# Processing Probabilistic Spatio-Temporal Range Queries over Moving Objects with Uncertainty

Bruce S.E. Chung
Department of Computer Science
National Tsing Hua University
Hsinchu, Taiwan, R.O.C.
se.chung@yahoo.com.tw

Wang-Chien Lee
Dept of Comp. Science & Engineering
Pennsylvania State University
State College, PA 16801, U.S.A.
wlee@cse.psu.edu

Arbee L.P. Chen
Department of Computer Science
National Chengchi University
Taipei, Taiwan, R.O.C.
alpchen@cs.nccu.edu.tw

## ABSTRACT

Range queries for querying the current and future positions of the moving objects have received growing interests in the research community. Existing methods, however, assume that an object only moves along an anticipated path. In this paper, we study the problem of answering *probabilistic range queries* on moving objects based on an uncertainty model, which captures the possible movements of objects with probabilities. Evaluation of probabilistic queries is challenging due to large objects volume and costly computation. We map the uncertain movements of all objects to a dual space for indexing. By querying the index, we quickly eliminate unqualified objects and employ an approximate approach to examine the remaining candidates for final answer. We conduct a comprehensive performance study, which shows our proposal significantly reduces the number of object examinations and the overall cost of the query evaluation.

## 1. INTRODUCTION

There are growing demands for moving objects monitoring functions in numerous mobile applications, such as traffic monitoring, fleet management, flight control, etc. For instance, by continuously receiving location updates from buses on roads, a bus control system can perform a better bus scheduling. To avoid traffic congestion, a range query [17], which “retrieves all the buses that will arrive within 1 mile of the station in the next 10 minutes”, may be issued to obtain estimated answers based on buses’ velocities or locations stored in the database.

Research on spatio-temporal databases that manage objects’ moving information has produced fruitful results in indexing and querying techniques. A number of efficient methods for managing moving objects have been proposed. Typically, they assume that an object moves on an anticipated path. As a result, an object’s most possible path based on a linear function [11,16,19] or recent information [18] are maintained. In fact, the real location of an object is known for certain to the server only when an update is received. The *uncertainty* of the object’s location increases as time grows until the next update is received. Thus, the uncertainty has a great impact on the accuracy of these proposed methods.

The scenarios of objects moving on routes have been widely considered in research studies appeared in the literature due to various moving object database applications (e.g., trucks in fleet management and buses in bus scheduling). Noticing that a route can be considered as a sequence of smaller line paths, we adopt *line-segment uncertainty* in this paper. Accordingly, for objects which moves along straight line paths, the uncertainty at any time is given by a line-segment [2].

Since most objects move without deviating from their recent moving behaviors drastically, the uncertainty can be bounded, i.e., an object’s future positions could be captured probabilistically using an *uncertainty probability density function* (pdf) [2]. A pdf of an object $O_i$ moving on a line-segment (i.e., in a one-dimensional (1-D) space) can thus be represented as $f_i(x,t)$ where $f_i$ is the probability of $O_i$ at the location $x$ at the time instant $t$. Therefore, an object’s location at any time can be estimated with an uncertainty probability defined by its pdf. Typically, the function $f_i$ can be derived from $O_i$’s past moving behavior or the moving velocity distribution. Thus, each object can be stored in the server database with a pdf instead of a location from the last update.

Typically, a moving object’s velocity distribution can be approximated as a normal distribution. In this paper, we employ the *Brownian motion with drift process* (called *Brownian motion* in short) [4,10,13] as an uncertainty model to produce a pdf from a normal velocity distribution for all objects. The Brownian motion model can be derived easily with only two parameters, i.e. mean velocity and variance. Thus, by adopting this model, computation cost at the server and communication cost between the server and the objects can be reduced.

EDBT'09, March 24-26, 2009, Saint Petersburg, Russia.

To query moving objects using an uncertainty model, [2] has proposed *probabilistic range queries (PRQ),* e.g., “retrieve the identifications of all the buses that will come within 1 mile of the station in the next 10 minutes with a probability more than 0.4.” Unlike a conventional range query, a probability is specified as part of query conditions. Additionally the answer is to be obtained over a *time period* instead of just a *time instant*. An object is returned as an answer if its probability inside the query range (1 mile) is greater than 0.4 at any moment during the next 10 minutes. This query, by taking into account both the location range over the time period, returns objects with probabilities satisfying the specified probability threshold.

Probabilistic condition in the query can only be evaluated on objects with expensive integral computation, incurring significant overhead. Several methods have been proposed to index objects with uncertain movement [3,20]. A common idea is to pre-determine the uncertainty intervals of objects with the same probability bounds. Several different bounds are pre-defined with various probability values and grouped by the proposed indexes. A query is processed by first verifying these bounds to reduce the number of objects under consideration. However, these indexes are not applicable to *time-varying uncertain data*. In other words, the assumed uncertainty model is time-independent, i.e., an object’s uncertain range and probability values remain unchanged no matter how much time passing by from the last update.

In this paper, we propose efficient techniques for querying uncertain time-varying data. We observe that the expensive query cost is due to: (1) the waste of time in evaluating objects far away from the query range, and (2) the expensive integral computation required for evaluating the probability. In order to solve problem (1), we transform the uncertain movements of objects into points in a dual space using the Hough Transform [8]; therefore, these points can be indexed by an arbitrary point access method [6]. After transforming the movements, however, only the most representative path with a timestamp is maintained in the index. As a result, some of the answers may get lost when querying on the index. We expand the query to avoid losing answers and transform the expanded query into a search range. Querying on the index using the search range allows pruning the unqualified points outside the search range. Thus, the cost of object examinations is reduced. After the elimination process, the remaining objects are examined to evaluate their probabilities. For problem (2), we present an approximate approach to reduce the cost of integral operations in evaluating the probabilistic query condition. While the proposed approximate approach results in false positives, none of the objects that should be included in the answers is lost. We also develop an error function to compute the probability bound for the answers.

A performance study is conducted to compare the running time of the probabilistic range queries using our strategy with a non-indexing method. We show the effectiveness of elimination process with different number of objects and the accuracy of the examination with different query parameters.

The rest of this paper is organized as follows. In Section 2, we discuss related works. In Section 3 we define the problem, discuss the Brownian motion model, and review the Hough Transform technique. In Section 4, we describe our indexing method and query processing algorithms for PRQ. In Section 5, we evaluate the proposed method. Finally, we conclude the paper in Section 6.

## 2. RELATED WORK

Here we review the existing techniques for range queries on moving objects and the probabilistic queries over uncertain data.

### 2.1 Range Queries over Moving Objects

Dual transformation based on Hough Transform has been employed for querying spatial-temporal data. Kollios et al. [11] map predicted trajectories of moving objects into points in a two-dimensional space by dual transformation and reduce the problem of querying the objects’ locations to an issue of point access.

Saltenis et al. [16] proposed time-parameterized R-tree (TPR-tree) to index moving objects. A minimum bounding box (MBR) and a velocity bounding vector (VBV) were used to cover the range where the covered objects will reach. Therefore, the extent of a node in the TPR-tree grows with time. While an update is performed as in the R*-tree, query processing needs to compare the query time with the update time in the MBRs. An improved index called TPR*-tree was proposed by Tao et al. [19]. TPR*-tree creates tighter extends and develops better deletion and tree-splitting operations to improve the performance of the TPR-tree.

