Universal Trajectory Queries for Moving Object Databases*

Hoda M. O. Mokhtar Jianwen Su
Department of Computer Science
University of California at Santa Barbara
{hmokhtar, su}@cs.ucsb.edu

Abstract

In this paper, we consider a data model for uncertain trajectories of moving objects. In our model, the trajectory is a vector of uniform stochastic processes. We study "universal range queries" which examine whether the spatial properties of being inside a region hold throughout an entire time interval. An example of universal range queries is: "Retrieve all trucks staying in Santa Barbara area from 17:00 to 18:00 today." The main technical contributions are efficient algorithms for computing probabilistic answers to universal range queries. We show that the algorithms are efficient using theoretical worst case analysis and empirical studies. Interestingly, the practical complexity is better than theoretical bounds.

1. Introduction

The locations of moving objects are continuously changing [27]. With the rapid advances in wireless communications and ubiquitous computing technologies, applications involving moving objects are fast growing. For example, global positioning systems (GPS) are now widely used in variety of applications on land, at sea, and in the air. Emerging applications include fleet management, localized information, emergency and safety services, network management, etc. Location based service revenues worldwide are expected to grow from approximately $1 billion in 2000 to over $40 billion in 2006 [1]. As a concrete example, automatic vehicle location (AVL) is presently a $650 million industry and expected to exceed $1 billion annually by 2004 [28]. AVL combines both GPS and wireless data networks for the purpose of tracking, monitoring and exchanging information with remote vehicles. Telematics is another rapidly growing real time application. Telematics utilizes wireless communication devices and location sensing-technology and in some cases a service provider to provide safe and convenient services to drivers. It provides emergency roadside assistance, stolen vehicle tracking, automatic crash notification, and navigation assistance. It is estimated that over 20 million telematics-enabled cars and light trucks will be on the road in the United States by 2006 [12], and the industry is expected to grow to at least $5 billion by 2005 [14].

Meeting this demand for applications involving location information opens a door for new research areas. Among the challenges is management and access of continuously moving objects. Over the last few years various problems in moving objects databases have been examined including modeling and query languages [20, 27, 8, 7, 9, 23, 13], handling large volume of location information through the use of efficient index structures [16, 11, 2, 19, 24], efficient data management, specifically, processing queries and handling updates [20, 21], and query evaluation [10, 21, 3].

Much of the earlier work was based on the assumption that exact trajectory information was available (or could be obtained) at every time instant. Unfortunately, this assumption cannot be guaranteed in real applications where trajectory information is associated inherently with uncertainty and lack of complete precise knowledge [15, 17, 25, 5]. For example, although GPS provides reasonably accurate locations, applications typically acquire position data at some (predetermined) time intervals, which means that the exact position between those intervals need to be "predicted". A variety of factors make predicting precise object locations an interesting research problem [15]. Those factors could either be random (such as sudden wind changes, unexpected speed slowdowns, etc.) in some applications [15], or a result of an object not sending instantaneous updates for its current location [15, 25]. Although interpolation techniques are widely available, for critical location-based applications the accuracy of interpolation is not sufficient [18, 29, 4].

Inspired by the importance of this subject, we further study the nature of moving object trajectories in presence of uncertainty. We view the location of a moving object in n dimensional real space at each time instant as a vector of n random variables, and the trajectory of a moving ob-

* Supported in part by NSF grants IIS-0101134 and IIS-9817432.

ject as a vector of stochastic processes. We focus on efficient query evolution in an uncertain trajectory setting.

Handling uncertainty in the context of moving object databases was only investigated very recently. In [15] the authors considered sampling errors in determining object locations and modeling such errors using uniform distributions over a disc with radius as a parameter, and investigated algorithms for answering window queries. The model was extended in [25] to consider queries on trajectories. Quantifiers such as "always", "possibly", "sometimes", etc. and their combinations were developed to indicate different degrees of likelihoods for window queries over trajectories. However, the model cannot provide quantitative reasoning about logical relationships over trajectories, since their approach treats uncertainty as "add-on"s to the data model, rather than as first-class objects.

Query evaluation for uncertain trajectories was investigated in [6, 5], the authors considered time instant queries and presented algorithms for evaluating probabilistic time instant range queries and time instant nearest neighbor queries as well as some probabilistic aggregate queries. Their techniques are only applicable for time instant query but cannot be easily extended to answering time interval queries studied in this paper.

Our model describes uncertainty of trajectories at a time instant using uniform distributions and also allows different dimensions to be constrained in a linear relationship. The latter can be used to represent objects moving along a road network. The focus of the paper is on the evaluation of a class of spatio-temporal queries named "universal range queries". A query in this class states a condition about objects staying inside a region throughout a time interval. Such queries easily arise in moving object database applications.

The technical contributions of this paper are: First, we introduce a data model for uncertain trajectories. The model is based on the use of (time-dependent) uniform distribution. Second, we present a number of algorithms for evaluating universal range queries. Finally, we present complexity results, both theoretical and empirical to show that our algorithms are efficient. Interestingly, while the theoretical complexity bound is linear in the size of a trajectory and the number of dimensions, empirical results show that in practical setting, the algorithms are far more efficient.

Although we focus in this paper on uniformly distributed motions, we also outline how the technique developed here could be extended for evaluating universal range queries over other motion distributions specially the Gaussian distribution.

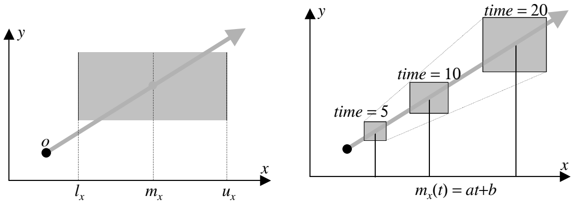
The paper is organized as follows. Section 2 introduces the model for the moving object trajectories and defines universal range queries. Section 3 discusses the basic algorithm for the simple case of 1-dimensional motions. Section 4 extends the algorithm to the general case of n-dimensional motions. Based on these algorithms, Section 5 establishes the complexity bound on evaluating universal range queries trajectories and presents experimental results. Conclusions are provided in Section 6.

2. A Model for Trajectories with Uncertainty

In this section, we introduce a data model for moving object trajectories with uncertainty. Intuitively, a trajectory in this model is represented as stochastic processes. Our model treats each coordinate of a trajectory as a stochastic process with a time-parametric uniform distribution. For a given time instant, the object location is a random variable whose values range over a (closed) interval representing the likely positions. We allow different coordinates of a trajectory to be constrained. We aim at studying efficient query evaluation techniques. We start with the following example which illustrates the main ideas of our model.

For the technical presentation in this paper we assume some familiarity with standard concepts in probability theory (e.g. [22]) including: random variables, probability distributions and in particular uniform distributions, (continuous) stochastic processes and their probability distributions, and independence of random variables, etc.

Example 2.1 Consider an object o moving toward northeast in a 2-dimensional plane as shown in Fig. 1. Due to uncertainty, the location (x_t, y_t) of o at a time instant t cannot be precisely known. One way to model this uncertainty is to treat x_t and y_t as random variables $\bar{x}_t$ and $\bar{y}_t$ (resp.). Assume that $\bar{x}_t$ and $\bar{y}_t$ are independent and have uniform distributions. Fig. 1(a) shows the range $[l_x, u_x]$ of possible values for $\bar{x}_t$ with a mean m_x.



(a) $\bar{x}_t$ at a time instant t

(b) m_x, l_x, u_x are linear functions of t

Figure 1. A trajectory with uncertainty

Since the time t changes, $\bar{x}_t$ is in fact a stochastic process with t as a parameter. The mean m_x and the range $[l_x, u_x]$ are functions of t. Fig. 1(b) shows the case when m_x, l_x, u_x are some linear functions of t with (1) positive coefficients and (2) the coefficient in u_x greater than that in l_x (and similar situation happens in the y-dimension).

