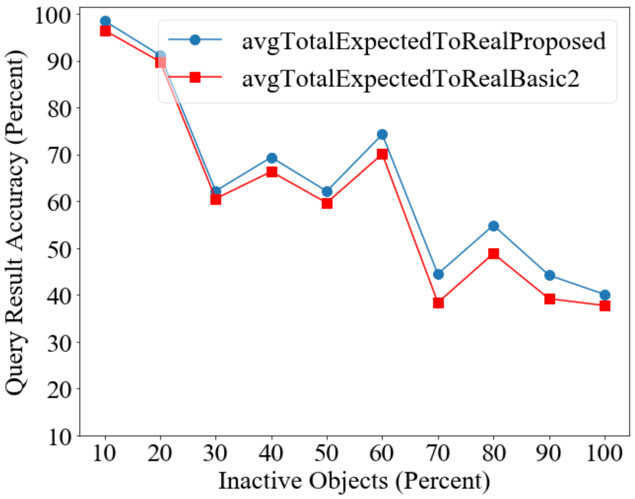

Figure 10. Average Query-Result Accuracy

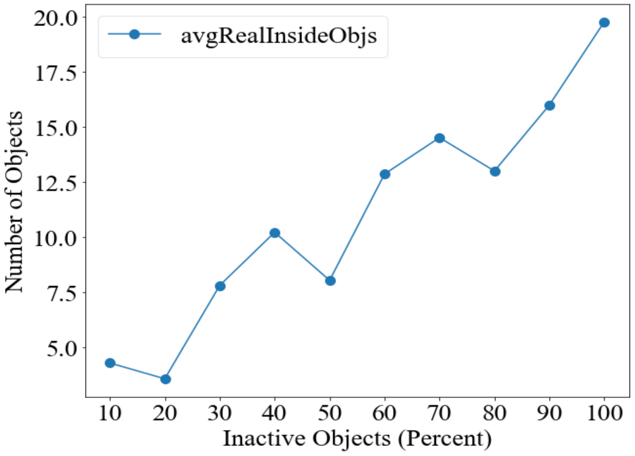

Figure 11. Average Number of Objects in Query Result

Interpretation. The main insight is that, regardless of the fact that the inherent model accuracy (cf. Figure 10. Average Query-Result Accuracy) of the proposed model and the second basic model are different, query-result accuracy is not. The takeaway is that one might want to select the proposed model as it has better privacy, i.e., less inherent model accuracy. Moreover, even for high shares of inactive objects, result accuracy is good, and one can compute results with decent accuracy almost in real time.