### 2.2 Query Processing on Uncertain Data

Cheng et al. [1] classify different types of probabilistic queries over imprecise data in sensor environments. Accordingly, several processing algorithms with probabilistic estimates were presented. The issue of quality measure for the answers was addressed by assuming that the uncertainty pdfs of moving objects are fixed and independent of time, which is unrealistic. Cheng et al. [2] also propose probabilistic queries over moving objects. Unlike the model in [1], the uncertainty model is dependent of time. The uncertainty range and pdf of a moving object is based on when the object updates and the query processes. The authors presented algorithms for range and nearest-neighbor query respectively.

Since both methods proposed by Cheng need to examine the uncertainty information of each object at least once for a query which incurs expensive integral operation, Cheng et al. [3] further develop several solutions to deal with this problem. The authors propose *probability threshold indexing* (PTI) to store the pre-computed probability bounds called *x-bounds*, which the uncertainty intervals in a PTI have the same probability bounds. Comparing the probability threshold in the query with the x-bounds, the cost of the query processing can be reduced. Tao et al. [20] followed this idea to propose *conservative functional box*, represented as a linear function, to bind the probabilities of objects for multi-dimensional uncertain data and arbitrary pdfs. *U-tree* is developed to index the conservative functional boxes.

Cheng and Tao develop efficient indexes to reduce the cost of processing probabilistic queries. Nevertheless, these techniques costs a lot because each object needs to be evaluated at least once and indexed based on pre-computed bounds, which do not always satisfy the arbitrary probability threshold of queries. Additionally, these works do not take into account the uncertainty changing with time. The uncertainty ranges and probabilities in these works are assumed to be constant instead of basing on time-varying pdfs.

## 3. PRELIMINARIES

In this section, we first describe an uncertainty model for moving objects and then formally define probabilistic range queries. Finally, we describe Hough Transform.

### 3.1 Uncertainty Model

Given a set of moving objects *Θ,* where the *i*th object in *Θ* is denoted as $O_i$, we assume that $O_i$ is moving in one dimension (1-D), e.g., *x*-axis, and is able to compute its own location *x* at any time *t*. Location updates are submitted to a server based on some update policies, which is out of scope of this paper since they do not affect the uncertainty model. Upon receiving an update from object $O_i$, the server calculates the uncertainty with respect to $O_i$'s location and timestamp. Based on [2], the uncertainty function can be defined in 1-D space as follows:

**Definition 1: Uncertainty Probability Density Function (pdf).** *The pdf of an object $O_i$, denoted by $f_i(x, t)$, is a probability value of $O_i$'s location x at time t.* ■

After receiving the update from $O_i$, the server calculates $f_i$ to represent the after-update motion of $O_i$. The server re-calculates $f_i$ when the next update is received. It has the property that $f_i(x_0, t_0) = 1$ if an update information $(x_0, t_0)$ is issued by any object $O_i$.

Since the past moving behavior (i.e., velocity distribution) of an object usually has an impact on its future motion in reality, it is feasible to derive $O_i$'s pdf $f_i$ from its past velocity distribution. However, the derived functions may incur excessive overhead due to extra cost for exchanging probabilistic parameters and performing complex integral operations. To reduce the overhead, we choose normal distribution to express velocity distributions of moving objects. Even though not all moving objects' velocity distributions fit well, most objects can be approximated by a normal distribution. Since the uncertainty of an object's location increases as the time goes, a good candidate model of uncertainty is *Brownian motion with drift process* [4,10,13] (called Brownian motion for simplicity in this paper). Lei et al. [12] and Rose [15] present methods to minimize the paging cost both based on the model Brownian motion. The one-dimensional Brownian motion is represented and formulated as follows:

**Definition 2: Brownian motion with drift process.** *The pdf of an object $O_i$, starting at location $x_0$ at time $t_0$, model by Brownian motion is denoted by*

$$f_i(x,t) = \frac{1}{\sqrt{2\pi D(t-t_0)}} \times \exp\{\frac{-(x-x_0-v(t-t_0))^2}{2D(t-t_0)}\}, t \ge t_0$$

*The mean function of $f_i$ is denoted by*

$$E_i[x(t)] = \mu_i(t) = x_0 + v_i(t-t_0) \cdot$$

*The variance function of $f_i$ is denoted by*

$$Var_i[x(t)] = \sigma_i^2(t) = D_i(t-t_0) \cdot$$ ■

The above function is based on a normal velocity distribution. The mean and the variance of the process are linear functions of the time interval $(t - t_0)$. $D_i$ is the diffusion parameter, also called the variance, with units of (*length*$^2$*/time*) and *v* is the average velocity (*length/time*), or named the mean, of $O_i$'s velocity distribution [15]. The distribution of Brownian motion is a Gaussian pdf at any time and is parameterized by the time interval $(t - t_0)$. The uncertainty probabilistic density function of an object can be evaluated by two parameters mean and variance.

**Example 1**. As shown in Figure 1, an object $O_i$ updates at the location 0 at time 0 with the mean $v_1$ = 10 *meter/min* and the variance $D_1$ = 4 *meter*$^2$*/min* for its velocity distribution. The time instants $t_1$, $t_2$, and $t_3$ refer to the moment after 1 min, 2 min and 5 min from the update. Note that $f_1(x, t_1)$ is a normal distribution, and so are $f_1(x, t_2)$ and $f_1(x, t_3)$. In other words, at any time *t* except when the update is received, $f_1(x, t)$ is a normal distribution. The mean and the variance functions are linear to time such as $\mu_1(t_1) = v_1 \times t_1 = 10$ and $\sigma_1^2(t_1) = D_1 \times t_1 = 4$. ■

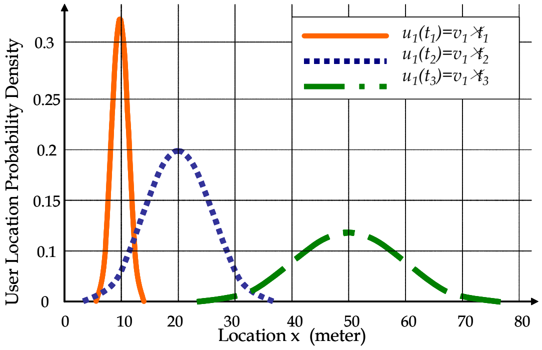


**Figure 1. An Example of Brownian motion.**

## 3.2 Probabilistic Range Query Definition

Based on the above model, the probability of an object inside a given location range can be obtained. Users are usually interested in objects in a query range with a certain probability or above. Therefore, a *probabilistic range query* (PRQ) [2] is defined by augmenting range queries with a *probability threshold*. An object whose probability over the specified query range satisfies the probability threshold will be returned as part of the query results. We modify and re-define PRQ from [2] in 1-D space as follows:

**Definition 3: Probabilistic Range Queries (PRQ).** *Given* (1) *a closed location interval* $[x_1, x_2]$ $(x_1, x_2 \in x-axis$ *and* $x_1 \le x_2)$, (2) *a time period* $[t_1, t_2]$ $(t_{now} \le t_1 \le t_2)$, *and* (3) *a probability threshold p. A PRQ returns the object* $O_i$ *if* $\exists p_{ij} \ge p$ *at any time instant* $t_j$ $(t_1 \le t_j \le t_2)$ *where*

$$p_{ij} = \int_{x_1}^{x_2} f_i(x, t_j)dx \qquad \blacksquare$$

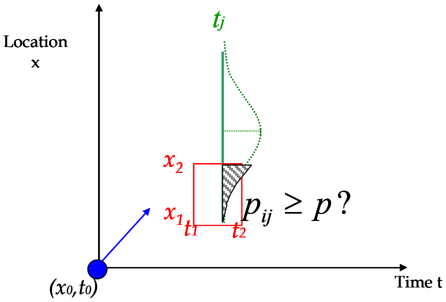


**Figure 2. An example of PRQ.**

The query $q$ in Figure 2 is a probabilistic range query. An object $O_i$ updated its information at $(x_0, t_0)$. Its uncertainty pdf is modeled by Brownian motion (the arrow indicates the mean velocity of the object's distribution). At any time $t_j$ $(t_1 \leqq t_j \leqq t_2)$, if the cumulated probability value $p_{ij}$ corresponding to where $O_i$ is located inside the interval $[x_1, x_2]$ (i.e., the grey area in the figure) is greater than or equal to the probability threshold $p$, then the object $O_i$ will be in the answers for the query $q$. On the contrary, if none of $p_{ij}$ satisfies the threshold, $O_i$ will not be returned by this query. This definition of PRQ is different from [2] as the time instant of a PRQ is generalized to a *time period*.

