Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2014, Article ID 365064, 13 pages
http://dx.doi.org/10.1155/2014/365064

Research Article

Continuous Probabilistic Skyline Queries for Uncertain Moving Objects in Road Network

Shanliang Pan, Yihong Dong, Jinfeng Cao, and Ken Chen

College of Information Science and Engineering, Ningbo University, Ningbo 315211, China

Correspondence should be addressed to Yihong Dong; dongyihong@nbu.edu.cn

Received 24 November 2013; Accepted 5 January 2014; Published 4 March 2014

Academic Editor: Zhongwen Guo

In moving environment, the positions of moving objects cannot be located accurately. Apart from the measuring instrument errors, movement of the objects is the main factor contributing to this uncertainty. This uncertainty makes dominant relationship of data instable, which will affect skyline operator. In this paper, we mainly study the continuous probabilistic skyline query for uncertain moving objects in road network. The query point is deemed to be stationary while moving objects are treated as targets with uncertainty described by a probability density function. After defining the notion of dominant probability and probabilistic skyline, we put forward a novel algorithm to deal with continuous probabilistic skyline query on road network. Firstly, we compute the dominant probability and skyline probability to get initial permanent p-skyline set. Then we define events to predict the time when dominant relationship between moving objects may change. Furthermore, we track and calculate events to update the probabilistic skyline in an incremental way. Two pruning strategies are proposed to cancel invalid events and objects in a bid to diminish search space. Finally, an extensive experimental evaluation on real datasets shows that probabilistic skyline sets in road network can be updated by the proposed algorithm. It demonstrates both efficiency and effectiveness.

1. Introduction

Skyline query aims to find a subset where all objects are not dominated by any other object in the dataset, helping users in multicriteria decision making, data mining and visualizing for database, and so forth.

In mobile database, a moving object reports its position and velocity to database service through wireless communication interface. Because of time delay and other technical limitations, the position obtained usually deviates from actual one. The deviation causes what is called uncertainty, which leads to instability of dominant relationship between objects. In real world, objects are restricted in road network, such as track, road, or highway. Position uncertainty of moving objects including its environment should be considered in skyline operation.

Until recently, a lot of work on skyline query had been focused on a static dataset, where the distances from query point to target objects are invariant. With rapid development of GPS technology and mobile devices, the location-based service (LBS) [1], which is pervasive in daily life, is becoming one of the most important applications in spatiotemporal database. However, in moving environment, moving objects whose position is detected by GPS system alternately are always in motion. Delay or error of the sensor in GPS, especially the self-motion of moving objects, makes the location of moving objects uncertain. Despite the existing uncertainty, the position is often treated as being accurate to be queried in many studies [2, 3]. Study on continuous skyline query on moving objects was first proposed by Huang et al. [4]. In his work, position of moving objects is considered as the data with certainty.

In fact, because of the sensors' error and the objects' movement, the position of objects in moving environment is uncertain and vague. Recently, a few researchers have been aware of this imprecision and made some contributions [5–7]. In our approach, the location of query object is exactly known as a stationary object, but the location of moving target objects is characterized by a certain distribution instead of a precise point. For example, a weapon-related crime takes place somewhere, which is located in point O, as Figure 1 shows. Police cars should be dispatched to intervene. The adequate police strength including the number of policemen and equipment in the dispatched car is needed. Also a car as near to the crime scene as possible is another factor for consideration in order to arrive in time. In Figure 1, table shows the parameters of each police car, including position of the cars, number of policeman, equipment level, and network distance to the query point O, where the value of equipment level is used to quantify the level of equipment, with greater value corresponding to higher level. If the police cars are treated as certain points, cars A and E are skyline objects. However, the cars are moving quickly, so their positions are uncertain to some extent. For example, suppose that the true position of car A is at point A', where the distance from A' to O is 130 instead of 125, and car B's actual position is at point B' whose distance to O is 125. In this case, car B can dominate car A, so car A is not skyline object. Because of road network limitation, the uncertain area is represented as a line centered on its acquired location, as Figure 1 shows. In order to simplify this scene, probability density function $f(t)$ assumes uniform distribution and the police car B is possible to be skyline object.

In this paper, we address the problem of continuous probabilistic skyline query for uncertain moving objects in road network. In our study case, the query object stays still, while the target objects are moving with uncertainty which is characterized by both a closed uncertainty region and a probability density function (pdf). The main contributions are as follows.

(1) Probabilistic skyline on uncertain moving object in road network is introduced as a new important issue in decision support or navigation systems.

(2) By analyzing the dominant relationship between moving objects in uncertain position environment, the dominant probability and skyline probability are defined in case the query is stationary with targets moving.

(3) Based on trigger events which are presented to track the change of dominant relationship, a novel algorithm Probabilistic Skyline query with Uncertainty in Road network (PSUR) is proposed to deal with the dynamic skyline query with uncertainty. A series of pruning strategies are introduced to optimize and fasten this incremental algorithm.

(4) A great deal of experiments on two truth datasets are conducted to analyze the uncertain region, numbers of static dimensions, velocity of moving objects, and dataset size that affect the algorithm. Contrast experiments from literature [8] are made to validate the proposed PSUR. The experimental results show that PSUR is effective and efficient.