In particular, the mean increases thus the object is moving eastwards (or northwards for the y-dimension), and the uncertainty range for the x-dimension increases. ∎

The above example suggests a general framework of modeling uncertainty in moving object trajectories. Specifically, we can extend any distribution to allow the time parameter and use it to represent the uncertain coordinate in one dimension. Although Example 2.1 considers only the case of independent random processes, our model also allows cases when objects move on a road network using a linear relation constraining the stochastic processes for the two dimensions.

In this paper we study trajectories whose uncertainty can be described using uniform distributions. Uniform distributions are simple and easier to reason about in query evaluation. They have been used in most of earlier work dealing with uncertain trajectories including [25, 17, 15, 5].

We assume that time is continuous (represented by real numbers) and fix t to be the time variable. Also, objects move in some $n > 0$ dimensional continuous physical space. We use random variables $\bar{x}_1, ..., \bar{x}_n$ to denote coordinates of object locations at a time instant and stochastic processes $\boldsymbol{x}_1, ..., \boldsymbol{x}_n$ to denote coordinates of object locations that can change over time.

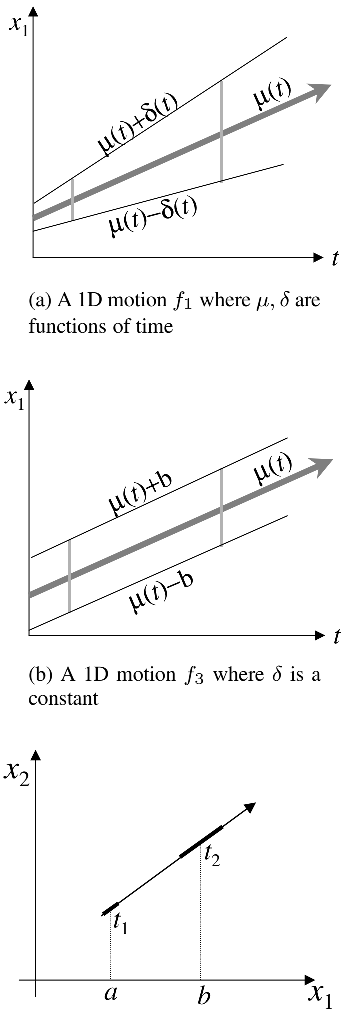
Let u, d be two real numbers where $d \geqslant 0$. We represent the uniform distribution over the interval $[u - d, u + d]$ by the pair (u, d). Note that u is the mean of the distribution. We define a *uniform stochastic (random) process* as a pair (μ, δ) where μ, δ are (continuous, piecewise) linear functions with the time parameter t such that for every real number a, $(\mu(a), \delta(a))$ is a uniform distribution.

Uniform stochastic processes are used to model moving object locations. When an object moves along a road, clearly the coordinates are related through a linear constraint. To model such situations, we represent one coordinate by a linear equation involving the other coordinate. In general, a *(linearly) constrained stochastic process* derived from a set of stochastic processes $S \subseteq \{\boldsymbol{x}_1, ..., \boldsymbol{x}_n\}$ is a linear constraint $\Sigma_{\boldsymbol{x}_i \in S}\, c_i \boldsymbol{x}_i + c_0$, where c_i's are real numbers. A linear constraint is *unary* if all but one coefficients are zero.

Definition: An n-dimensional *motion* is an n-vector $(f_1, ..., f_n)$, where for each $1 \leqslant i \leqslant n$, f_i is either a uniform stochastic process or a constrained stochastic process derived from S, and $S = \{\boldsymbol{x}_j \mid f_j$ is a uniform stochastic process$\}$. A motion is *unconstrained* if it does not contain any constrained processes.

Example 2.2 Consider an object o moving northeast along a road defined by the linear relation: $x_2 = 5x_1 + 3$. Let the position of o at any time instant be given by a pair of stochastic processes $(\boldsymbol{x}_1, \boldsymbol{x}_2)$. Assume that the distribution of $\boldsymbol{x}_1$ is (μ, δ), where $\mu = 2t + 3$ and $\delta = t + 2$ (Fig. 2(a)). Then, the motion of o can be expressed as: $m = (f_1, f_2)$ where $f_1 = (2t + 3, t + 2)$ and $f_2 = (5x_1 + 3)$. Fig. 2(c) shows the constrained motion of o. Note that the uncertainty region is a line.

Consider another object with 1-dimensional motion $m = f_3 = (2t + 3, b)$ (Fig. 2(b)) where $\delta = b$ is a constant. ∎



(a) A 1D motion f_1 where μ, δ are functions of time

(b) A 1D motion f_3 where δ is a constant

(c) A constrained 2D motion

Figure 2. Examples of motions

Incorporating uncertainty in the representation of a trajectory creates an uncertainty volume around the mean trajectory. This uncertainty volume is governed by the linear function δ in the uniform stochastic processes of a motion. In this setting, the uncertainty volume can grow or shrink over time. Increasing uncertainty volume is useful when an object location information is becoming old. Decreasing uncertainty volume is appropriate when an object is moving closer to the time when its location is updated.

Let $m = (f_1, ..., f_n)$ and $m' = (f'_1, ..., f'_n)$ be two motions and a a time instant. The motions m and m' *meet at time a* if for each $1 \leqslant i \leqslant n$,

1. f_i is a uniform stochastic process iff f'_i also is,
2. If f_i and f'_i are uniform stochastic processes, $f_i(a) = f'_i(a)$, and
3. If $f_i = \Sigma_{\boldsymbol{x}_i \in S}\ c_i\boldsymbol{x}_i + c_0$ and $f'_i = \Sigma_{\boldsymbol{x}_i \in S}\ c'_i\boldsymbol{x}_i + c'_0$ where S is a set of uniform stochastic processes, then $\Sigma_{\boldsymbol{x}_i \in S}\ c_ie_i + c_0 = \Sigma_{\boldsymbol{x}_i \in S}\ c'_ie_i + c'_0$ where e_i is the mean of the random variable $\bar{x}_i$ in $f_i(a)$ (or $f'_i(a)$).

Definition: Let k be a nonnegative integer. A *trajectory* is a sequence $(a_0, m_0, a_1, m_1, ..., a_k, m_k)$, where for each $0 \leqslant i \leqslant k$, a_i is a real number or a special symbol in $\{-\infty, \infty\}$ and m_i is a motion such that $a_0 \neq \infty$, $a_0 < a_1 < \cdots < a_k$, and for each $1 \leqslant i \leqslant k$, m_{i-1} and m_i meet at time a_i. A trajectory is *unconstrained* if every motion in it also is.

Example 2.3 Consider an object moving in 1-dimensional space whose trajectory is defined as: $T = (2, (3t+1, t+1), 3, (2t+4, t+2), 4, (-3t+24, t+1), 5, (2t-1, t+1))$. T is illustrated in Fig. 3. ▮

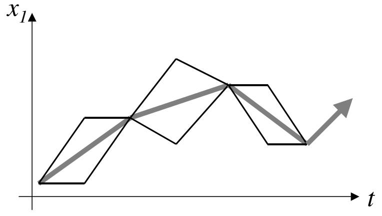


Figure 3. A 1-dimensional trajectory

We assume the existence of an infinite set of object identifiers. A *moving object database* (MOD) is a finite set of pairs (o, T) where o is an object identifier and T a trajectory such that each object identifier is associated to exactly one trajectory. A MOD D is *unconstrained* if every trajectory in D is unconstrained.