## 3.3 Hough Transform

Hough Transform [8] is widely used in pattern recognition. A number of indexing structures have been proposed based on this technique. Jagadish [9] proposes to use Hough Transform to index line segments, and Kollios et al. [11] transformed moving objects' trajectories into points for indexing. Hough Transform maps a hyperplane $h$ from $R^d$ to a point in $R^d$. We present the concept in a two-dimensional case. Three basic properties of Hough Transform between the primal 2-D space and the 2-D dual space (called by [11]) are as follow.

➢ *A line S: y = mx + b in the primal space can be mapped to a point S': (m, b) in the dual space, where m is the slope of S and b is the intercept of y-axis in the primal space.*

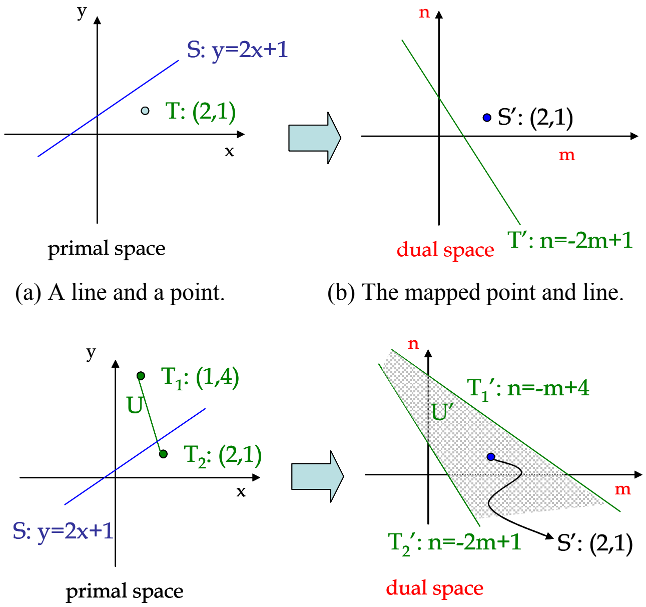
➢ *A point T:* $(x, y)$ *in the primal space can be mapped to a line T': n = - xm + y in the dual space, where x is the slope of T' and y is the intercept of n-axis in the dual space.*

➢ *A line segment U defined by the two points* $T_1$: $(x_1, y_1)$ *and* $T_2$: $(x_2, y_2)$ *in the primal space can be mapped to an area U' in the dual space, and U' is bounded in the two lines* $T_1$*':* $n = - x_1m + y$ *and* $T_2$*':* $n = - x_2m + y$*. If a line S intersects the line segment U in the primal space, then the mapped point S' form S is definitely included by U'.*

We take an example to illustrate these properties.

**Example 2.** Given a line $S$: $y = 2x + 1$ and a point $T$: (2, 1) in the primal space as shown in Figure 3(a), we map the line $S$ to a point $S'$: (2, 1) and the point $T$ to a line $T'$: $n = -2m + 1$ in the dual space (see Figure 3(b)) based on the first two properties. If the line $S$: $y = 2x + 1$ intersects the segment $U$, defined by two points $T_1$: (1, 4) and $T_2$: (2, 1) in the primal space (see Figure 3(c)), the mapped point S' in the dual space is included in the area $U'$ where $U'$ is mapped from $U$ (see Figure 3(d)). ■

The transform with the above properties is called *Hough-X transform*, which does not treat a vertical line since its slope is infinite. In contrary, Hough-Y [9] transform treats vertical lines but not horizons. Our method is based on Hough-X transform since no vertical lines appear in our method.



(a) A line and a point.

(b) The mapped point and line.

(c) *S* intersects the line segment *U*.

(d) The area *U'* includes the point *S'*.

**Figure 3. Hough Transform.**

## 4. INDEXING AND QUERY PROCESSING

A naïve method to process the probabilistic range query is to retrieve all objects and to evaluate their pdfs for the probability values. The objects whose probabilities satisfy the probability threshold of the query are returned. Two problems arise from this approach. First, as the number of moving objects is very large, it is inefficient to retrieve all objects in query processing. Second, as shown in Definition 3, the calculation of each object's pdf involves an expensive integral operation at a time instance. This is particularly impractical as the query is defined with a time period.

Since the movements of an object are uncertain, it is infeasible to maintain all possible future positions of any object. One important characteristic of Brownian motion is that the average velocity during the closest two updates is the same, i.e. an object's expected movement can be represented with a linear function of time. Using Hough Transform, we transform an object's expected movement, i.e., a line, into a point in a dual space. By mapping the moving objects into a dual space, points in the dual space become much easier to be indexed. Meanwhile, a query can be transformed into a search range in the dual space. Therefore, we can eliminate unqualified objects for the query via the index. Moreover, we develop an approximate approach to examine the remaining objects to prevent the costly integrals. Furthermore, we can assure the returned answers to be within bounded errors.

### 4.1 Uncertain Movements Indexing

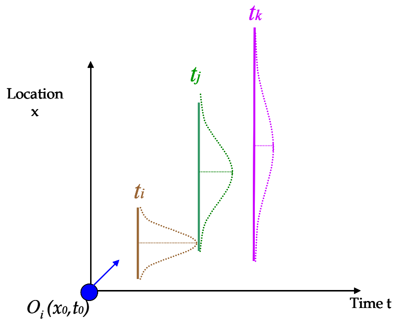
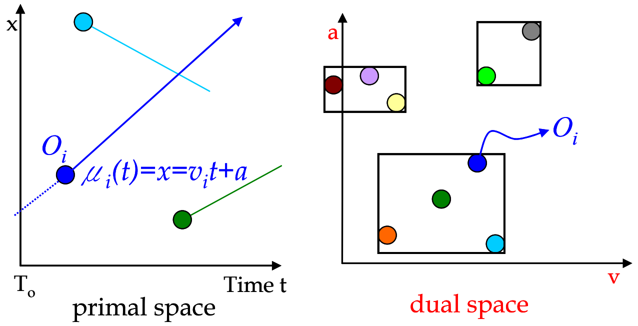


**Figure 4. The uncertain movements of $O_i$.**

The location uncertainty of an object increases from its last update (as shown in Figure 4). The probability of an object $O_i$'s location varies at different time instant, i.e., the probability value depends on the $O_i$'s location $x$ at the instant time $t$. Therefore, three attributes, *location*, *time*, and *probability,* are to be considered when we index uncertain movements of objects. Based on [3], we can intuitively derive a curve that bounds objects' locations with the same probability. However, it's difficult to formulate the curve as a function. Further, to answer query on these curves, numerous curves need to be maintained in the index.