The rest of this paper is organized as follows. In Section 2, we summarize the related work of skyline computation. In Section 3, after describing basic notion of skyline and uncertain model for moving objects, we define dominant probability and probabilistic skyline. Section 4 introduces trigger events to track how dominant probability relationship varies when objects are moving with uncertainty. Two pruning strategies are adopted to reduce the search space. Section 5 describes PSUR algorithm to update p-skyline set. Section 6 gives experimental evaluation on two real datasets. Finally, the conclusion is reached in Section 7.

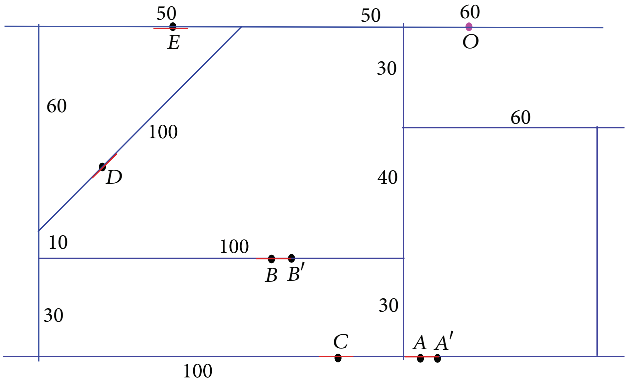


Figure 1: An example of skyline query for uncertain data.

2. Related Work

Skyline queries are hot areas of current database research which recently have attracted more and more attention. It is introduced into relational database firstly by Borzsony et al. [9] with two proposed processing algorithms Block-Nested-loops algorithm (BNL) and Extended Divide & Conquer algorithm (D&C). As an improved method of BNL proposed by Dr. Chomicki et al. Sort-Filter-Skyline algorithm (SFS) [10] constructs the multidecision dominative order chains for the ordered data. NN algorithm [11], based on R-tree, searches the nearest neighbor recursively. It speeds up the skyline query by reducing the comparison numbers of BNL among objects. Nevertheless, it will cost more time and space in searching subspace repeatedly. Overcoming this limitation, the sorted R-tree based BBS [12] is proposed. It is one of the best skyline query methods in centralized datasets.

In recent years, skyline query has been extended to the dynamic datasets. The R-tree based I-Eager and I-Lazy algorithms were brought forth by Tao and Papadias [13] who firstly studied how to update and maintain the skyline results in dynamic datasets.. Tian et al. [14] introduced GICSC updating skyline query sets dynamically which better fits for low dimension metrics. Kontaki et al. [15] exerted efforts to maintain k-dominant skyline objects with maximum user's preference. Huang et al. [4] probed into the continuous skyline query for certain moving objects. It assumes that all points including the query point move in a predictable way. After analyzing dominating relationship between points in the space, Huang presented a continuous tracking algorithm-CSQ to maintain skyline sets dynamically.

In sensor networks and moving environment, the characters of the objects are not exactly known due to the limitations of measuring equipment and objects' movement. A lot of work has focused on uncertain data. The authors in [5] studied the execution of probabilistic range and nearest-neighbor queries in mobile environment. The author in [16] focused on the situation in which the location of a query object is not exactly known. In research of skyline query field, Fiedler [17] firstly proposed skyline operator on uncertain data in his dissertation. Pei et al. [18] proposed a probabilistic skyline model for multiinstance data, where each object is part of the skyline with a certain probability. They presented two algorithms BUM and TDM to study skyline query on uncertain data of the possible world model (PWM). Lian and Chen [19] studied reverse-skyline query. They modeled the probabilistic reverse skyline query on uncertain data, in both monochromatic and bichromatic case, and proposed two effective pruning methods, MPRS and BPRS, to reduce the search space of query processing. Zhang et al. [20] explored how to maintain skyline sets when uncertain data are updated. An AR-tree is constructed for all uncertain data, and the maintenance is inserting, deleting, and updating on this AR-tree. The authors in [21] designed partitioning method to compute skyline probabilities for discrete data with uncertainty. The authors in [8] investigated skyline probabilities with a parametric form pdf (e.g., a Gaussian function or a Gaussian mixture model). The authors in [22] proposed a sliding window skyline model to study the execution of the probabilistic skyline query over uncertain data streams [23]. The authors studied a new problem of range-based skyline queries. Two novel algorithms I-SKY and N-SKY were presented to solve the probabilistic and continuous range-based skyline queries.

In road network, Huang and Jensen [24] assumed that the user's movement is constrained to a road network. The authors defined route nearest-neighbour skyline queries to consider the computation efficiency. Deng et al. [25] studied multisource skyline query in road networks. Three different road networks of multisource query methods were presented. The authors in [26] proposed a new method to process continuous skyline in road network based on precomputing the shortest range data of targets. The authors in [27] introduced route skyline computation in a multiattribute graph. Top routes are computed iteratively in an efficient way and pruning technique is adopted in order to reduce the search space. The authors in [28] focused on extracting the path skylines and proposed PathSL to generate an optimal skyline for moving objects.

In spite of much work focusing on uncertainty of skyline queries or route skyline queries, there is little work completed for skyline query concerning uncertainty of moving objects. The authors in [4, 22–25] ignored the uncertainty of moving objects, while the authors [8, 18, 21] dealt with discrete data with independent dominant relationship between objects. It is the first time to compute continuous probabilistic skyline query with regard to uncertainty of moving objects in road network.