In this paper we consider a special type of spatio-temporal queries that we refer to as "universal range queries." Specifically, a universal range query retrieves all objects that have stayed in a region during a time interval.

Example 2.4 Consider a GIS application which also tracks various types of vehicles. Clearly, there are many queries of interest [26] concerning, for example, at a fixed time retrieve the locations of certain kind of vehicles, or during a time interval find vehicles occurring *at least once* inside a region. The following are some queries concerning vehicles *staying* in a region during the *entire* time (assuming the spatial regions are approximated as rectangles):

- Q_1 : *Retrieve delivery trucks that stay in the Santa Barbara area from 5pm to 7pm.*
- Q_2 : *Find police cars that stay on State Street from 10am to 12noon with probability at least 75%.*
- Q_3 : *Find the probability that bus #12 staying inside UCSB campus from 3pm to 3:10pm.* ▮

A common problem in evaluating queries Q_1, Q_2, Q_3 is to provide a probabilistic answer for the spatio-temporal predicates in the queries. For example, if the answer to Q_1 can be computed, the technique for evaluating Q_1 can then be used to compute the probabilities of the corresponding predicate in Q_2 and Q_3.

Note that for a given time instant, a motion (trajectory) is a vector of random variables. Reasoning about the probabilities of events concerning random variables (i.e., for a time instant) can be done using traditional methods in probability theory. Extending to interval properties on continuous random processes is not straightforward. The focus of this paper is on evaluating such universal range queries.

Definition: A *universal range* (UR) query over an n-dimensional MOD is a pair (I, W) where I is a (closed) time interval and W a bounded box (cross product of closed intervals) in the n-dimensional space.

Let PROB$[E]$ denote the probability of occurrence of event E. We denote by $T(\tau)$ a vector of random variables representing a trajectory T at a time instant τ. For each UR query $Q = (I, W)$ and trajectory T, we define the *probability of T satisfying Q* as: $P(T, Q) = \text{PROB}[\forall\tau \in I, T(\tau) \in W]$.

Let D be a MOD and Q a UR query. The *answer* of Q on D is defined as the set:

$$Q(D) = \{(o, p) \mid (o, T) \in D, p = P(T, Q), p > 0\}$$

For convenience, if (o, T) is a moving object in a MOD, we also use $P(o, Q)$ to mean $P(T, Q)$.

For a given UR query $Q = (I, W)$, the key to evaluating Q is to compute the probability of each object o with trajectory T satisfying the query Q, i.e.,

$$P(T, Q) = \text{PROB}[\forall\tau \in I, T(\tau) \in W]$$

3. One Dimensional Motions

In this section we consider evaluation of UR queries in moving object databases. As discussed in Section 2 a key property of UR queries states that spatial window conditions on moving objects hold for every time instant during a given time interval, i.e., universally true for the whole time interval. In this section we focus on UR queries evaluation over a 1-dimensional motion.

When trajectories are precisely given, a UR query can be typically evaluated by *eliminating* trajectories that *fail* to satisfy the condition at *some* time instants, i.e., evaluating a "$(\forall t)$-condition" by removing objects satisfying some "$(\exists t\ \neg)$-condition". In the presence of uncertainty, there are no known algorithms to quantify the results, to the authors' knowledge.

We first motivate the problem of evaluation of UR queries through proposing two naive solutions that although they are simple, they yield incorrect solutions. Next, we discuss our approach towards evaluating UR queries for 1-dimensional motions. In Sections 4 and 5 we use the algorithm for evaluating UR queries over higher dimensional motions and over trajectories.

3.1. Naive Incorrect Solutions

In this subsection we illustrate through examples two approaches for evaluating UR queries for 1-dimensional motions. The examples show that even though the solutions are simple, the answers are incorrect.

Example 3.1 Consider the query Q_1 in Example 2.4, which involves checking if a truck stays in Santa Barbara region (approximated by a rectangular region R) during the whole interval from 5pm to 7pm. Clearly, the truck is in R during the period iff each of its x- and y-coordinates is within the x- and y-projection of R during the time interval.

Consider the x-dimension. Let $[c, d]$ denote the time interval of interest, $[x_1, x_2]$ the x-projection of R, and $\boldsymbol{x}$ be the uniform stochastic process of the x dimension for a delivery truck.

The core of evaluating this query is to compute the probability of the truck's motion completely contained in the region formed by the x-dimension and the time dimension. A naive approach is to compute this probability by the integral:

$$\int_c^d \int_{x1}^{x2} f_{\boldsymbol{x}}\, dx\, dt$$

where $f_{\boldsymbol{x}}$ is the distribution of $\boldsymbol{x}$ at time t. While the computation is simple, it is *wrong*. ▮

The reason for the solution of Example 3.1 being wrong is due to the fact that $\boldsymbol{x}$ is in fact a time-parametric distribution function and describes a stochastic process. The use of $f_{\boldsymbol{x}}$ in Example 3.1 treats $\boldsymbol{x}$ as a two dimensional joint density function. As a consequence, Example 3.1 may result in a value > 1, hence not a valid probability.

Example 3.2 Continuing with Example 3.1. One could attempt a simple fix by turning $f_{\boldsymbol{x}}$ into a joint density function for both $\boldsymbol{x}$ and $\bar{t}$ (the random variable for time). For example, let $f_{\bar{t}} = \frac{1}{d-c}$ be a uniform distribution of $\bar{t}$ over the time interval $[c, d]$. Assuming that $\bar{t}$ and $\boldsymbol{x}$ are independent, we can compute the probability by replacing solution of Example 3.1 with

$$\int_c^d \int_a^b f_{\boldsymbol{x}} f_{\bar{t}}\, dx\, dt$$

However, the solution is still *incorrect*. ▮

There are a few reasons for the incorrectness of Example 3.2. One is that picking a uniform (or any other) distribution for $\bar{t}$ seems arbitrary. More importantly, even with the assumption that $f_{\boldsymbol{x}} f_{\bar{t}}$ is a proper joint density function, Example 3.2 computes the probability of "there exists a time instant in $[c, d]$ when $\bar{x}$ is in $[a, b]$" rather than the probability of "for each time instant in $[c, d]$, $\bar{x}$ is in $[a, b]$" as required by the query. Furthermore, Example 3.2 totally disregards the constraint that each trajectory has to be continuous.

Examples 3.1 and 3.2 suggest that probabilistic evaluation of UR queries in Example 2.4 is fundamentally different from time instant queries. This motivates the need for different techniques for evaluating universal range queries.

In the next subsection we present a general technique for evaluating 1-dimensional UR queries for moving objects with uncertain trajectory information.

3.2. Complexity Result

The previous subsection motivated the need for an alternative efficient approach that correctly answers UR queries. In this subsection we present our technique for evaluating those queries. The main result of this section is the following complexity result.

Theorem 3.3 Let $Q = (I, W)$ be a 1-dimensional UR query, m a 1-dimensional motion of an object o, $P(o, Q) = \textsc{Prob}[\forall \tau \in I, m(\tau) \in W]$ can be computed in $O(1)$ time (i.e., constant time).

We discuss the idea of proofing Theorem 3.3 below and outline a proof in Section 3.3.

If E_1, E_2 are events, we denote by $\textsc{Prob}[E_1 \mid E_2]$ the conditional probability of occurrence E_1 given E_2.

We assume without loss of generality that $W = [l, u]$. Therefore, we need to compute the probability $\textsc{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)]$, where $\boldsymbol{x}_1(\tau)$ is the random variable denoting the motion along x_1 at time τ.