(a) The mean functions of objects' pdfs.

(b) The points are indexed by the R-tree.

**Figure 5. Uncertain movements transform and indexing.**

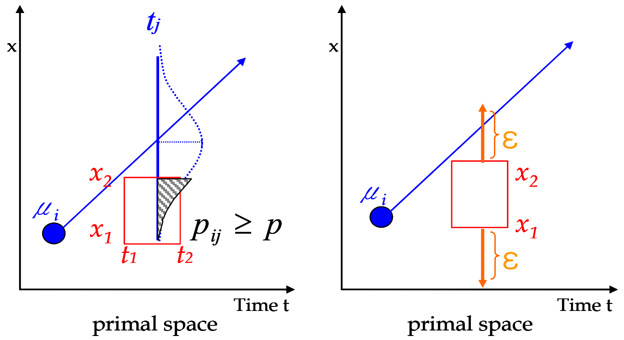
Since the mean function of Brownian motion is linear (see Definition 2), the curve of $f_i$, which is a normal distribution function, is symmetrical with respect to $E_i$. Hence, we index the mean functions of the moving objects. These linear functions are illustrated in Figure 5(a). The equation of each line is $\mu_i(t) = x = v_it + a$ in the plane $(v, t)$ where $v_i$ is the slope (the mean velocity in this case) of $O_i$ and $a$ is the intercept of $x$-axis. Note that $t$ is corresponding to the pre-defined time origin $T_o$.

With the first property of Hough Transform, we can map any mean function $\mu_i(t) = x = v_it + a$ as a line to a point ($v_i$, $a$) as shown in Figure 5(b). In the dual space, the horizontal and vertical axes indicate the velocity and the intercept, respectively. A vertical line is never mapped because no object moves in a velocity of infinity. In the dual space, these points are easy to index by a number of point access methods [6]. Kollios et al. [11] has conducted a complete study on the quality of various indexing structures. Since our work does not focus on indexing structures, we choose to use R-tree [5] for its popularity.

## 4.2 Query Transformation

Our indexing scheme aims at obtaining the mean functions of the objects' pdfs efficiently. The original problem of querying uncertain movements becomes an issue of querying which lines cross the range of the query. Thus, the range query should be transformed to another query that supports querying of lines. Unfortunately, some answers are lost when the results of querying lines are taken as the answers of querying uncertain movements.

### *4.2.1 Query Expansion*



(a) $p_{ij} = 0.4$ is larger than the threshold $p = 0.3$.

(b) The query is expanded by the parameter ε.

**Figure 6. The solution for the lost answers.**

Figure 6(a) illustrates an example where an object $O_i$ is missed if objects (represented as lines) are queried by specified by a range in time-location space with a probability threshold $p = 0.3$. Since the line of $\mu_i$ does not cross the query range, $O_i$ is not qualified. However, $O_i$ is indeed an answer since there exists a probability value $p_{ij}$ = 0.4 (> $p$). This issue can be addressed by expanding the query range by ε (as shown in Figure 6(b)). Lemma 1estimates ε based on different query conditions.

**Lemma 1**: *A probabilistic range query q is specified by a location interval* $[x_1, x_2]$ ( $x_1 \le x_2$ ), *a time period* $[t_1, t_2]$ ( $t_1 \le t_2$ ), *and a probability threshold p (0<p<1). If an object* $O_i$ *satisfies the query conditions of q, then the mean function* $\mu_i$ *of* $O_i$*'s pdf crosses the range defined by location interval* $[x_1 - \varepsilon, x_2 + \varepsilon]$ *and time period* $[t_1, t_2]$ *where*

(1) *if p≧0.5, then ε=0;*

(2) *if p<0.5, then* $\varepsilon < (\frac{0.5 - p}{p})(x_2 - x_1)$. ■

**Proof:** Let $p_{ij}$ be the evaluated probability of $O_i$'s location being inside the interval $[x_1, x_2]$ at any time instant $t_j$ where $t_1 \le t_j \le t_2$.

(1) If $p \geq 0.5$. Since the locations in Brownian motion at any time is a normal distribution, the curve of $f_i$ is symmetrical with respect to mean function $\mu_i$ at any time instant. Thus, the cumulated probability from any end to the center where $\mu_i$ is equal to 0.5. If $\mu_i$ does not cross $q$, $q$ must be in one of the two sides separated by the mean function $\mu_i$. The cumulated probabilities over any finite location intervals, not across $\mu_i$, are never larger than or equal to 0.5. If the mean function $\mu_i$ does not cross $q$, its probability $p_{ij}$ cannot satisfy the probability threshold $p$. Therefore, we do not expand the query when $p$ is larger than or equal to 0.5.

(2) If $p < 0.5$. We expand the query only when $p$ is less than 0.5. As mentioned in (1), the cumulated probability is not larger than 0.5 in one side of the curve of the normal pdfs. Consider that an object's $p_{ij}$ satisfies $p$ but its $\mu_i$ does not cross $q$, then $p_{ij}$ is greater than or equal to $p$ as shown in Figure 6(a). The cumulated probability over the interval between $\mu_i(t_j)$ and the nearest end of $[x_1, x_2]$ to $\mu_i(t_j)$ (it is $x_2$ in this case) is less than (0.5 – $p$). In normal distribution, it is obvious that the probability at the location near the center (i.e. mean) is definitely greater than that on the side. If a pdf cumulated over two different intervals, which have the same length without crossing the center, then the cumulated probability over the interval near the center is greater than the one on the side. In other words, with the same cumulated probability value, the length of the interval near the center is less than the interval on the side. (0.5 – $p$)/$p$ is the multiple of $p$, and $p$ is cumulated over the length ($x_2$ – $x_1$). Therefore, since it is nearer than the interval $[x_1, x_2]$ to $\mu_i(t_j)$ the length ε of the interval between $\mu_i(t_j)$ and $x_2$ is less than $(\frac{0.5 - p}{p})(x_2 - x_1)$. ■

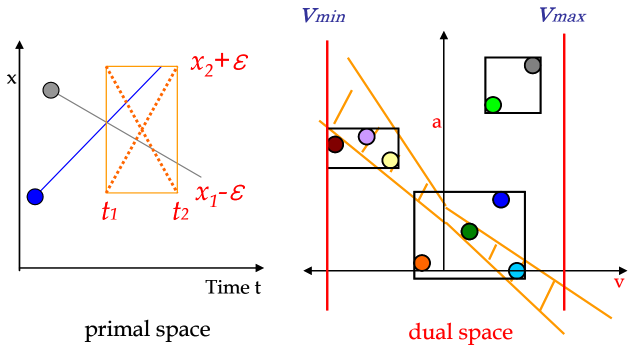
Therefore, we expand the query by ε to obtain all the objects which should be in the answers. Lemma 1 also confirms that if a line $\mu_i$ does not cross the expanded query range, there is not a probability $p_{ij}$ satisfying the query probability threshold $p$.