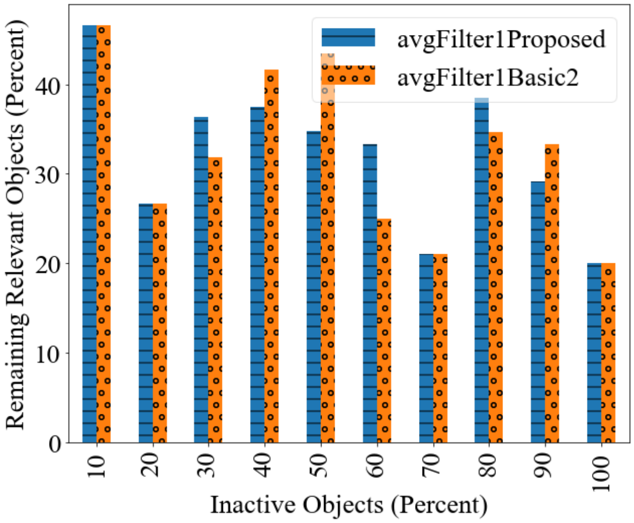

Figure 12. Filter 1 Performance

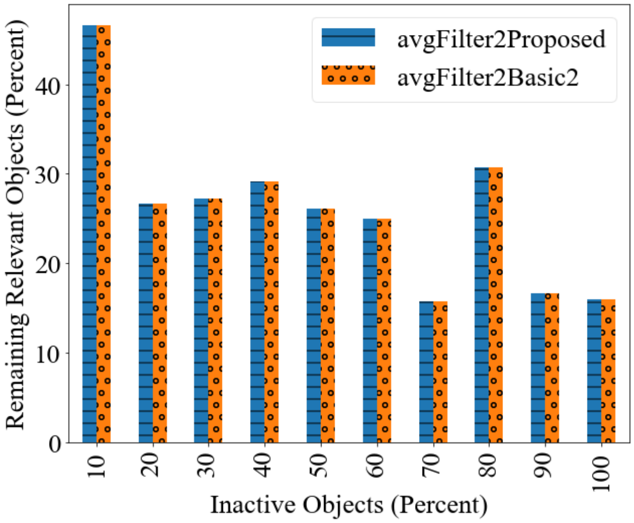

Figure 13. Combined Filter Performance

7.4 Experiment 3: Query Result Accuracy with Spatial Privacy

Experiment Objective. In this experiment, we investigate how the two parameters (1) the ratio of private objects (i.e., objects not reporting their motion data, but only their initial location as a spatial area/a sphere) and (2) the size of the radius used to hide the real position affect the query results. We proceed similarly to Experiment 2. This allows quantifying the effect on query-result accuracy. The only difference is that we now have location privacy in instead of inactive objects. To highlight the difference, we now speak of private objects instead of inactive objects.

Procedure. To study the effect of location privacy, we differentiate between two cases. First, we add a fixed radius to the private objects and vary the ratio of the private objects in Sec. 7.4.1. Second, we keep the ratio of private objects fixed and change the radius (Sec. 7.4.2). Based on the results from Experiment 2, we restrict results to the proposed model and the second basic model to stay within the page limit. One can also interpret these poor results of the first basic model from the perspective of inherent accuracy so that, based on the results from Experiments 1 and 2, one can already decide not to use the first basic model. A clear distinction between the other models in turn is not yet possible. Hence, in the remaining experiments we continue comparing the general model and the second basic model.

7.4.1 Query Result Accuracy Contingent on Spatial Noise with Cloaked Areas of Fixed Size

Parameters. In the previous experiments the private objects have only reported their *exact initial* locations. Now however, each object announces its initial location as a sphere with a radius of 300 m and continues its movement without any location update afterwards. We examine the ratio of objects following this pattern to see how they affect the analysis conducted in Sec. 7.3.

Observations. Ratio of Private Objects. We have compared the proposed model and the second basic model. As for Experiment 2 (cf. Figure 10), Figure 14 plots the query-accuracy ratio. For example, for 10% of the private objects in the city of Berlin, query accuracy is about 70% for both models. In contrast, if all objects are private, the models drop to zero percent accuracy respectively. The results reveal that both models give almost the same results for the number of objects located inside the query in all cases. Generally, even with objects hiding themselves inside a sphere with a radius of 300 m, queries have a decent accuracy.

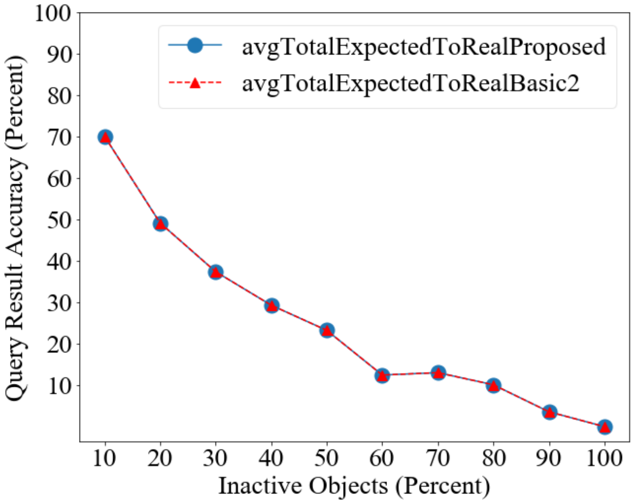

Figure 14. Average Query-Result Accuracy for Fixed Spatial Noise (300 m)

Observation. Trade-off between accuracy and performance. Like in Section 7.3, we deploy a filter to sort out irrelevant objects. In the presence of spatial noise, uncertainty degrees tend to be even less than before. Therefore, employing a minimum threshold filtering mechanism causes a large number of relevant objects to be filtered out. So we only apply Filter 2 in this case. Figure 15 shows its performance.