3. The Probabilistic Skyline for Uncertain Moving Objects

3.1. Skyline on Certain Points. Suppose that the point set is $S = \{s_1, s_2, \ldots, s_{\text{Num}}\}$, where Num is the number of points, each point in an n-dimensional numeric space $D = \{d_1, d_2, \ldots, d_n\}$. The original concept of skyline is based on the notion of dominance. Let $X = (x_1, x_2, \ldots, x_n)$, $Y = (y_1, y_2, \ldots, y_n)$, $X \in S$, $Y \in S$, if $\forall i$, $x_i \le y_i$ $(1 \le i \le n)$, and $\exists j$, satisfy $x_j < y_j$ $(1 \le j \le n)$, then X is said to dominate Y, donated by $X \prec Y$. Given a set of objects S, an object X is a skyline point if there is not any other point dominating X. The skyline on S is the set of all skyline points.

Table 1

Position of police car	A	B	C	D	E
Number of policemen	6	6	6	4	5
Equipment level	3	3	2	3	2
Network distance to O	125	130	140	155	90

3.2. Uncertain Model for Moving Objects. After Wolfson et al. [1] firstly studied the uncertainty of moving objects, a lot of work has focused on this field [5–7]. An uncertain region [5] of a moving object Mi at time t, denoted by $Ui(t)$, is a closed irregular region with a velocity $\vec{v}$ such that the recorded location O_i can be found only inside this region. The pdf represents the probability density distribution of an object in its uncertain region. The uncertainty pdf [5] of an object Mi, denoted by $f(o_i)$, is a pdf of O_i, that has a value of 0 outside $Ui(t)$.

3.3. The Dominant Probability. What we mainly study on is as follows. There is a set of moving target objects whose centers are $O = \{o_1, o_2, \ldots, o_{\text{Num}}\} \in R^n$ and a stationary object q as a query point to continuously compute updated probabilistic skyline dataset.

Dissimilar to the traditional skyline query, in moving environment, the spatial location is changing with time. All attributes are divided into dynamic attributes and static ones. Let us suppose that there are m dynamic attributes, k static attributes, and n attributes where $n = m + k$. For each object $O_i = (x_1^i, x_2^i, \ldots, x_d^i)^T$, the static attributes construct a vector denoted by $O_i^S = (x_{S1}^i, x_{S2}^i, \ldots, x_{Sk}^i)^T$, while the dynamic ones are made up of a vector $O_i^d = (x_{d1}^i, x_{d2}^i, \ldots, x_{dm}^i)^T$, where $n = k + m$. For instance, in Figure 1, the number of police and equipment level belong to static attributes because they are invariant in our study case. However, the distance from O_i to q is varying continuously with police car O_i moving. In order to simplify the description, dynamic attributes are supposed to include only spatial position (Table 1).

Definition 1 (dominant probability). Let O_i, O_j be two moving objects with uncertainty and q be a stationary query point. Then probability that O_j dominates O_i is

$$p(o_j \prec o_i) = \int_{Ur_{O_j}} \int_{Ur_{O_i}} f_{O_i}(\vec{x}_i)\, f_{O_j}(\vec{x}_j)\ \text{isdom}(\vec{x}_j, \vec{x}_i)\, do_i do_j, \tag{1}$$

where $f_{O_i}(\vec{x}_i)$ and $f_{O_j}(\vec{x}_j)$ are probability density function of object O_i and O_j, $\vec{x}_j \in Ur(O_j)$, $\vec{x}_i \in Ur(O_i)$, and function

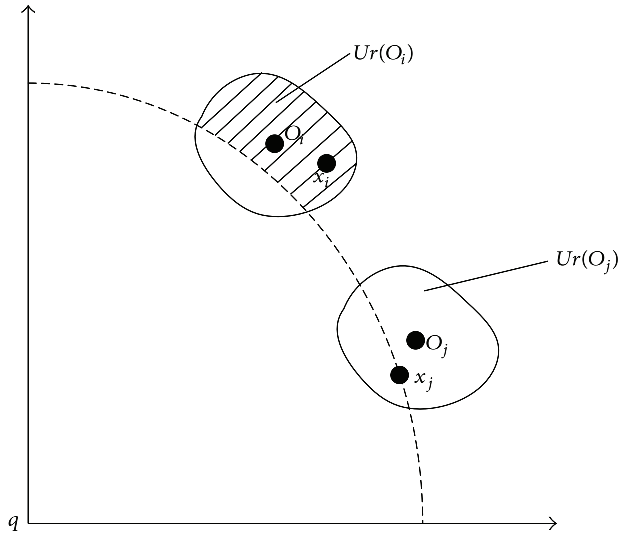


FIGURE 2: Dominant probability.

isdom $(\vec{x}_j, \vec{x}_i) = \begin{cases} 1 & \text{dist}(\vec{x}_j,q)\leq\text{dist}(\vec{x}_i,q) \\ 0 & \text{otherwise} \end{cases}$, $\text{dist}(\vec{x}_j, q)$ is Euclidean distance between x_j and q.

According to the definition of dominance in (1), we maintain that O_j can have dominance probability on O_i only if the static attributes of O_j dominate the static ones of O_i. As shown in Figure 2, for arbitrary point x_j in $Ur(O_j)$, each point x_i in $Ur(O_i)$ that satisfies $\text{dist}(\vec{x}_j, q) \leq \text{dist}(\vec{x}_i, q)$ will be dominated by x_j, as the hatched area shows.