### *4.2.2 Query Representation*

After indexing the uncertain movements of the objects and expanding the query range of a PRQ, we reduce the problem of querying uncertain movements into an issue of determining which lines cross the query range. These lines, representing the most likely moving paths, are transformed into the points and indexed in the dual space. Consider that a range query specified by the two intervals [$x_1$ - ε, $x_2$ + ε] and [$t_1$, $t_2$] with probability threshold $p$ as shown in Figure

7(a). If an infinite line crosses a rectangle, then this line intersects at least one of the diagonal lines of the rectangle. Thus, the issue of determining lines crossing the range can be decomposed into two problems: 1) which lines with positive slopes intersect the line segment defined by the two points ($t_1$, $x_2$ + $\varepsilon$) and ($t_2$, $x_1$ - $\varepsilon$); and 2) which lines with negative slopes intersect the line segment defined by the two points ($t_1$, $x_1$ - $\varepsilon$) and ($t_2$, $x_2$ + $\varepsilon$). Since the slopes of the all lines are the mean velocities of moving objects, they are constrained by various conditions, e.g., speed limit of the cars and roads. We can define the $V_{max}$ and $V_{min}$ as the velocity bounds of objects. Let $v_i$ be the mean velocity of an object $O_i$'s velocity distribution. A query is transformed into a search range (see Figure 7(b)), expressed using a linear constraint query [7] as:

- *If* $v \geq 0$, *then* $Q = C_1 \wedge C_2 \wedge C_3 \wedge C_4$, *where* $C_1 = a + vt_2 \geq (x_1 - \varepsilon)$, $C_2 = a + vt_1 \leq x_2 + \varepsilon$, $C_3 = v \leq V_{max}$ *and* $C_4 = v \geq V_{min}$.
- *If* $v < 0$, *then* $Q = D_1 \wedge D_2 \wedge D_3 \wedge D_4$, *where:* $A_1 = a + vt_1 \geq (x_1 - \varepsilon)$, $C_2 = a + vt_2 \leq x_2 + \varepsilon$, $C_3 = v \leq V_{max}$ *and* $C_4 = v \geq V_{min}$.



(a) The expanded query in the primal space.

(b) The searching range mapped from the query in (a).

**Figure 7. Query transformation.**

The linear constraint query is transformed by the two diagonal line segments using the third property of Hough Transform discussed in section 3.3. The search range is transformed by the expanded query in the primal space to query which points are inside the range. The process of querying the points on the index can be efficiently performed by eliminating the points outside the search range since all points are indexed by the R-tree.

## 4.3 Examination Process

While the above strategy allows us to obtain candidate objects, not all of candidates satisfy the query conditions. To evaluate the probabilistic condition of queries, a costly integral operation, $p_{ij} = \int_{x_1}^{x_2} f_i(x,t_j)dx$, for each candidate needs to be evaluated. It can be evaluated at every time instant $t_j$ until there is a $p_{ij}$ satisfying the query threshold or $t_j$ is out of the query time period. However, this approach is inefficient due to excessive number of the integral computation for each object.

### 4.3.1 Examination Region

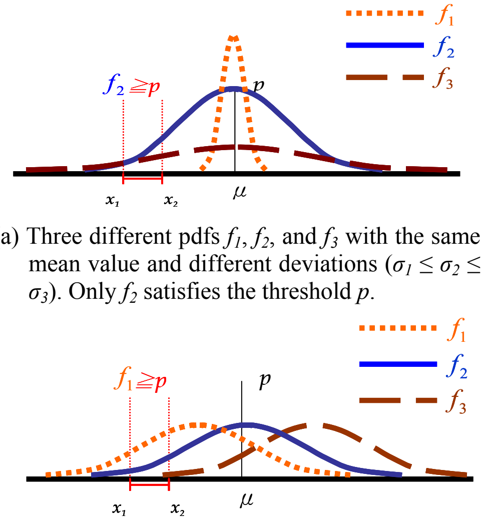
To address the inefficiency of objects examination, we propose a technique to reduce the overhead of probability evaluation for each object. If an object $O_i$ is in the answers, then:

$$\exists t_j \Rightarrow \int_{x_1}^{x_2} f_i(x,t_j)dx \geq p$$

Since $f_i$ is a normal distribution at time $t_j$, it can be written as:

$$\Phi(\frac{x_2 - u_i(t_j)}{\sigma_i(t_j)}) - \Phi(\frac{x_1 - u_i(t_j)}{\sigma_i(t_j)}) \geq p \qquad (1)$$

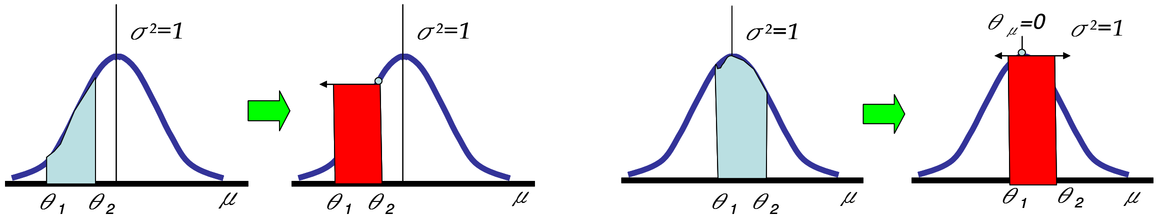
where $\mu_i(t_j)$ is the mean function and $\sigma_i(t_j)$ is the deviation function of $f_i(x,\ t_j)$ of $f_i$ at time $t_j$. Here, $\Phi(x)$ denotes the cumulative distribution function of a standard normal distribution. Figure 8(a) shows an example of what values of $\mu_i(t_j)$ and $\sigma_i(t_j)$ make equation (1) satisfy the probability threshold $p$. Three pdfs $f_1$, $f_2$, and $f_3$ have the same mean value and different deviations $\sigma_1$, $\sigma_2$, and $\sigma_3$, respectively, where $\sigma_1 \leq \sigma_2 \leq \sigma_3$. Only the cumulated probability over the interval [$x_1$, $x_2$] of $f_2$ satisfies the threshold $p$. Figure 8(b) shows another example. Three pdfs $f_1$, $f_2$, and $f_3$ have the same deviation value and different mean values $\mu_1$, $\mu_2$, and $\mu_3$ where $\mu_1 \leq \mu_2 \leq \mu_3$. Only $f_1$ with the nearest mean value to the interval [$x_1$, $x_2$] satisfy the threshold. Only some values of the mean with the corresponding deviations make Equation (1) satisfy the threshold.



(a) Three different pdfs $f_1$, $f_2$, and $f_3$ with the same mean value and different deviations ($\sigma_1 \leq \sigma_2 \leq \sigma_3$). Only $f_2$ satisfies the threshold $p$.

(b) Three different pdfs $f_1$, $f_2$, and $f_3$ with the same deviation value and different mean values ($\mu_1 \leq \mu_2 \leq \mu_3$). Only $f_1$ satisfies the threshold $p$.

**Figure 8. Evaluating the probability values of different pdfs.**



(a) The approximate integral where the interval [$\theta_1$, $\theta_2$] is not across the mean value.

(b) The approximate integral where the interval [$\theta_1$, $\theta_2$] is across the mean value.

**Figure 10. The approximate process of integral operations.**

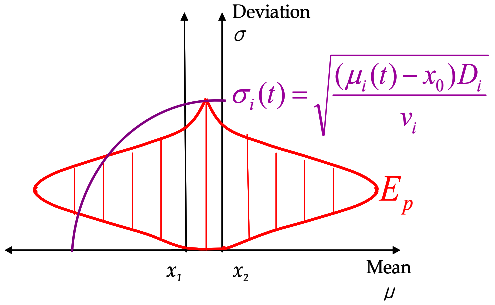


**Figure 9. The closed area $E_p$.**

From the above, we observe that the mean and deviation values are correlated. When the mean value of a pdf $f_i$ is fixed, only values within a deviation range can make $f_i$ satisfy the query threshold. Likewise, when the deviation value is fixed, a mean range is used for evaluating the probabilistic condition. Accordingly, Equation (1) can be evaluated as a closed area $E_p$ (where the suffix $p$ is the query probability threshold) in as shown in Figure 9. We define $E_p$ in Definition 4.