In some cases, we even see that *all* relevant objects are filtered out. In the best case, only 28% of the relevant objects are part of the final result. So filtering, in order to speed up query evaluation, is not a valid choice in the presence of fixed spatial noise.

This result is particularly relevant as the query-response times generally increase. Recall that we have selected the filter thresholds so that all response times are under 1 s in Experiment 2. Figure 16 graphs the response times for different shares of private objects.

We observe that the response time increases linearly with the ratio of private objects. Generally, the time increases from the millisecond order to the order of seconds. This is because of heavy integral computations over the spatial region of the object. However, even if all objects are private objects, the time does not exceed 21 s. But query accuracy drops significantly.

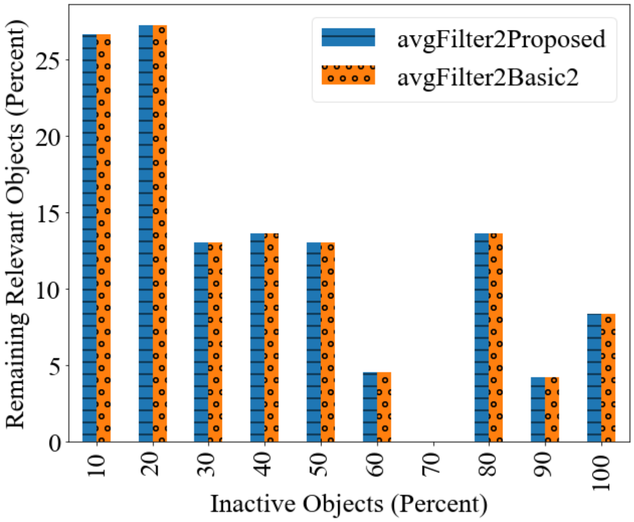

Figure 15. Combined Filter Performance with Spatial Noise

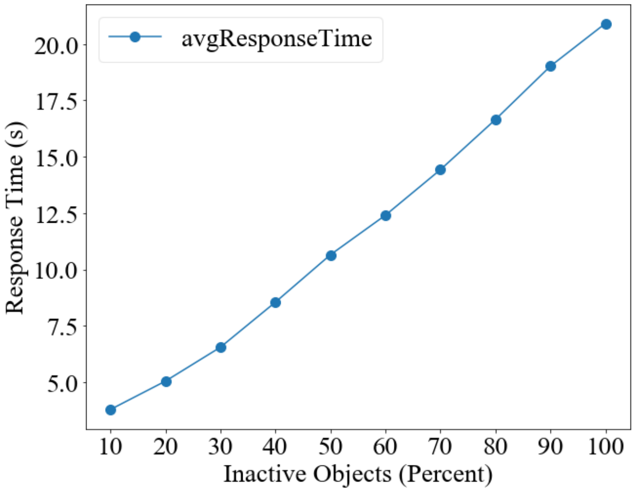

Figure 16. Average Response Time

Interpretation. By introducing this spatial noise, query-result accuracy drops drastically compared to the previous setup. In this scenario (private objects with cloaked areas of fixed size), the additional noise added by the privacy mechanism reduces the effectiveness of filtering mechanism even more than in the scenario where the objects announce their exact initial location.

7.4.2 Query Result Accuracy Analysis Contingent on Spatial Noise of Cloaked Areas of Varying Size

Parameters. In this subsection, we keep the ratio of private objects fixed at 10%, and we change the radius of the covering sphere from 100 m to 1km in steps of 100 m, to study the effect of spatial noise on query-result accuracy.

Observation: Ratio of Private Objects. The results in Figure 17. **Average Query-Result Accuracy** indicate that accuracy is fluctuating between 60 and 70%. In contrast to the results with a fixed radius, accuracy is not monotonically decreasing and does not reach a value of zero. This indicates that accuracy does not primarily depend on the radius, but on the ratio of private objects.

Observation: Trade-off between accuracy and performance. As we can see, the general performance of the filtering mechanism decreases significantly, but is better than using a fixed radius to hide objects. In the best case, 31% of the relevant objects are part of the final result. Interestingly, the average response time is also independent of the radius, having values between 3.6 and 3.8 s. The graph (which we do not explicitly show here) is nearly the same as in Section 7.4.1 for the 10% ratio.