3.4. The Skyline Probability

Definition 2 (skyline probability). The skyline probability of object O_i is the likelihood that object O_i is not dominated by any other object and is defined below:

$$\begin{aligned}\Pr(o_i) &= \Pr\left(\underset{O_j\in O}{\wedge} O_j! \prec O_i\right) \\ &= \int_{Ur_{O_i}} f_{O_i}(\vec{x}_i) \\ &\quad \times \prod\left(1 - \int f_{O_j}(\vec{x}_j)\,\text{isdom}(\vec{x}_i, \vec{x}_j)\,do_j\right) do_i,\end{aligned} \tag{2}$$

where $f_{O_i}(\vec{x}_i)$ and $f_{O_j}(\vec{x}_j)$ are probability density function of object O_i and O_j, respectively, $\vec{x}_j \in Ur(O_j)$, $\vec{x}_i \in Ur(O_i)$.

Definition 3 (p-skyline). Let $p \in [0\cdots 1]$ be a threshold. The p-skyline is the set of objects for which the following property holds:

$$S_\tau = \{O_i \in O \mid \Pr(O_i) \geq p\}. \tag{3}$$

3.5. Uncertain Model in Road Network. The road network can be treated as a nondirection graph $G = \langle V, E, W\rangle$, where V is node set representing crossroad, E is edge representing roads between two crossroads, and W is length of E.

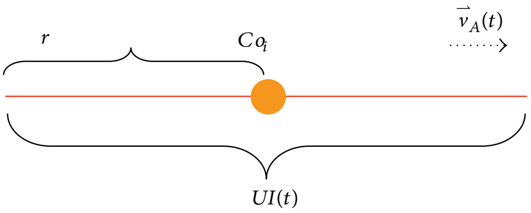


FIGURE 3: Uncertain model in road network.

In road network, the objects are limited movement, so the uncertain domain is denoted by a line segment with $f(t)$ distribution, which $f(t)$ is pdf, as Figure 3 shows.

Denote $d(v, v')$ by the shortest path between nodes v and v'. $d(v, v') = \infty$ if there exists no path from v to v'. In road network, the distance is represented by shortest path.

Definition 4 (minimum distance function). The minimum distance between uncertain object O_i to query q at time t is denoted by $d_{\min}(q, o_i) = d(q, o_i) - r$.

Definition 5 (maximum distance function). The maximum distance between uncertain object O_i to query q at time t is denoted by $d_{\max}(q, o_i) = d(q, o_i) + r$.

In continuous movement, the dominant relationship of two objects will change. For two moving objects O_i and O_j,

$$p(o_j \prec o_i) = \begin{cases} 1 \\ \quad d_{\max}(q, o_j) \leq d_{\min}(q, o_i) \\ 0 \\ \quad d_{\min}(q, o_j) \geq d_{\max}(q, o_i) \\ \int_{Ur_{Oj}} \int_{Ur_{Oi}} f_{O_i}(\vec{x}_i)\, f_{O_j}(\vec{x}_j)\,\text{isdom}(\vec{x}_j, \vec{x}_i)\,do_i do_j \\ \quad d_{\min}(q, o_j) < d_{\max}(q, o_i), \\ \quad d_{\max}(q, o_j) > d_{\min}(q, o_i). \end{cases} \tag{4}$$

The maximum and minimum distance functions maybe intersect with each other among moving objects. Two distance functions of object O_i and O_j are given in Figure 4. The maximum distance of O_i is less than the minimum distance of O_j prior to time t_1, so O_j cannot dominate O_i. O_j might begin to dominate O_i from t_1 to t_2, as the hatched area shows, so does it from t_3 to t_4, and from t_5 to t_6.

4. Tracking by Events

The probabilistic skyline set is updated in an incremental way. The key step is how to predict the time when the dominant relationship changes. It's hard to know the accurate time when p-skyline just change. But we can estimate period of time during which the p-skyline may change.

4.1. Trigger Events. As shown in Figure 4, from time t_1 to time t_2, the dominant relationship between O_i and O_j might change, because the maximum distance of O_i is greater than the minimum distance of O_j. So we call the time between

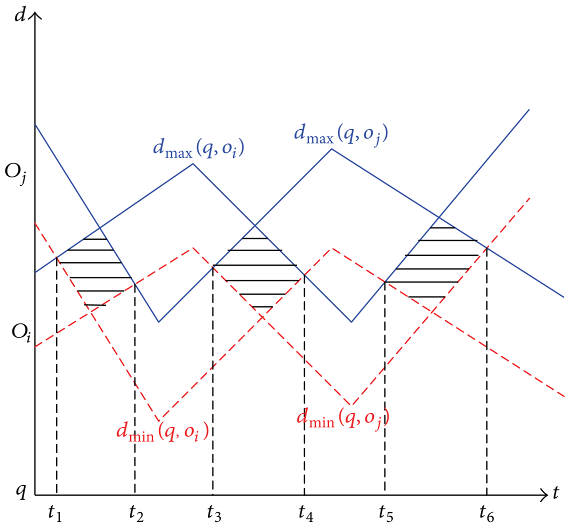


Figure 4: Transformation of dominant relationship.

t_1 and t_2 an event. An event on O_i can be presented as $event(o_i, o_j) = \langle o_i, o_j, s\_time, e\_time \rangle$, where O_i and O_j are two objects whose dominant relationship may be varied when s_time and e_time are start time and end time of the event, respectively.