The main idea for computing $\textsc{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)]$ is summarized as follows. We divide the time interval I into small slices of width Δ, and compute the conditional probability P_τ of the object o staying within the window $[l, u]$ for a small time slice $[\tau, \tau + \Delta]$ of I if o starts from a point inside $[l, u]$. We denote the random process at time $\tau + \Delta$ by $\boldsymbol{x}_1(\tau + \Delta)$. In other words, P_τ is defined as follows:

$$P_\tau = \text{Prob}[l \leqslant \boldsymbol{x}_1(\tau + \Delta) \leqslant u \mid l \leqslant \boldsymbol{x}_1(\tau) \leqslant u].$$

We approximate the motion of o within the time slice $[\tau, \tau + \Delta]$ by a straight line. The probability P_τ can be computed by considering each point x_τ, inside the intersection of the query window $[l, u]$ and the range of $\boldsymbol{x}_1(\tau)$, and all the possible angles for o to move from this point that results in a point $x_{1,\tau+\Delta}$ in the intersection of $[l, u]$ and the range of $\boldsymbol{x}_1(\tau + \Delta)$ (i.e., at the time $\tau + \Delta$). Informally, $\text{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)]$ is then the product $\Pi_{i \geqslant 0}^{|I|/\Delta} P_{\tau_i}$ where τ_0 is the lower bound of I and $\tau_{i+1} = \tau_i + \Delta$.

In the following subsection we use the above relations to develop an evaluation technique for UR queries.

3.3. An Evaluation Technique

Using the above idea we obtain formulas for computing the probability of 1-dimensional motions m satisfying Q, $P(o, Q)$, in the following theorem.

Theorem 3.4 Let (I, W) be a 1-dimensional UR query and o an object with a uniform 1-dimensional motion (μ, δ), where $I = [I_{\text{low}}, I_{\text{high}}]$, and $W = [l, u]$. Furthermore, let $\boldsymbol{x}_1$ denote the stochastic process for the motion, $[L_{\max}, U_{\min}] = W \cap [\mu - \delta, \mu + \delta]$, and $|I|$ be the length of I. The probability of o satisfying Q is given below:

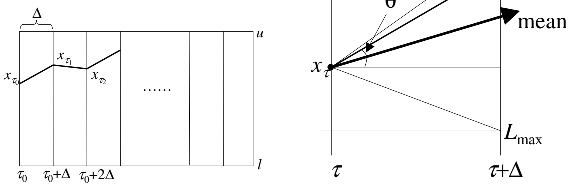
1. If δ, is not constant,

$$P(o, Q) = \text{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] \\ = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \cdot \frac{e^{2|I|} \cdot ((U_{\min} - L_{\max}))^{2|I|}}{\left(\frac{(2aI_{\text{high}} + 2b)^{I_{\text{high}} + \frac{b}{a}}}{(2aI_{\text{low}} + 2b)^{I_{\text{low}} + \frac{b}{a}}} \right)^2}$$

2. If δ is a constant,

$$P(o, Q) = \text{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] \\ = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \left(\frac{U_{\min} - L_{\max}}{2\delta} \right)^{2|I|}$$

In general, Theorem 3.4 gives a general formula for computing the probability of an object staying within a query window W during a time interval $I = [I_{\text{low}}, I_{\text{high}}]$. The formulas in Theorem 3.4 are basically the product of 2 terms. Since the motion is uniform, the first term $\frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})}$ is the probability of the object being initially inside W at the start of the time interval I_{low}, i.e., $\text{Prob}[l \leqslant \boldsymbol{x}_1(I_{\text{low}}) \leqslant u]$.



(a) Time slices division

(b) Staying within during $\tau, \tau + \Delta$

Figure 4. Illustrating the time slicing technique

The following discussion focuses on the second term, which captures the probability of assuming the object starts within W, it stays inside W for the entire interval.

An observation is that for a unit time interval ($|I| = 1$), the probability of the object staying within W is basically the ratio of the valid query window (i.e. intersection of x-dimension range and x-dimension projection of the query window) to the range of the x-distribution (i.e. $[\mu - \delta, \mu + \delta]$). Such observation is reasonable due to uniform distribution.

We now first show in more details our approach for deriving the second term in the above result. Then we present different cases that represent the different possible spatial relationships between the motion and the query window. Those cases are derived from the above theorem.

Let $x_{1,\tau}$ be some position of o at the time τ such that $L_{\max} \leqslant x_{1,\tau} \leqslant U_{\max}$. Let $\boldsymbol{x}_1(\tau + \Delta)$ be the random variable for the position of o at the time $\tau + \Delta$. We assume the trajectory motion of o within the time slice to be a straight line. Let $\bar{\theta}$ be a random variable for the angle of o moving from $x_{1,\tau}$ to $x_{1,\tau+\Delta}$. Let I_{low} denote the lower bound of the time interval I. Our approach proceeds as follows:

We start by dividing the time interval I into N small slices of width Δ where $\Delta = I/N$. Since o moves along a straight line in the interval $[\tau, \tau + \Delta]$, and $\bar{\theta}$ is a random variable for the angle of o moving from $x_{1,\tau}$ to $x_{1,\tau+\Delta}$.

- We define $\bar{\Theta} = \tan(\bar{\theta}) = \frac{\bar{x}_{1,\tau+\Delta} - x_{1,\tau}}{\Delta}$. (Note that we are only interested in $\bar{\theta}$ values that define the possible angles for the motion so that o remains inside the window $[L_{\max}, U_{\min}]$ during the time slice.)
- Since the o moves with motion ($\boldsymbol{x}_1$ denotes the random process with uniform distribution function given by (μ, δ)), the distribution $f_{\bar{\Theta}}$ of $\bar{\Theta}$ during a time slice Δ is

$$f_{\bar{\Theta}} = \begin{cases} \frac{\Delta}{2\delta} & \text{if } \frac{(\mu - \delta) - x_1}{\Delta} \leqslant \Theta \leqslant \frac{(\mu + \delta) - x_1}{\Delta} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Having obtained the distribution for $\bar{\Theta}$, we next consider each Δ and compute the conditional probability, denoted as P_τ of the object o staying within the window $[l, u]$ for a $[\tau, \tau + \Delta]$ of I if o starts from a point in $[l, u]$. In other words, P_τ is defined as

$$P_\tau = \textsc{Prob}[l \leqslant \boldsymbol{x}_1(\tau + \Delta) \leqslant u \mid l \leqslant \boldsymbol{x}_1(\tau) \leqslant u]$$

Thus P_τ can be expressed as follows:

$$P_\tau = \int_{x_1 = L_{\max}}^{x_1 = U_{\min}} \int_{\Theta = \frac{L_{\max} - x_1}{\Delta}}^{\Theta = \frac{U_{\min} - x_1}{\Delta}} f_{\bar{\Theta}} f_{\bar{x}_1} \, d\Theta \, dx_1 \tag{2}$$

where $f_{\bar{x}_1} = \frac{1}{2\delta}$ is the distribution for $\bar{x}_1$. Note that $L_{\max}$ and $U_{\min}$ may be functions of time.