**Definition 4**: *Given a probability threshold p and an interval [$x_1$, $x_2$], a mean value $\mu$ and a deviation value $\sigma$ of a normal probability function f(x) become a point ($\mu$, $\sigma$) the mean-derivation space. An $E_p$ is a closed area in this space. Any probability value of an f(x) cumulated over the interval [$x_1$, $x_2$] is greater than or equal to p.*

The area $E_p$ is symmetrical with respect to ($x_1$ + $x_2$)/2 because of the symmetry of the normal distribution. Only when the values of $\mu$ and $\sigma$ are both inside the closed area, the corresponding cumulated probability value can satisfy the threshold. The point on the sideline of $E_p$ indicates that the cumulated probability value of the pdf defined by this point is equal to $p$. If the point is inside $E_p$, then the probability value is greater than $p$. From Definition 2, the mean and deviation functions can be combined as follows:

$$\sigma_i(t)=\sqrt{\frac{(\mu_i(t)-x_0)D_i}{v_i}} \tag{2}$$

Equation (2) becomes a curve in $\mu$–axis and $\sigma$–axis (as shown in Figure 9), and therefore, if it intersects $E_p$ during the time period [$t_1$, $t_2$], then the object will be one of the answers. Unfortunately, Equation (1) cannot be written as a closed form function since the equation involves normal distribution integral. Thus, in the next section, we provide an approximation to Equation (1).

### 4.3.2 *Approximate Examination*

First, when the mean value $\mu_i$ is not inside the interval [$x_1$, $x_2$] and $\mu_i$ is greater than $x_2$, the probability of $f_i$ at the location $x_2$ is greater than the probability at location $x_1$. When we transform the pdf $f_i$ into a normal distribution function $n(\theta)$, the location $x_2$ and $x_1$ can be transformed into the location $\theta_2$ and $\theta_1$ in $n(\theta)$. As shown in Figure 10(a), the probability of $n_i$ at the location $\theta_2$ is also greater than $\theta_1$ and:

$$\theta_2=\frac{x_2-u_i}{\sigma_i} \text{ and } \theta_1=\frac{x_1-u_i}{\sigma_i}$$

We then derive the following equation from Equation (1):

$$n(\theta_2)(\theta_2-\theta_1)>\Phi(\theta_2)-\Phi(\theta_1)$$

where $n(\theta_2)$ is the probability value of $n$ at the location $\theta_2$. Then we combine this equation with Equation (1):

$$\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_2-\mu_i}{\sigma_i})^2\Big/2}\left(\frac{x_2-x_1}{\sigma_i}\right)>p$$

Since the pdfs are symmetrical with respect to the mean values, the margin functions of $E_p$ where $\mu > x_2$ or $\mu < x_1$ are as follows:

$$\begin{aligned}&\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_2-\mu}{\sigma})^2\Big/2}\left(\frac{x_2-x_1}{\sigma}\right)=p,\text{ where }\mu>x_2\\&\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_1-\mu}{\sigma})^2\Big/2}\left(\frac{x_2-x_1}{\sigma}\right)=p,\text{ where }\mu<x_1\end{aligned} \tag{3}$$

Second, as shown in Figure 10(b), when the mean value $\mu_i$ of a pdf $f_i$ is inside the interval, $\mu_i$ is transformed into the location $\theta_\mu$ which is also the mean value in the normal distribution $n$. Because $\theta_\mu$ is the greatest probability at the location 0 in the normal distribution, we have:

$$n(\theta_\mu)(\theta_2-\theta_1) > \Phi(\theta_2)-\Phi(\theta_1)$$
$$\Rightarrow \frac{1}{\sqrt{2\pi}}(\frac{x_2-x_1}{\sigma_i}) > p$$

The margin formula of $E_p$ where $\mu < x_2$ and $\mu > x_1$ is:

$$\frac{1}{\sqrt{2\pi}}(\frac{x_2-x_1}{\sigma}) = p, \text{where } \mu > x_1 \text{ and } \mu < x_2 \tag{4}$$

From Equations (3) and (4), we derive the following lemma for objects examination.

**Lemma 2:** *Given an interval [$x_1$, $x_2$] and a probability threshold p, the closed area $E_p$ can be approximated as:*

$$\mathit{if}\ \mu > x_2, \mathit{then}\ \frac{1}{\sqrt{2\pi}}e^{-(\frac{x_2-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma}) > p$$

$$\mathit{if}\ \mu < x_1, \mathit{then}\ \frac{1}{\sqrt{2\pi}}e^{-(\frac{x_1-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma}) > p$$

$$\mathit{if}\ \mu \le x_2 \text{ and } \mu \ge x_1, \mathit{then}\ \frac{1}{\sqrt{2\pi}}(\frac{x_2-x_1}{\sigma}) > p$$

*where σ ≥ 0. The error of the approximate $E_p$ can be bounded with the following function:*

$$\mathit{if}\ \mu > x_2\ \mathit{or}\ \mu < x_1, \mathit{then}$$
$$\mathit{Error} \le \left\| (\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_2-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma})) - (\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_2-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma})) \right\|$$

$$\mathit{if}\ \mu \le x_2\ \mathit{and}\ \mu \ge (x_1+x_2)/2, \mathit{then}$$
$$\mathit{Error} \le \left\| (\frac{1}{\sqrt{2\pi}}(\frac{x_2-x_1}{\sigma})) - (\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_1-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma})) \right\|$$

$$\mathit{if}\ \mu \ge x_1\ \mathit{and}\ \mu \le (x_1+x_2)/2, \mathit{then}$$
$$\mathit{Error} \le \left\| (\frac{1}{\sqrt{2\pi}}(\frac{x_2-x_1}{\sigma})) - (\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_2-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma})) \right\| \quad \blacksquare$$

**Proof:** (1) If $\mu > x_2$ or $\mu < x_1$, we choose the greatest probability value from the interval [$\theta_2$, $\theta_1$] to multiply the length of the interval [$\theta_2$, $\theta_1$]. Because the probability value cumulated over the interval [$\theta_2$, $\theta_1$] lies in between the values of n($\theta_2$) and n($\theta_1$), it has the following properties:

$$\mathit{if}\ \mu > x_2, \mathit{then}\ n(\theta_2)(\theta_2-\theta_1) > \Phi(\theta_2)-\Phi(\theta_1) > n(\theta_1)(\theta_2-\theta_1)$$
$$\mathit{if}\ \mu < x_2, \mathit{then}\ n(\theta_1)(\theta_2-\theta_1) > \Phi(\theta_2)-\Phi(\theta_1) > n(\theta_2)(\theta_2-\theta_1)$$

We can bind the error from the equation:

$$\mathit{if}\ \mu > x_2\ \mathit{or}\ \mu < x_1, \mathit{then}$$
$$\mathit{Error} \le \left\| (\frac{1}{\sqrt{2\pi}})e^{-(\frac{x_2-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma}) - (\frac{1}{\sqrt{2\pi}})e^{-(\frac{x_1-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma}) \right\|$$