4.2. The Pruning Strategy for Events. By analyzing relationship of objects we know that if static characters of any two objects O_i and O_j have dominant relationship, supposing that $O_j^S \prec O_i^S$, an event maybe occur. With the query object moving, a considerable amount of events will occur. In order to improve effectiveness of our algorithm, we propose a series of event pruning strategies.

Pruning Strategy 1. If the dominant relationship in static attributes between two moving objects O_i and O_j does not exist, the intersection of those two distance functions will cause no variation to p-skyline set. In this case, the intersection will not cause any event.

Proof. Suppose $\vec{o}_i = (x_1^i, x_2^i, \ldots, x_d^i)$, $\vec{o}_j = (x_1^j, x_2^j, \ldots, x_d^j)$, whose static characters are $\vec{o}_i^S = (x_{s1}^i, x_{s2}^i, \ldots, x_{sk}^i)$ and $\vec{o}_j^S = (x_{s1}^j, x_{s2}^j, \ldots, x_{sk}^j)$, respectively. It is known that the static characters of O_i and O_j do not possess any dominant relationship, so $\exists k_p, k_q, x_{k_p}^i > x_{k_p}^j$ and $x_{k_q}^i < x_{k_q}^j$. Even considering their dynamic attributes, the dominant relationship does not exist at any time. In conclusion, if the dominant relationship in static attributes between two objects does not exist, the intersection of their distance functions will not cause any change for their dominant probability. Events cannot take place. □

Pruning Strategy 2. Suppose $\exists o_i, o_j, p(o_j \prec o_i) = 1$. Event$(i, j, t_{\text{begin}}, t_{\text{end}})$ is an event interrelated to O_i and O_j. If $\exists$event$(O_i, O_k, t'_{\text{begin}}, t'_{\text{end}})$ $(0 < k < \text{Num}$ and $k \neq i)$ and $t'_{\text{begin}} < t_{\text{begin}}$, the computation for event$(O_i, O_k, t'_{\text{begin}}, t'_{\text{end}})$ is invalid before t_{begin}. However, if $t'_{\text{end}} < t_{\text{begin}}$, event$(i, k, t'_{\text{begin}}, t'_{\text{end}})$ is also invalid.

Proof. Because $p(o_j \prec o_i) = 1$, it is known that $p(o_i \prec o_j) = 0$ prior to time t_{begin}. The update for event$(O_i, O_k, t'_{\text{begin}}, t'_{\text{end}})$ is invalid. It is also invalid when $t'_{\text{end}} < t_{\text{begin}}$. □

5. Continuous Probabilistic Skyline Queries Algorithm

5.1. Initialization. The initialization framework is presented in this section. Each object has static and dynamic attributes, so we should compute the dominance relation on static attributes firstly. For two objects O_i and O_j, O_j might dominate O_i only when $o_j^S \prec o_i^S$ (Algorithm 1). In order to compute fast, static skyline set denoted by S_{static} is introduced if only the dominant relationship for static characters is considered.

In any case objects move, S_{static} always belongs to skyline. In initialization step, the moving path and distance function of each moving object need to be precomputed according to the information of road network and moving objects, apart from S_{static}.

5.2. PSUR Algorithm. Not all skyline probability of objects will change at one moment. If maximum/minimum distance function of one object cuts that of others in Cartesian coordinates, skyline probability of this object maybe vary (Algorithm 2). In other words, trigger events include all possible variations of p-skyline for each moving object.

In order to simplify the problem, we suppose that all moving objects preserve their velocity with uniform speed. If not, the precomputing cannot be processed in advance. All events should be recomputed again. The movement of moving objects to query point can be picked up with its shortest path to query point. The intersecting time for the distance function can be recomputed and put into event queue sorted by time in ascending order. For each time, skyline probabilities of moving objects under trigger events need to be updated.

5.3. Algorithm Analysis and Discussion. The cost incurred by our method consists of three components: initialization, events computing, and updating by tracking.

In initialization period, computing static dominant relationship will cost $O(d \cdot n(n-1)/2)$, where d is static dimensional degree. The most cost is skyline probability computation. The famous Simpson integration method, which is introduced to compute skyline probability of each moving objects Num, and time interval t. Probabilistic $O((\text{Len}/h)^2)$, where Len is uncertain segment length and h is step width of Simpson, so the cost of skyline probability of each moving object is $O((\text{Len}/h)^2(n-1))$. After skyline probability computation, judgment for p-skyline objects will cost $O(n)$. Above

```
Input: one query object, $n$ moving objects;
Output: Initialization $p$-skyline;
$S_{\text{static}}$ = NULL
for any point $o_i$ in $O$
  compute_path($o_i$); //get the moving path of objects
  compute_distance($o_i$); //computing distance function
  $O_i$ = NULL
  for any point $o_j$ ($j$! = $i$) in $O$
        compare($o_j^S$, $o_i^S$); //ensure whether $O_j$ dominate $O_i$ in static attributes;
          if $o_j^S \prec o_i^S$
            flag($o_j$, $o_i$) = 1;
            push $o_j$ into the $O_i$;
        else
            flag($o_j$, $o_i$) = 0;
      if the $O_i$ = NULL
  $S_{\text{static}} = S_{\text{static}} \cup o_i$;
```