Interpretation. Both estimation models do not deviate too much from reality, and therefore both models can be employed. Comparing the results of this section to the ones of Section 7.4.1 shows that the deviation from the real values using variable noise and a fixed share of inactive objects is much less. So variable noise yields better results compared to a variable ratio of inactive objects. Even when increasing this variable noise, the accuracy remains almost in a fixed range. The average response time also drops drastically in comparison with the previous setup. So one can achieve higher query accuracy if fewer objects are inactive, and they hide in a large area.

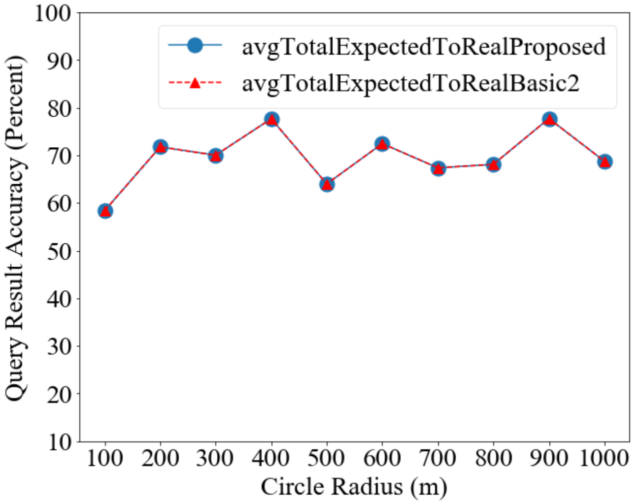

Figure 17. Average Query-Result Accuracy

7.5 Experiment 4: Comparing Spatial and Temporal Privacy

Finally, we compare the effect of temporal and spatial privacy mechanisms. Recall that temporal privacy means decreasing the rate when to report the exact location. To compare both mechanisms, we exploit that temporal privacy also defines a spatial area where the object is located in. So we focus on the size of the area where temporal and spatial privacy mechanisms hide the object. The bigger this area, the better is the privacy. So our results will help the user in selecting an appropriate privacy-protection mechanism.

Parameters and Procedure. We study the effect of temporal privacy by increasing the time period when objects provide location information. The smallest value is 60s. We increase this value by 60s until reaching the average trip length. To compare the results to spatial privacy, we compute the radius of spheres objects are located in. This radius is equivalent to the radius in Experiment 3. For example, if 10% of the objects ”hide” within a radius of 200m, the query accuracy is the same. This is independent of whether one explicitly announces the radius, or it is a result of temporal privacy.

Observation: Coverage Radius Size. Figure 18 shows the average, minimum and maximum hiding radius. As expected, the hiding radius increases when the object does not update its location for a longer time period. To give an intuition on the numbers: The area covered by the average sphere becomes larger than 300km² if none of the objects updates its location for 9 minutes. This is approximately one third of the Berlin area. By taking a closer look at the results, one can see that all three parameters are almost the same for location-reporting intervals of 6 minutes or less. For larger intervals, the minimum and maximum radii start to deviate from the average radius. Nevertheless, since the extent of this deviation from the average is rather small, the method seems to be robust.

Interpretation. The results show that for a time of up to 6 minutes, the average coverage area grows linearly even when the number of private objects increases. In addition, even for small location-reporting intervals such as 60 s, the radius of almost 2 km is large. Thus, any user depending on her privacy preferences could use spatial or temporal privacy according to the results of this experiment, while query-result accuracy remains good. For example, a user who hides her location in a sphere with 4 km radius enjoys the same privacy level as another user who does not update her location for 4 minutes.