Plugging Equation (1) into Equation (2) above, and with simple mathematical manipulations we get:

$$\begin{aligned} P_\tau &= \int_{x_1 = L_{\max}}^{x_1 = U_{\min}} \int_{\Theta = \frac{L_{\max} - x_1}{\Delta}}^{\Theta = \frac{U_{\min} - x_1}{\Delta}} \frac{\Delta}{2\delta} \frac{1}{2\delta} \, d\Theta \, dx_1 \\ &= \frac{(U_{\min} - L_{\max})^2}{(2\delta)^2} \end{aligned} \tag{3}$$

Now as we have the probability of staying within each slice, we compute the probability of staying within the window $[L_{\max}, U_{\min}]$ for the whole time interval I. This can be obtained by the following mathematical manipulations:

$$\begin{aligned} &\textsc{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] \\ &\quad = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \cdot \prod_{i>1}^{|I|/\Delta} P_{\tau_i} \\ &\quad = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \cdot e^{\ln \prod_{i>0} P_{\tau_i}} \\ &\quad = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \cdot e^{\sum_{i>0} \ln P_{\tau_i}} \end{aligned} \tag{4}$$

where $\tau_{i+1} = \tau_i + \Delta$ as defined earlier and $\frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})}$ is the probability of the object being initially inside the query window at time I_{low}. Let E_i be the event of condition $l \leqslant \boldsymbol{x}(\tau_i) \leqslant u$ is satisfied. *Then*,

$$P_{\tau_i} = \textsc{Prob}[E_i \mid E_{i-1}] \quad \textit{and}$$

$$\begin{aligned} &\textsc{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}(\tau) \leqslant u)] \\ &\quad = \textsc{Prob}[E_0] \cdot \prod_{i>0} \textsc{Prob}[E_i \mid E_0 E_1 \cdots E_{i-1}] \end{aligned}$$

Since the trajectory within a slice only depends on the starting point, E_i, E_j are conditionally independent for $|i - j| \geqslant 2$. Equation (4) follows from this.

Finally, since we are interested in the continuous case, taking the limit as $\Delta \to 0$, Equation (4) can be expressed in a continuous form as:

$$\begin{aligned} &\textsc{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] \\ &\quad = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \cdot e^{\int_I \ln(\lim_{\Delta \to 0} P_\tau) d\tau} \end{aligned} \tag{5}$$

Following the result of Equation (3), it is clear that $\lim_{\Delta \to 0} P_\tau = P_\tau$. Based on this analysis and together with some mathematical manipulations, Theorem 3.4 can be proved.

Although the general case is given in Theorem 3.4, it is also useful to consider special cases based on the the spatial relationships between the query window W and the range of the distribution R. There are 6 possible cases depending on the spatial configuration between W and R, assuming the boundaries of W and R do not intersect except for endpoints. However, only 3 cases are actually interesting. Two other cases correspond to $W \cap R = \varnothing$ which yield probability 0. And the last case is symmetric to one of the first three cases.

Case 1 shown in Fig. 5 is the most trivial case. In this case $R \subseteq W$, and the formula in Theorem 3.4 yields 1. This is correct since the object stays within W for the time interval I, i.e., object o always satisfies the UR query (I, W). Case 2 shown in Figure 5 is when W is totally enclosed in R. Therefore, $L_{\max} = l$ and $U_{\min} = u$, where $W = [l, u]$. Case 3 shown in Fig. 5 is when W is partially enclosed in R. Therefore, $L_{\max} = f(t)$ and $U_{\min} = u$, where $W = [l, u]$ and $R = [f(t), g(t)]$. (There is a symmetric case for case 3.)

In the remainder of this section we give further discussion for Cases 2 and 3. Remember that (I, W) is a 1-dimensional UR query and o an object with uniform 1-dimensional motion m.

Corollary 3.5 Consider Case 2 of Fig. 5 where $L_{\max} = l$ and $U_{\min} = u$. Let $\boldsymbol{x}_1(\tau)$ be the random variable of m with a uniform distribution function at time τ defined by (μ, δ), where μ and δ are linear functions of time. Letting $\delta = at + b$, we have

$$\begin{aligned} &\textsc{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] \\ &\quad = \frac{|W|}{g(I_{\text{low}}) - f(I_{\text{low}})} \cdot \frac{e^{2|I|} \cdot |W|^{2|I|}}{\left(\frac{(2aI_{\text{high}} + 2b)^{I_{\text{high}} + \frac{b}{a}}}{(2aI_{\text{low}} + 2b)^{I_{\text{low}} + \frac{b}{a}}} \right)^2} \end{aligned}$$

The corollary is obtained by plugging the value $|W|$ into the term $U_{\min} - L_{\max}$ in Theorem 3.4.

Corollary 3.6 Consider Case 2 of Fig. 5 again where $L_{\max} = l$ and $U_{\min} = u$. Let $\boldsymbol{x}_1(\tau)$ be a random variable with a uniform distribution defined by (μ, δ) at time τ, where μ is a linear function of time and δ is a constant. Then,

$$\begin{aligned} P(m, Q) &= \textsc{Prob}[\forall \tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] \\ &= \frac{|W|}{g(I_{\text{low}}) - f(I_{\text{low}})} \left(\frac{|W|}{2\delta} \right)^{2|I|} \end{aligned}$$

The above corollary is also obtained by plugging the value $|W|$ into the term $U_{\min} - L_{\max}$ in Theorem 3.4.

Corollary 3.7 Consider Case 3 of Fig. 5 where $L_{\max} = f(t)$ and $U_{\min} = u$. Let $\bar{x}_1(\tau)$ be the random variable of motion m with a linear uniform distribution function at time τ defined by (μ, δ). Letting $\delta = at+b$ and $u-f(t) = ct+d$, we have

$$P(m,Q) = \textsc{Prob}[\forall\tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] =$$

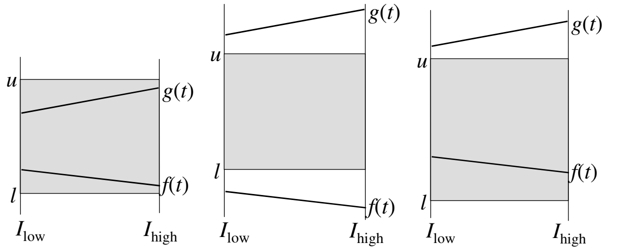
$$\frac{u-f(I_{\text{low}})}{g(I_{\text{low}})-f(I_{\text{low}})} \cdot \frac{[(cI_{\text{high}}+d)^{2I_{\text{high}}+\frac{2d}{c}}][(aI_{\text{low}}+b)^{2I_{\text{low}}+\frac{2b}{a}}]}{[(cI_{\text{low}}+d)^{2I_{\text{low}}+\frac{2d}{c}}][(aI_{\text{high}}+b)^{2I_{\text{high}}+\frac{2b}{a}}]}$$

The corollary is obtained by replacing $U_{\min} - L_{\max}$ in Theorem 3.4 by $u - f(t)$, where u is the top bound of W, and $f(t)$ is a linear function defining the lower bound of the distribution.

For the case of constant δ the result is a simplified form of the one proved for Corollary 3.7. The only difference is that δ is now a constant:

$$\textsc{Prob}[\forall\tau \in I(l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] =$$

$$\frac{u - f(I_{\text{low}})}{g(I_{\text{low}}) - f(I_{\text{low}})} \cdot \frac{(cI_{\text{high}} + d)^{2I_{\text{high}}+\frac{2d}{c}}}{((2\delta)^{2|I|})(cI_{\text{low}} + d)^{2I_{\text{low}}+\frac{2d}{c}}}$$



(a) Case1 (b) Case 2 (c) Case 3

Figure 5. Spatial relationships between query window and trajectory motion region

Using the above special cases we can now compute the probability of any motion staying within a query window $P(m, Q)$. This can be obtained by dividing the time interval into several intervals causing the motion in each interval to resemble one of the cases considered above. Theorem 3.3 follows easily. We use the following example to illustrate the main idea.