Algorithm 1: Initialization.

```
Input: the set of moving objects $D$, query point $q$, $p$ is the probabilistic threshold, the time interval $[t_s, t_e]$.
Output: the $p$-Skyline at any time instant $t$ within $[t_s, t_e]$.
initialization();
// compute events;
for any point $o_i$ in $O$
  for each point $o_j$ in $O_i$
        compute begin time $s\_time$ and end time $e\_time$ of each event;
        if $s\_time \le t_e$ && $e\_time \ge t_s$
        push it into $Q_e$; //valid event;
  //handle events;
$time = t_s$, $L_p = L_s$
  while ($time \le t_e$)
   while ($Q_e$ != NULL)
        $e = Q_e \cdot$ erase(); // pop queue's head;
         while ($e \cdot t_{\text{begin}} \le time$)
        $Q_{\text{handle}} \cdot$ push_back($e$); // push $e$ into queue;
        $e = Q_e \cdot$ erase();
    handle($Q_{\text{handle}}$); //update $Q_e$ and $S_p$
  time++;
```

Algorithm 2: PSUR algorithm.

all, time cost in initialization is $O(d \cdot n(n-1)/2 + (\text{Len}/h)^2(n-1)+n)$.

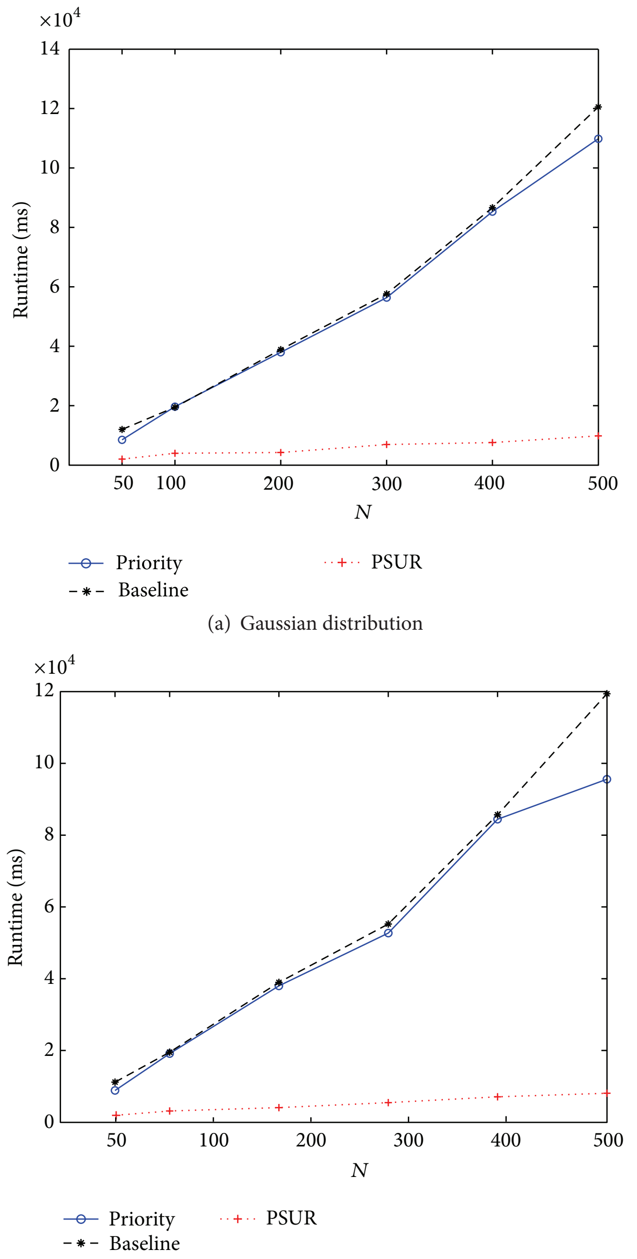
In second step, the worst cost of comparison for static dominant relationship between objects is $O(n(n-1)/2)$. The interaction time for arbitrary two moving objects' distance function will cost $O(\delta_1)$, where δ_1 is constant. Thus it will cost $O(\delta_1 \cdot n(n-1)/2)$ in this period.

Handling events generates at most $n(n-1)/2$. In the worst case, each event is valid in time period $[t_s, t_e]$, and this requires $(t_e - t_s)$ times to deal with events. In every processing, skyline probability related to trigger events will cost $O((\text{Len}/h)^2(n-1))$, while update computing for event is $O(\delta_2)$, where δ_2 is constant. The total cost in this period is $O((\text{Len}/h)^2(n-1)\cdot(t_e-t_s)\cdot n(n-1)/2+\delta_2\cdot(t_e-t_s)\cdot n(n-1)/2)$.

In conclusion, the total cost is added together for these three periods, which is equivalent to $O(n^2)$.

6. Experimental Evaluation

6.1. Datasets. Two real road networks are used to test the effect and efficiency of the proposed algorithm PSUR. One is the famous seashore city Oldenburg in Germany, which includes 6105 nodes and 7036 edges. The other is Cixi city of China, which contains 244 intersection nodes and 407 edges, much smaller than the first one. We assume that the uncertain area is segment along the road. Two distributions of uniform and Gaussian distribution for probability density function are adopted because these two functions are by far the most important and commonly used in statistics. Moving objects are generated randomly. Baseline and priority algorithms proposed in [8] are used to compare with our method PSUR to verify efficiency and effect. The main parameters are segment length Len, static dimension number d, numbers of



(a) Gaussian distribution

(b) Uniform distribution

Figure 5: Numbers of moving objects effects of the performance in Cixi city (d = 5, Len = 10 and t = 20).

Table 2: Event size in Cixi road network (d = 5, Len = 10, t = 20).