(2) If $\mu \le x_2$ and $\mu \ge (x_1 + x_2)/2$, the greatest probability value between the interval [$\theta_2$, $\theta_1$] is at the mean location 0. The cumulated probability over the interval [$\theta_2$, $\theta_1$] is between $n$(0) and $n(\theta_1)$. Similarly, if $\mu \ge x_1$ and $\mu \le (x_1 + x_2)/2$, then the cumulated probability over the interval [$\theta_2$, $\theta_1$] is between $n$(0) and $n(\theta_2)$. It can be written as:

$$\mathit{if}\ \mu \le x_2\ \mathit{and}\ \mu \ge (x_1+x_2)/2, \mathit{then}$$
$$n(\theta_\mu)(\theta_2-\theta_1) > \Phi(\theta_2)-\Phi(\theta_1) > n(\theta_1)(\theta_2-\theta_1)$$
$$\mathit{if}\ \mu \ge x_2\ \mathit{and}\ \mu \le (x_1+x_2)/2, \mathit{then}$$
$$n(\theta_\mu)(\theta_2-\theta_1) > \Phi(\theta_2)-\Phi(\theta_1) > n(\theta_2)(\theta_2-\theta_1)$$

Therefore, we derive the error function as follows:

$$\mathit{if}\ \mu \le x_2\ \mathit{and}\ \mu \ge (x_1+x_2)/2, \mathit{then}$$
$$\mathit{Error} \le \left\| (\frac{1}{\sqrt{2\pi}}(\frac{x_2-x_1}{\sigma})) - (\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_1-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma})) \right\|$$

$$\mathit{if}\ \mu \ge x_1\ \mathit{and}\ \mu \le (x_1+x_2)/2, \mathit{then}$$
$$\mathit{Error} \le \left\| (\frac{1}{\sqrt{2\pi}}(\frac{x_2-x_1}{\sigma})) - (\frac{1}{\sqrt{2\pi}}e^{-(\frac{x_2-\mu}{\sigma})^2 \big/ 2}(\frac{x_2-x_1}{\sigma})) \right\| \quad \blacksquare$$

We propose an object examination algorithm (see Figure 11) to determine which of the remaining objects satisfy the query conditions. This algorithm first evaluates the approximate closed area $E_p$ by Lemma 2, and then examines each object by using the area $E_p$. The objects examination first inspects the satisfactions of $t_1$ and $t_2$ in line 4. Next, it intersects the function $\sigma_i(t)$ of an object $O_i$ to the area $E_p$ in line 6. If there is any point intersected by $\sigma_i(t)$ and $E_p$ inside the range between $\sigma_i(t_1)$ and $\sigma_i(t_2)$, then object $O_i$ is one of the answers since there is at least one point inside $E_p$ in the query time period [$t_1$, $t_2$]. Line 6 is simple since we can derive the $\sigma_i(t)$ with Equations (3) and (4). This process produces false positives because some points inside $E_p$ do not satisfy the query condition. Based on Lemma 2, we have the same error bounds for the correctness of the answers obtained from the algorithm.

```
Algorithm Objects Examination ($O$, $q$)

/* Input: $O$ is the set of moving objects. A query $q$ has the
predicates of a time period [$t_1$, $t_2$], a location interval [$x_1$, $x_2$], and a
probability threshold $p$.

Output: $A$ contains the moving objects that satisfy the conditions
of query $q$. */

1. Evaluate $E_p$ from the query $q$
2. while ($O$ is not empty)
3.     get an object $O_i$ from $O$
4. if (($\mu_i(t_1)$, $\sigma_i(t_1)$) or ($\mu_i(t_2)$, $\sigma_i(t_2)$) inside $E_p$)
5.     insert $O_i$ into $A$
6. else if ($\mu_i(t)$ has intersection with $E_p$ between $\mu_i(t_1)$ and $\mu_i(t_2)$)
7.     insert $O_i$ into $A$
8. delete $O_i$ from $O$
9. return $A$
End Objects Examination
```

**Figure 11. The algorithm for objects examination.**

## 5. EXPERIMENTS

We conduct experiments to evaluate our proposal for processing probabilistic spatio-temporal range queries over uncertain object movements. We first show the running time of querying with time instant and time period. Then, we display the percentage of the elimination and examination stages in the total query time. Next, we show the ratio of the dropped objects in the processes of elimination to the examination. Finally, we compare the accuracy of the querying results using our method with a method without using index (labeled as non-indexing method in Figure 12).

Note that a closed form formula for the integral of the normal distribution does not exist, but several approximate functions can be employed. In our experiments, we use the functions in [14]. The error of this integral function is less than $7.5\times10^{-8}$, when performed in the non-indexing method. All the algorithms are implemented in C++ and carried out on a 3.2GHz Intel Pentium IV PC with 1G main memory, running Windows XP SP2.

### 5.1 Experimental Setting

The experimental data is generated as follows. We simulate $N$ objects moving on a line segment forthright [0, 200000], which has the length 200 kilometers. We vary $N$ from 100$K$ to 1$M$. The time unit used in our experiments is 1 second (1s). In the initial stage (i.e., at time $t$ = 0), $N$ objects are uniformly distributed on the forthright. The objects speeds are randomly generated from $v_{min}$ = 10 *meter/sec* to $v_{max}$= *50 meter/sec* (10 *meter/sec* is equal to 36 *km/h* and 50 *meter/sec* is equal to 180 *km/h*.) and the direction is randomly positive or negative. The objects velocity variance is also randomly assigned from 4 *meter*$^2$*/sec* to 16 *meter*$^2$*/sec*. Then the objects start moving. Each object re-generates its speed and variance until the distance between the location it moves to and the previous updated location is up to 500 *meters*. At each time instant we execute 100 random queries, where the length of the location is randomly chosen from 100 to 10000, the time length is from 1 to 600. Note that we have two kinds of queries with *time instant* and *time period*, respectively. We randomly generate a time instant from 1 to 3600 to assign the start time of a time period. We implement R-tree as our indexing structure. We keep all the information including the index and moving objects' data in the main memory instead of in the hard disk since performance of index is out of scope of this study.

### 5.2 Performance Study

Figure 12 presents the query execution time using our approach (labeled by HT since it's based on Hough Transform) in comparison with the non-indexing method for 500$K$ and 1$M$ objects. The HT-process includes two stages: a) elimination, and b) approximate examination. Non-index method retrieves all objects for evaluation by cumulative normal density function, which as shown in the figure, has basically constant costs. The query execution time of HT-process decreases as the query probability increases till the probability threshold exceeds 0.5, because the generated queries with the same location interval are mapped to the same search range when the probabilities exceeds 0.5.

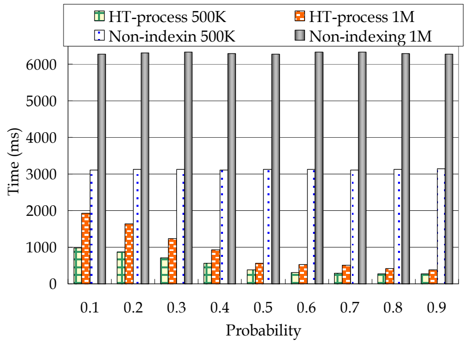


**Figure 12. Execution time for the queries with time instant.**

Figure 13 shows the execution time for querying with time period using our approach. The execution time on 1$M$ objects decreases more significantly than on 500$K$ objects as the probability threshold increases because the elimination and examination are effective. The ratio of the 1M to the 500K in the low probability is larger than the high probability because the time of the examination stage increases more than the elimination. Comparing with the execution time of time instant, the growth of time involves three factors, the extension of the searching range, the increasing number of objects, and the more complicated operations in the examination stage.