Example 3.8 Consider the spatial relation between W and the range of the distribution R shown in Fig. 6. To compute the probability of the object staying within W during $I = [I_{\text{low}}, I_{\text{high}}]$, we use the spatial cases defined above. The time interval is thus divided into the shown set of motions (i.e. $\{m_1, m_2, m_3\}$). To compute the probability in this case, we simply apply the results for cases 1, 3 and 2 resp. ∎

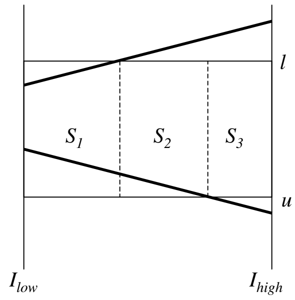


Figure 6. Compound spatial relationships

The above discussion proves the correctness of Theorem 3.3 since the results show that for computing the probability of any 1-dimensional motion satisfying a 1-dimensional UR query we need to apply any combinations of the above cases at most 3 times. Since each case takes only a constant time $O(1)$, the overall computational complexity is thus $O(1)$. Therefore, we can efficiently compute $P(m, Q)$ for 1-dimensional uniform motions and 1-dimensional UR queries.

3.4. An Extension for Gaussian Motions

In the above discussion we focused on motions whose distributions are defined to be uniform. In this subsection we show that the previous technique is also applicable for objects whose motion could be defined by any probability distribution function, specifically *Gaussian* distributions.

We define a *Gaussian stochastic (random) process* as a pair (μ, σ) where μ is the mean of the distribution and σ is the standard deviation, both μ, σ are (continuous, piecewise) linear (real) functions with the time parameter t such that for every real number a, $(\mu(a), \sigma(a))$ is a Gaussian distribution.

We define an n-dimensional *Gaussian motion* as an n-vector $(f_1, \cdot\cdot\cdot, f_n)$ where for each $1 \leqslant i \leqslant n$, f_i is either a Gaussian stochastic process or a constrained Gaussian stochastic process from S, and $S = \{x_j \mid f_j \text{ is a Gaussian stochastic process}\}$. A motion with no constraints is an *unconstrained* motion.

A *trajectory* is defined just similar to its definition in Section 2 except that here each motion is a Gaussian motion.

Continuing with the same definition of the random variables and with the same derivation steps applied in the previous subsection, our goal is to find analogous equations for Equations (1) and (3) for the Gaussian motion. The resulting new equations can then be used to obtain an answer for UR queries as proposed in the previous subsection. Thus the main objective is to first use Equation (2) that express $\textsc{Prob}[l \leqslant x_1(\tau + \Delta) \leqslant u \mid l \leqslant x_1(\tau) \leqslant u]$. Then we plug into it the new definition for Equation (1), so that we end up with an equation similar to Equation (3). Finally, we follow the same approach to obtain the result for the continuous case using the result of Equation (5).

Under the assumption of Gaussian motion, the distribution function for the Gaussian stochastic random variable $\bar{x}_1$ can be expressed by the Gaussian distribution function as:

$$f_{\bar{x}_1} = \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}(\frac{(x_1-\mu)}{\sigma})^2} dx_1 \tag{6}$$

Since we have $\bar{\Theta} = \tan(\bar{\theta}) = \frac{\bar{x}_{1,\tau+\Delta} - x_{1,\tau}}{\Delta}$. We obtain a definition for the distribution of $\bar{\Theta}$ to be:

$$f_{\bar{\Theta}} = \frac{\Delta}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}(\frac{\theta\Delta + x_1 - \mu}{\sigma})^2} \tag{7}$$

Following the above guide lines, we get the following main result for *Gaussian* motions.

Theorem 3.9 Let (I, W) be a 1-dimensional UR query and o an object with a Gaussian 1-dimensional motion $m = (f)$ where $f = (\mu, \sigma)$, $I = [I_{\text{low}}, I_{\text{high}}]$, and $W = [l, u]$. Furthermore, let $\boldsymbol{x}_1$ denote the stochastic process for the motion, $[L_{\max}, U_{\min}]$ are the window boundaries, and $|I|$ be the length of I. The probability of o satisfying Q is given below:

$$P(o, Q) = \textsc{Prob}[\forall \tau \in I (l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \cdot \exp\left(\int\limits_I [\ln(\phi - \psi + \omega)]\, dt \right)$$

where,

$$\phi = \frac{1}{2\sqrt{\pi}} \int\limits_{y=0}^{\frac{1}{\sqrt{2}}(\frac{-U+\mu}{\sigma})^2} e^{-y^2} dy$$

$$\psi = -\frac{2}{\pi} \int\limits_{y=0}^{\frac{1}{\sqrt{2}}(\frac{-U+\mu}{\sigma})} e^{-y^2} dy \cdot \int\limits_{w=0}^{\frac{1}{\sqrt{2}}(\frac{-L+\mu}{\sigma})} e^{-w^2} dw$$

$$\omega = \frac{1}{2\sqrt{\pi}} \int\limits_{y=0}^{\frac{1}{\sqrt{2}}(\frac{-L+\mu}{\sigma})^2} e^{-y^2} dy$$

This result follows from plugging Equations (6) and (7) into Equation (2). Then through the use of mathematical manipulations and error function (i.e. as erf) [22], we obtain a formula for $P[o, Q]$ that is free of Δ. Thus substituting in Equation (5), the $\lim_{\Delta \to 0}$ can be eliminated and the above result is deduced.

Corollary 3.10 Under the same assumptions in Theorem 3.9 and in addition assuming that μ and σ are constants, then,

$$P(o, Q) = \textsc{Prob}[\forall \tau \in I (l \leqslant \boldsymbol{x}_1(\tau) \leqslant u)] = \frac{U_{\min} - L_{\max}}{\text{range of } \boldsymbol{x}_1(I_{\text{low}})} \cdot (\phi + \psi + \omega)^{|I|}$$

where, ϕ, ψ, and ω are as defined in Theorem 3.9.

Although in this subsection we illustrated the case of *Gaussian* motions, the technique is still valid for arbitrary motion distributions as long as a distribution for $\bar{\Theta}$ can be obtained through transformation of motion random variables [22]. Then the new distribution can be simply plugged into Equations (2) and (5) to get the result for UR query for those motions.

4. Multi-dimensional Motions

In the previous section we only considered the simplest case of a 1-dimensional motion for moving objects. In this section we extend our results to single motions in n-dimensional space. Specifically, we show how to evaluate UR queries for n-dimensional motions, i.e., computing the probability. We consider two different cases of moving object motions, unconstrained motions and motions with unary constraints, respectively.

4.1. Unconstrained Motions

Recall that a motion is unconstrained if it does not contain any constrained stochastic processes. Thus, an unconstrained motion corresponds to an object moving in an n-dimensional space where the uncertainty along each dimension is independent of other dimensions.

Due to the pairwise independency of the n dimensions, Theorem 3.4 can be easily extended to n-dimensions. It is known that the probability of a set of independent random variables is the product of the probabilities of the random variables individually. We have:

Theorem 4.1 Let (I, W) be a UR query and $m = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ an unconstrained motion in n-dimensional space. If W_i denotes the projection of W on the i^{th}-dimension,

$$\textsc{Prob}[\forall \tau \in I, m(\tau) \in W] = \prod_{1 \leqslant i \leqslant n} \textsc{Prob}[\forall \tau \in I, \boldsymbol{x}_i(\tau) \in W_i].$$

Furthermore, the query can be evaluated in $O(n)$ time.

4.2. Motions with Unary Constraints

We consider the two dimensional case with one coordinate derived from the other; generalization to n dimension is straightforward. In this case, let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two random processes corresponding to the two coordinates of a motion in which $\boldsymbol{x}_2$ is constrained by $c_1\boldsymbol{x}_1 + c_0$. Recall that this can also be understood as whenever $\boldsymbol{x}_1$ has a value, the value for $\boldsymbol{x}_2$ is also fixed. The linear constraint relation can be easily used for modeling the trajectory of a vehicle moving along a road.