Numbers of moving objects	Event size before pruning	Event size after pruning
50	669	35
100	2721	95
200	10389	205
300	35009	400
400	62723	514
500	96890	736

Table 3: Event size in Oldenburg road network (d = 5, Len = 10, t = 20).

Numbers of moving objects	Event size before pruning	Event size after pruning
50	245	5
100	455	15
200	1557	41
300	3945	131
400	6464	236
500	9951	367

moving objects Num, and time interval t, and probabilistic threshold p is 0.5 in all experiments.

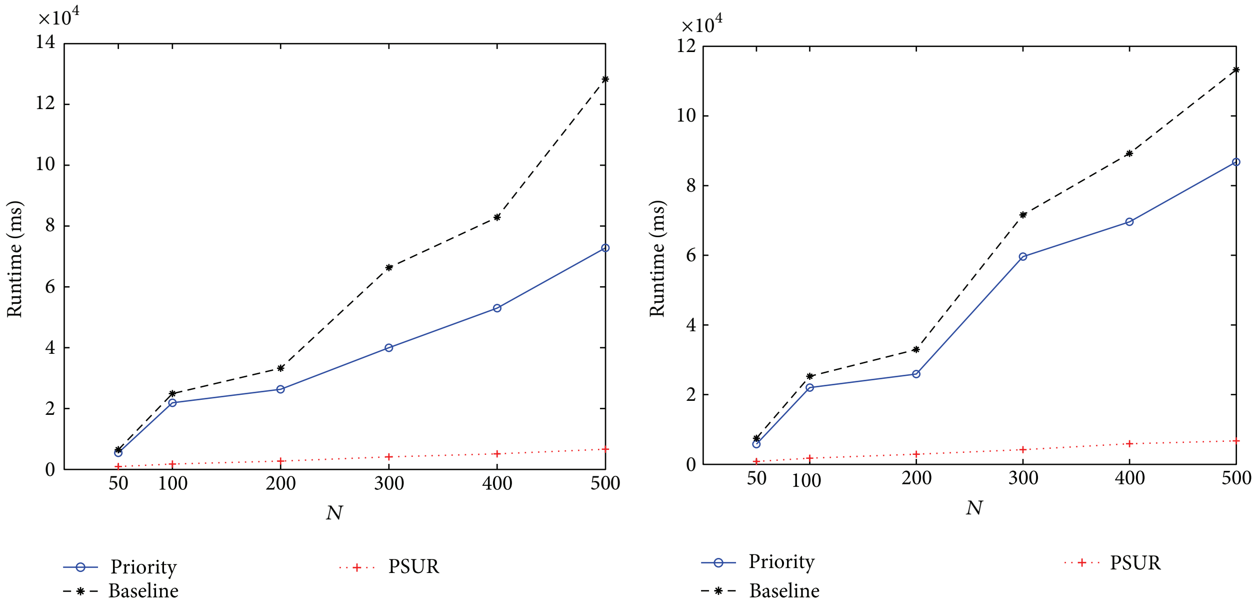
We conducted our experiments on desktop PC running on Windows XP professional. The PC has Intel Core 2Duo 2.93 GHz and 3 GB RAM memory. All experiments were coded in Visual C++ 2008.

6.2. Numbers of Moving Objects. In this experiment, suppose d = 5, Len = 10, and t = 20. Figure 5 shows performance of PSUR versus baseline and priority [8] when numbers of moving objects vary on Cixi's road network, both in uniform distribution and Gaussian distribution. Likewise, Figure 10 demonstrates the performance on road network in Oldenburg. The figures show that runtime increases with object's numbers whichever the model is. No matter what model is used, uniform or Gaussian, the cost in the same dataset is similar. Among these three methods, the response time of our algorithm is only one-tenth of the other two methods. It is because baseline and priority need recompute the probabilistic skyline set at each time while PSUR is an incremental method.

If there is the same number of moving objects in two real networks, the density of moving objects in Cixi is bigger than that in Oldenburg, because Cixi is smaller than Oldenburg. Therefore the chances for interaction of moving objects in Cixi are more those that in Oldenburg. Trigger events generated in Cixi are evidently more than those in Oldenburg, as Tables 2 and 3 show. The density of moving objects leads to this difference, because it is more sparsely distributed in Oldenburg than in Cixi for the same number of objects. As a result, by observing the performance between two city networks under the same data model, such as Figures 5(a) and 6(a), it is noted that the runtime in the smaller city Cixi costs a little more than that in Oldenburg.

The pruning strategy has done effective work on event size, as Tables 2 and 3 show. Event size will grow with the number of moving objects, because the number of the intersection of distance function rises.