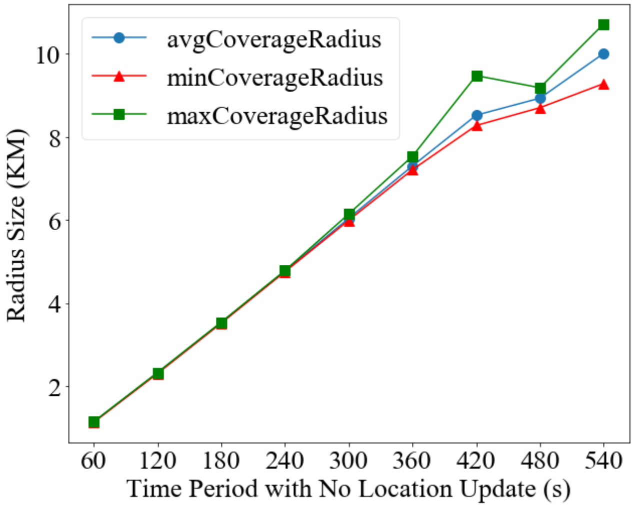

Figure 18. Coverage Radius

7.6 Experiment Summary

Our experiments indicate that for the second basic and the proposed model, good query-result accuracy can be expected in general. This shows the overall validity of our general model. Interestingly, the inherent model accuracy of the proposed model is worse than the inherent accuracy of the second basic model, but query-result accuracy is not. This indicates that one can design models respecting the privacy of the user, in terms of inherent inaccuracy, which nevertheless have good query-result accuracy. Finally, we have shown what accuracy loss due to different forms of injected privacy is to expect. All in all, our approach allows comparing approaches that have not been compared before and, second, gives way to the design of models with the privacy-accuracy challenge resolved.

8. Conclusions and Future Work

In this paper, we have studied how to compare different models estimating the movements of objects, from different perspectives. The location estimations of different models generally are not comparable. We have proposed a general model which captures the notions of time and privacy in a unified manner and allows the comparison of estimation models. Next, we differentiate between two different types of accuracy when comparing models, namely inherent model accuracy and query accuracy. The former considers the deviation of the estimation model from the real path without considering a specific query, in contrast to the latter one. We have shown that an estimation model with higher inherent accuracy may have less accuracy from a query perspective. We have carried out a comprehensive experimental study to study the applicability of the general model and the effectiveness of the error metrics in use. Our results indicate that one can design estimation models respecting privacy, in terms of inherent inaccuracy, which nevertheless have good query-result accuracy.

9. Bibliography

[1] R. Cheng, D. V. Kalashnikov and S. Prabhakar, "Querying imprecise data in moving object environments," *IEEE TKDE*, vol. 16, pp. 1112-1127, 2004.

[2] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao and S. Prabhakar, "Indexing multi-dimensional uncertain data with arbitrary probability density functions," in *Proceedings of the 31st international conference on Very large data bases*, 2005.

[3] D. Pfoser and C. S. Jensen, "Querying the trajectories of on-line mobile objects," in *Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access*, 2001.

[4] Z. Yan, C. Parent, S. Spaccapietra and D. Chakraborty, "A hybrid model and computing platform for spatio-semantic trajectories," in *Extended Semantic Web Conference*, 2010.

[5] B. S. E. Chung, W.-C. Lee and A. L. P. Chen, "Processing probabilistic spatio-temporal range queries over moving objects with uncertainty," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, 2009.

[6] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *International Conference on Pervasive Computing*, 2005.

[7] R. Cheng, Y. Zhang, E. Bertino and S. Prabhakar, "Preserving user location privacy in mobile data management infrastructures," *Lecture Notes in Computer Science*, vol. 4258, pp. 393-412, 2006.

[8] M. F. Mokbel, C.-Y. Chow and W. G. Aref, "The new casper: Query processing for location services without compromising privacy," in *Proceedings of the 32nd international conference on Very large data bases*, 2006.

[9] C. Dwork, "Differential Privacy: A Survey of Results," in *Theory and Applications of Models of Computation*, Berlin, 2008.

[10] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis and C. Palamidessi, "Geo-indistinguishability: Differential privacy

for location-based systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer \& communications security*, 2013.

[11] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider and M. Vazirgiannis, "A foundation for representing and querying moving objects," *ACM Transactions on Database Systems (TODS),* vol. 25, pp. 1-42, 2000.

[12] O. Wolfson, S. Chamberlain, S. Dao, L. Jiang and G. Mendez, "Cost and imprecision in modeling the position of moving objects," in *Data Engineering, 1998. Proceedings., 14th International Conference on*, 1998.

[13] A. P. Sistla, O. Wolfson, S. Chamberlain and S. Dao, "Modeling and querying moving objects," in *Data Engineering, 1997. Proceedings. 13th International Conference on*, 1997.