In the general case, an n-dimensional motion $(\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ with unary constraint consists of uniform stochastic processes and constrained stochastic processes where only one coefficient is nonzero. In other words, the constraints are all of the form $c_i\boldsymbol{x}_i + c_0$ where $\boldsymbol{x}_i$ is a uniform stochastic process.

Let (I, W) be a UR query and m a motion with unary constraints in n dimensional space. Consider each constraint $\boldsymbol{x}_j = c_i\boldsymbol{x}_i + c_0$ in m. For m to satisfy the UR query condition, the event $\boldsymbol{x}_j \in W_j$ must hold where W_j is the projection of W on j^{th} dimension. Define $C_j^i = \{z \mid c_i z + c_0 \in W_j\}$ and it is easy to see that C_j^i is an interval. Therefore the intersection $W_i' = W_i \cap (\bigcap_j C_j^i)$ defines the interval for $\boldsymbol{x}_i$ which is equivalent to all query conditions on $\boldsymbol{x}_i$ and on constrained processes derived from $\boldsymbol{x}_i$. The following is immediate:

Theorem 4.2 Let (I, W) be a UR query and $m = (\boldsymbol{x}_1, ..., \boldsymbol{x}_n)$ a motion with unary constraints in n dimensional space. Suppose W_i' is the intersection defined in the above discussion.

$$\textsc{Prob}[\forall \tau \in I, m(\tau) \in W] = \prod_{\boldsymbol{x}_i \text{ is a uniform stochastic process}} \textsc{Prob}[\forall \tau \in I, \boldsymbol{x}_i(\tau) \in W_i'].$$

Furthermore, the query can be evaluated in $O(n)$ time.

Example 4.3 Consider a car moving on a road defined as $x_2 = x_1 + 2$. Consider a UR query with window $W = [3, 10] \times [6, 10]$. In this case, $W_1 = [3, 10]$ and $W_2 = [6, 10]$. For the car to stay in W_2, its x_1 value has to be in the interval $[4, 8]$. Thus $W_1' = W_1 \cap [4, 8] = [4, 8]$. ▮

Example 4.4 Consider a car traveling from Santa Barbara to Berkeley. Assume that there exist 4 different routes that can be followed to reach the destination. However, following any of those routes will cause the driver to pass through a desert area (approximated by a rectangle). Assume a driver wants to choose a route that minimizes his/her chances of being in the desert area from 2pm to 3pm. We can use the technique in Section 4.2 to evaluate this query, since the routing information can be viewed as trajectories. ▮

5. Evaluating Universal Range Queries

In this section we consider the evaluation of UR queries over unconstrained trajectories and trajectories with unary constraints. The technical results concern the computation of the probability of a trajectory satisfying a UR query. We give two types of results on complexity. On the theoretical side, we show that the complexity of evaluating one such trajectory is linear in both the numbers of dimensions and motions in the trajectory. We also conduct an experimental study and show that in practical settings, the complexity is actually better: the number of probability computations is sub-linear in the number of motions and/or dimensions.

Theorem 5.1 Let T be a trajectory with k motions and Q a UR query in n-dimensional space. If T includes no constraints or only unary constraints, the probability $P(T, Q)$ can be computed in $O(nk)$ time.

The idea of the proof is to consider motions individually and apply Theorems 4.1 and 4.2 in a straightforward manner. The probability of the trajectory is then the product of that for all motions.

Although the linear complexity bound indicates efficient computation, there is room to improve further for actual algorithms. For a given UR query Q, we consider the times $P(m, Q)$ is computed where m is a 1-dimensional motion. Clearly if the uncertainty region of m is disjoint from or totally contained in the query window, $P(m, Q)$ is simply 0 or 1 (respectively). Moreover, in the computation of $P(T, Q)$ if there is a motion m such that $P(m, Q) = 0$, $P(T, Q) = 0$ thus there is no need to do any probability computation.

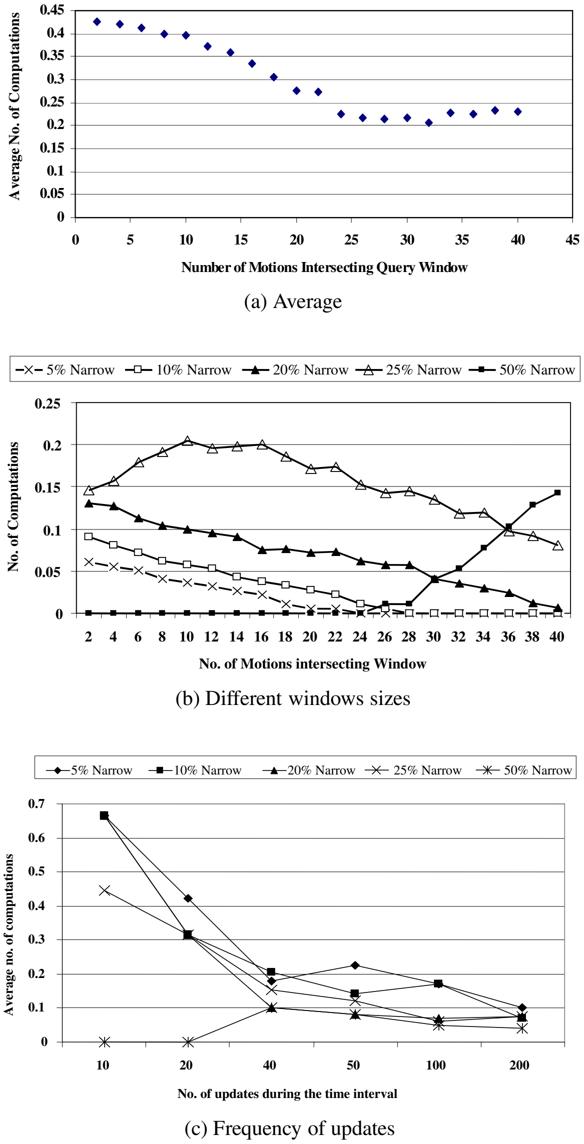
Therefore the idea of the algorithm is to perform containment and disjointness checking for each motion with respect to the query window and calculate probabilities only when it is necessary.

The algorithm is quite simple and omitted here. Below we present experimental results which counts the number of times the probability computations are done in a UR query. The results confirm our intuition.

The experiments are conducted for 1-dimensional case. We generated 1-dimensional trajectories consisting of 200 motions with the following characteristics: motion lengths are uniformly distributed between 2 and 5, moving velocities are uniformly distributed between 30 and 120. Given a start time, a position, a velocity, and an update time of a motion, we computed the next update position. The trajectories generated are similar to that in Figure 3, where the locations at update times are precise and the uncertainty is proportional to the distance to the closest update position. For uncertainty range δ, we generated deviation at the middle point between two consecutive updates, for fast objects δ is large and for slow objects δ is small.

The query workload consists of windows that have different spatial position and that have various widths. We considered 20 sets of windows whose temporal extents span over $2i$ $(1 \leqslant i \leqslant 20)$ motions. Spatial extents of query windows cover $5\%, 10\%, 20\%, 25\%, 50\%$ of the entire space. This gave a query workload of 100 different universal range queries.

Fig. 7(a) shows the average number of times motion probability computations need to compute. The average is over queries with different temporal extents. It is interesting to observe that the number of computation actually decrease, in contrast to the worst case linear complexity in Theorem 5.1. The main reason is that as more motions are covered, it is more likely that some motion probability become 0 or 1.