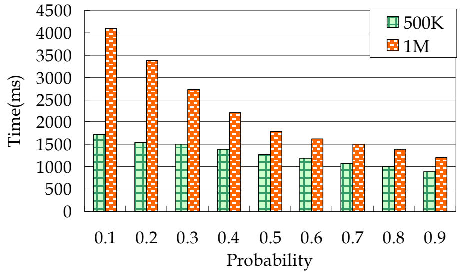


**Figure 13. Execution time for the queries with time period.**

### 5.3 Effectiveness Analysis

In this section, we compare the effectiveness between the elimination and the examination stages in the query process. We also evaluate the correctness of the answers using our methods for the probabilistic range queries.

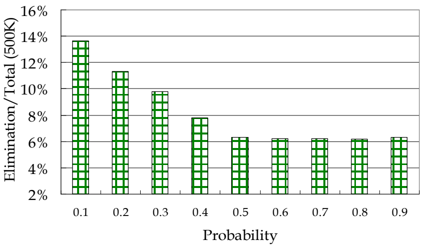


**Figure 14. The proportion of the elimination process.**

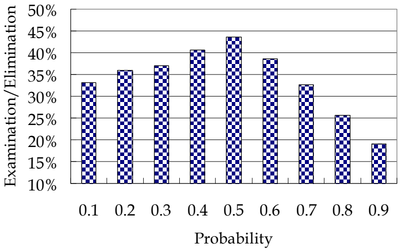


**Figure 15. The proportion of the examination process.**

Figure 14 illustrates the ratio of the remaining objects to the total 500*K* objects with respect to different probabilities for queries with time period. Since the query is not expanded when the probability exceeds 0.5, the corresponding portion of the elimination stage does not decrease. Figure 15 shows the ratio of the answers examined among candidates remaining from the elimination stage. The ratio increases as the probability increases until it is over 0.5. It shows that our query expansion is effective at for the queries with low probability threshold.

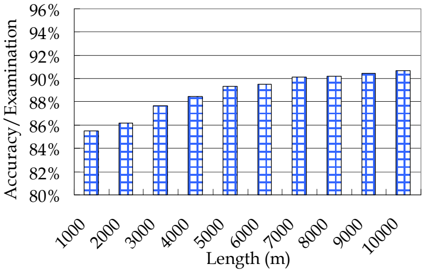


**Figure 17. Accuracy vs. Query Length.**

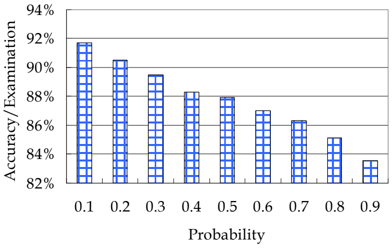


**Figure 16. The accuracy of the examined answers.**

We further evaluate the correctness of the answers among candidates examined at the examination stage. The results of non-indexing method are used as the correct answers. Only the queries with time instant are considered because the non-indexing method does not support queries with time period. As shown in Figure 16, the accuracy decreases when the probability increases because our method produces false positive results. Figure 17 shows the relation between the accuracy and the length of the location interval. The examination stage approximately evaluates the stratifications of the objects according to its mean and variance. The error becomes larger when the mean is near the ends of the interval. The ratio of the objects near the ends rises when the length of the query interval gets shorter.

## 6. CONCLUSION AND FUTURE WORK

In this paper, we investigate the problem of probabilistic query on objects with uncertain movements. We employ Brownian motion model for all the moving objects. In this model, every moving object's uncertain movements are represented as a probability density function. We extend conventional probabilistic range queries (PRQ) with a time period and a probability threshold. To process the query efficiently, we transform all objects uncertain movements into simple points and indexed these points for efficient querying. We developed approximate formulas and an algorithm with error bounds to evaluate probabilities of the moving objects and to ensure the correctness of the querying answers. Experimental results show the effectiveness and efficiency of our approach. We plan to further study the problem in the high dimensional space by developing methods to reduce the query cost via dimensionality reduction of the mapped space.

## 7. ACKNOWLEDGEMENT

Arbee Chen was supported in part by NSC of Taiwan under Grant no. 97-2221-E-004-006-MY3. Wang-Chien Lee was supported in part by NSF of United States under Grant no. IIS-0534343 and CNS-0626709.

## 8. REFERENCES

[1] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Evaluating Probabilistic Queries over Imprecise Data. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp 551-562, 2003.

[2] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying Imprecise Data in Moving Object Environments. In *IEEE Transactions on Knowledge and Data Engineering*, pp 1112-1127, 2004.

[3] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter. Efficient Indexing Methods for Probabilistic

Threshold Queries over Uncertain Data. In *Proceedings of the 30th International Conference on Very Large Data Bases*, pp 876-887, 2004.

[4] W. Feller. *An Introduction to Probability Theory and Its Applications 2nd edn.* (Wiley), pp 340-391, 1957.

[5] A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp 47-57, 1984.

[6] V. Gaede, O. Günther. Multidimensional Access Methods. In *ACM Computing Surveys*, pp 170-231, 1998.

[7] J. Goldstein, R. Ramakrishnan, U. Shaft, J. B. Yu. Processing Queries by Linear Constraints. In *Proceedings of the 16th ACM PODS Symposium on Principles of Database Systems*, pp 257-267, 1997.

[8] P. V. C. Hough. *Method and Means for Recognizing Complex Patterns*, U. S. Patent No. 306964, 1962.

[9] H. V. Jagadish. On Indexing Line Segments. In *Proceedings of 16th International Conference on Very Large Data Bases*, pp 614-625, 1990.

[10] S. Karlin and H.M. Taylor. *A First Course in Stochastic Processes 2nd edn.* (Academic Press), pp. 340-391, 1975.

[11] G. Kollios, D. Papadopoulos, D. Gunopulos, and J. Tsotras. Indexing Mobile Objects using Dual Transformations. In *The International Journal on Very Large Data Bases*, pp 238-256, 2005.

[12] Z. Lei, C. U. Saraydar, and N. B. Mandayam. Paging Area Optimization based on Interval Estimation in Wireless Personal Communication Networks. In *Mobile Networks and Applications*, pp 85-99, 2000.

[13] A. Papoulis. *Probability, Random Variables and Stochastic Processes 3rd edn.* (McGraw-Hill), 1991.

[14] H. Packard. Normal and Inverse Normal Distribution for the HP-67. "*http://www.hpmuseum.org/software/67pacs/67ndist.htm*".

[15] C. Rose. Minimizing the Average Cost of Paging and Registration: A Timer-based Method. In *Wireless Networks*, pp 109-116, 1996.

[16] S. Saltenis, C. S. Jensen, S. T. Leutenegger, and M. A. Lopez. Indexing the Positions of Continuously Moving Objects. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp 331-342, 2000.

[17] A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Modeling and Querying Moving Objects. In *Proceedings of the 13th International Conference on Data Engineering*, pp 422-432, 1997.

[18] Y. Tao, C. Faloutsos, D. Papadias, and B. Liu. Prediction and Indexing of Moving Objects with Unknown Motion Patterns. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp 611-622, 2004.

[19] Y. Tao, D. Papadias, and J. Sun. The TPR*-Tree: An Optimized Spatio-Temporal Access Method for Predictive Queries. In *Proceedings of 29th International Conference on Very Large Data Bases*, pp 790-801, 2003.

[20] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar. Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density. In Proceedings of the 31st international conference on Very large data bases, pp 922-933, 2005.