[14] O. Wolfson, B. Xu, S. Chamberlain and L. Jiang, "Moving objects databases: Issues and solutions," in *Scientific and Statistical Database Management, 1998. Proceedings. Tenth International Conference on*, 1998.

[15] O. Wolfson, A. P. Sistla, S. Chamberlain and Y. Yesha, "Updating and querying databases that track mobile units," in *Mobile Data Management and Applications*, Springer, 1999, pp. 3-33.

[16] J. Bernad, C. Bobed, S. Ilarri and E. Mena, "Handling location uncertainty in probabilistic location-dependent queries," *Information Sciences,* vol. 388, pp. 154-171, 2017.

[17] Y. Tao, D. Papadias and J. Sun, "The TPR*-tree: an optimized spatio-temporal access method for predictive queries," in *Proceedings of the 29th international conference on Very large data bases-Volume 29*, 2003.

[18] G. Kollios, D. Papadopoulos, D. Gunopulos and J. Tsotras, "Indexing mobile objects using dual transformations," *The VLDB Journal, The International Journal on Very Large Data Bases,* vol. 14, pp. 238-256, 2005.

[19] S. Šaltenis, C. S. Jensen, S. T. Leutenegger and M. A. Lopez, Indexing the positions of continuously moving objects, vol. 29, ACM, 2000.

[20] Y. Tao, C. Faloutsos, D. Papadias and B. Liu, "Prediction and indexing of moving objects with unknown motion patterns," in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, 2004.

[21] T. Bernecker, T. Emrich, H.-P. Kriegel, N. Mamoulis, M. Renz and A. Züfle, "A novel probabilistic pruning approach to speed up similarity queries in uncertain databases," in *Data Engineering (ICDE), 2011 IEEE 27th International Conference on*, 2011.

[22] R. Cheng, J. Chen, M. Mokbel and C.-Y. Chow, "Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data," in *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, 2008.

[23] S. Prabhakar, Y. Xia, D. V. Kalashnikov, W. G. Aref and S. E. Hambrusch, "Query indexing and velocity constrained indexing: Scalable techniques for continuous queries on moving objects," *IEEE Transactions on Computers,* vol. 51, pp. 1124-1140, 2002.

[24] K. Zheng, G. Trajcevski, X. Zhou and P. Scheuermann, "Probabilistic range queries for uncertain trajectories on road networks," in *Proceedings of the 14th International Conference on Extending Database Technology*, 2011.

[25] H. M. O. Mokhtar and J. Su, "Universal trajectory queries for moving object databases," in *Mobile Data Management, 2004. Proceedings. 2004 IEEE International Conference on*, 2004.

[26] R. Zhang, H. V. Jagadish, B. T. Dai and K. Ramamohanarao, "Optimized algorithms for predictive range and knn queries on moving objects," *Information Systems,* vol. 35, pp. 911-932, 2010.

[27] H. Fischer, A history of the central limit theorem: From classical to modern probability theory, Springer Science & Business Media, 2010.

[28] J. G. Shanthikumar and U. Sumita, "A central limit theorem for random sums of random variables," *Operations Research Letters,* vol. 3, pp. 153-155, 1984.

[29] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing,* vol. 7, pp. 1-18, 2008.

[30] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st international conference on Mobile systems, applications and services*, 2003.

[31] C. Zhang and Y. Huang, "Cloaking locations for anonymous location based services: a hybrid approach," *GeoInformatica,* vol. 13, pp. 159-182, 2009.

[32] C. Düntgen, T. Behr and R. H. Güting, "BerlinMOD: a benchmark for moving object databases," *The VLDB Journal, The International Journal on Very Large Data Bases,* vol. 18, pp. 1335-1368, 2009.

[33] A. M. Hendawi, J. Bao and M. F. Mokbel, "iRoad: a framework for scalable predictive query processing on road networks," *Proceedings of the VLDB Endowment,* vol. 6, pp. 1262-1265, 2013.

[34] A. M. Hendawi and M. F. Mokbel, "Panda: a predictive spatio-temporal query processor," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012.

[35] A. M. Hendawi, M. Ali and M. F. Mokbel, "Panda∗: A generic and scalable framework for predictive spatio-temporal queries," *GeoInformatica,* vol. 21, pp. 175-208, 2017.