(a) Average

(b) Different windows sizes

(c) Frequency of updates

Figure 7. Number of probability computations

Fig. 7(b) gives a closer look into the previous result. It shows the average for each temporal extent covering $2i$ motions for $1 \leqslant i \leqslant 20$. For windows of small spatial extents, more probability computations are needed on the average initially, and this number decreases as the temporal extent enlarges. The reason for this phenomenon is that as more motions are considered, the chances of some motion being disjoint with the query window increase. Windows with larger spatial extents tend to have more probability computations initially since they are more likely to intersect motions.

However, when the spatial region of the window is too large (50% of the space), it is likely that the uncertainty region of motions will be completely contained in the window and thus the number of probability computation becomes very small. This difference also exists for the temporal extent analysis. As more motions are covered, the number of computations rises instead of decreases. It is expected that after some point the number should go down, resembling the small window cases.

Fig. 7(c) shows the effect of the update frequency on the number of computations needed by the technique. It considers different window spatial extents (5%,10%,20%,25%,50%) and shows the effect of varying the update frequency for the different spatial cases. For windows with small spatial extents, more probability computations are needed on the average when the frequency of update is very low. The reason for this phenomenon is that fewer motions are considered and due to narrow windows, the likelihood of intersection increases. Windows with larger spatial extends tend to have fewer average number of needed computation, the reason is that more uncertainty regions will be likely to be completely contained in the query window. It could be easily observed as well that as the frequency of updates increases, the number of computations decreases and this result follows the result in Fig. 7(a) since more updates indeed means more number of motions.

The experimental results are encouraging as they give some realistic indication of the complexity of the problem, while Theorem 5.1 merely gives the worst case bound.

6. Conclusions and Future Work

In this paper we discussed moving objects with imprecise trajectory information. We borrowed techniques from probability theory and proposed a data model for uncertain trajectories. The model proposed views a trajectory as a uniform stochastic process. Using this model, we presented a general technique for the evaluation of a type of spatio-temporal queries named universal range (UR) queries. In those queries the answer set is basically a set of objects and probability values that specify the probability of the object satisfying the UR query. In addition, we presented experimental results that show that in reality complexity results are much cheaper than expected theoretical analysis.

This paper presents a start for more possible work in this area that handles uncertain trajectories. For example, it is unclear how to evaluate UR queries where the spatial regions are not rectangles. It is also interesting how to index trajectories with uncertainty and evaluate UR and other queries more efficiently.

References

[1] Allied Business Intelligence Inc. *Location Based Services: A Strategic Analysis of Wireless Technologies, Markets and Trends*, January 2001. http://www.alliedworld.com/abiprdisplay2.jsp?pressid=7.

[2] P. Agarwal, L. Arge, and J. Erickson. Indexing moving points. In *19th ACM Symposium on Principles of Database Systems*, pages 175–186, 2000.

[3] R. Benetis, C. S. Jensen, G. Karciauskas, and S. Saltenis. Nearest neighbor and reverse nearest neighbor queries for moving objects. In *Sixth International Database Engineering and Applications Symposium*, Edmonton, Alberta, Canada, July 2002.

[4] A. Bhattacharya and S. Das. LeZi-Update: An information-theoretic framework for personal mobility tracking in PCS networks. *Wireless Networks*, 8:121–135, 2002.

[5] R. Cheng, D. Kalashnikov, and S. Prabhakar. Evaluating probabilistic queries over imprecise data. In *ACM International Conference on Management of Data SIGMOD*, 2003.

[6] R. Cheng, S. Prabhakar, and D. V. Kalashnikov. Querying imprecise data in moving object environments. In *Proc. Int. Conf. on Data Engineering*, Bangalore, India, Mar. 2003.

[7] M. Erwig, R. H. Guting, M. Schneider, and M. Vazirgiannis. Spatio-temporal data types: an approach to modeling and querying moving objects in databases. *GeoInformatica*, 3(3):269–296, 1999.

[8] L. Forlizzi, R. H. Guting, E. Nardelli, and M. Schneider. A data model and data structures for moving objects databases. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 2000.

[9] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Varirgiannis. A foundation for representing and querying moving objects. *ACM Trans. Database Syst.*, 25(1), 2000.

[10] G. Kollios, D. Gunopulos, and V. J. Tsotras. Nearest neighbor queries in a mobile environment. In *Spatio-Temporal Database Management*, pages 119–134, 1999.

[11] G. Kollios, D. Gunopulos, and V. J. Tsotras. On indexing mobile objects. In *Proc. ACM Symp. on Principles of Database Systems*, pages 261–272, 1999.

[12] P. Leroux. Creativity, reliability to drive telematics. *ZDnet*, August 2002.

[13] H. Mokhtar, J. Su, and O. Ibarra. On moving object queries. In *Proc. ACM Symp. on Principles of Database Systems*, 2002.

[14] C. Orski. The state of u.s. telematics. *Newsletter of the ITS Cooperative Deployment Network*, January 2002.

[15] D. Pfoser and C. Jensen. Capturing the uncertainty of moving-object representations. In R. H. Güting, D. Papadias, and F. H. Lochovsky, editors, *Advances in Spatial Databases, 6th International Symposium, SSD'99, Hong Kong, China, July 20-23, 1999, Proceedings*, volume 1651 of *LNCS*, pages 111–132. Springer, 1999.

[16] D. Pfoser, C. Jensen, and Y. Theodoridis. Novel approaches to the indexing of moving object trajectories. In *Proc. Int. Conf. on Very Large Data Bases*, 2000.

[17] D. Pfoser and N. Tryfona. Capturing fuzziness and uncertainty of spatiotemporal objects. In *ADBIS*, pages 112–126, 2001.

[18] T. Roos, P. Myllymaki, and H. Tirri. A statistical modeling approach to location estimation. *IEEE Trans. on Mobile Computing*, 1(1):59–69, January-March 2002.

[19] S. Saltenis, C. Jensen, S. Leutenegger, and M. Lopez. Indexing the positions of continuously moving objects. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 2000.

[20] A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and querying moving objects. In *Proc. Int. Conf. on Data Engineering*, 1997.

[21] Z. Song and N. Roussopoulos. k-nearest neighbor search for moving query point. In *Proc. Int. Sym. on Spatial and Tempporal Databases*, pages 79–96, 2001.

[22] H. Stark and J. W. Woods. *Probability and Random Processes with applications to signal processing*. Prentice Hall, third edition, 2002.

[23] J. Su, H. Xu, and O. Ibarra. Moving objects: Logical relationships and queries. In *Proc. Int. Sym. on Spatial and Tempporal Databases*, pages 3–19, 2001.

[24] Y. Tao and D. Papadias. MV3R-tree: A spatio-temporal access method for timestamp and interval queries. In *Proc. Int. Conf. on Very Large Data Bases*, 2001.

[25] G. Trajcevski, O. Wolfson, F. Zhang, and S. Chamberlain. The geometry of uncertainty in moving objects databases. In *Advances in Database Technology - EDBT 2002, 8th Int. Conf. on Extending Database Technology*, pages 233–250. Springer, March 2002.

[26] M. Vazirgiannis and O. Wofson. A spatiotemporal model and language for moving objects on road networks. In *Proc. Int. Sym. on Spatial and Tempporal Databases*, 2001.

[27] O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang. Moving objects databases: issues and solutions. In *Proc. Int. Conf. on Statistical and Scientific Database Management*, pages 111–122, 1998.

[28] http://www.webtechwireless.com/industryinfo.htm.

[29] J. Zhang and L. Gruenwald. Spatial and temporal aware, trajectory mobility profile based location management for mobile computing. In *Proc. MDDS*, 2002.