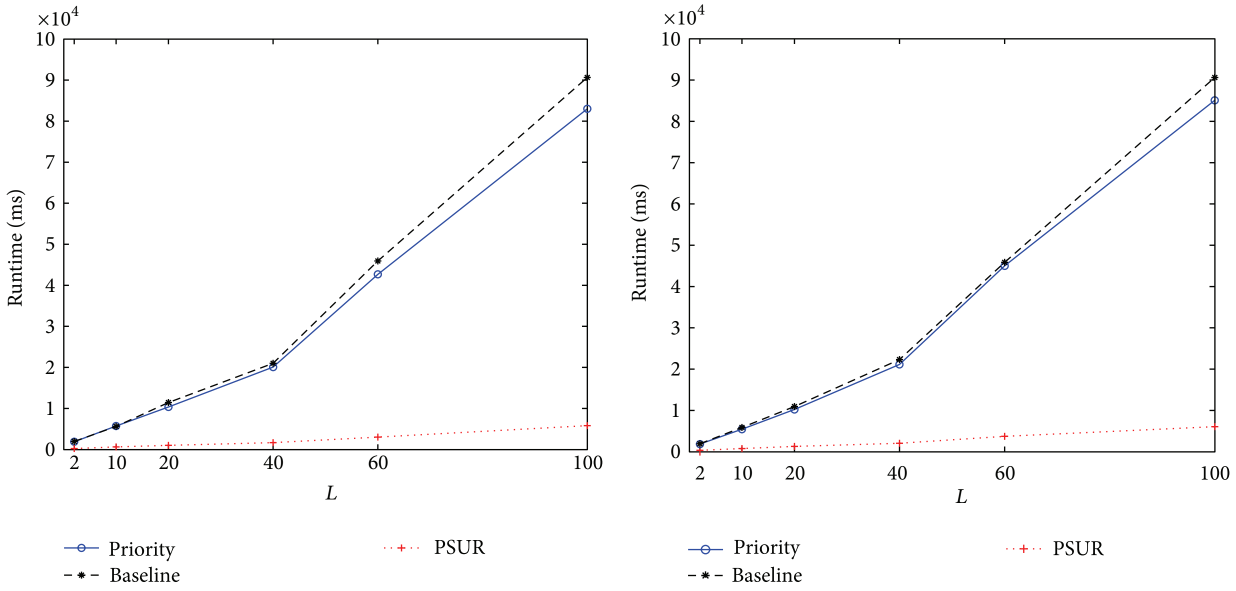
6.3. The Effect of Uncertain Segment Length. In this section, parameters are as follows: Num = 50, d = 5 and t = 20. Figures 7 and 8 show performance of three algorithms when segment length of uncertainty varies on Cixi road network



(a) Gaussian distribution

(b) Uniform distribution

FIGURE 6: Numbers of moving objects effect of the performance in Oldenburg city (d = 5, Len = 10 and t = 20).



(a) Gaussian distribution

(b) Uniform distribution

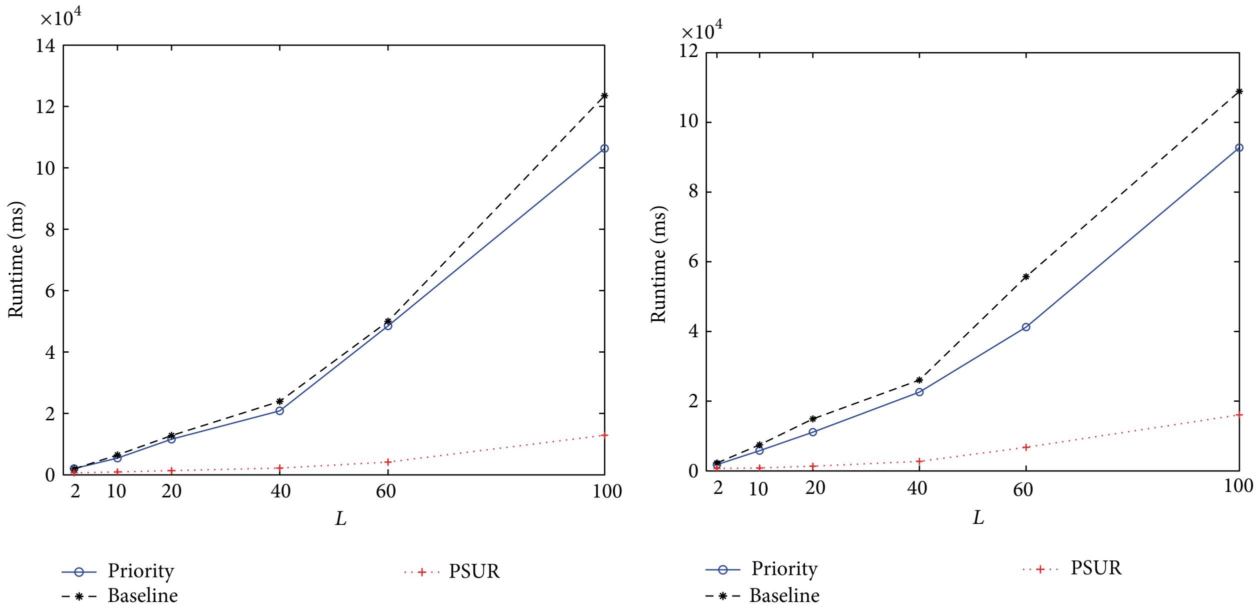
FIGURE 7: Segment length effects of the performance in Cixi city (Num = 50, d = 5, t = 20).

and Oldenburg city, respectively, both in uniform distribution and Gaussian distribution.

The magnitude of the uncertain length may also affect the performance of algorithm PSUR. It is seen that the longer the segment length is, the more runtime is required. If the length becomes much longer, for any two objects O_i and O_j, the probability that $\Pr(O_j \prec O_i) \neq 0$, $\Pr(O_j \prec O_i) \neq 1$ happen becomes greater. According to the definition of event, the size of event queue will increase, so will the times to track and handle events. The calculation of dominance probability and skyline probability grows with this increasing probability, so that the cost will go up.

It is shown that the segment length of uncertainty affects event's size, as shown in Tables 4 and 5. With increase of segment length, the number of events becomes great.

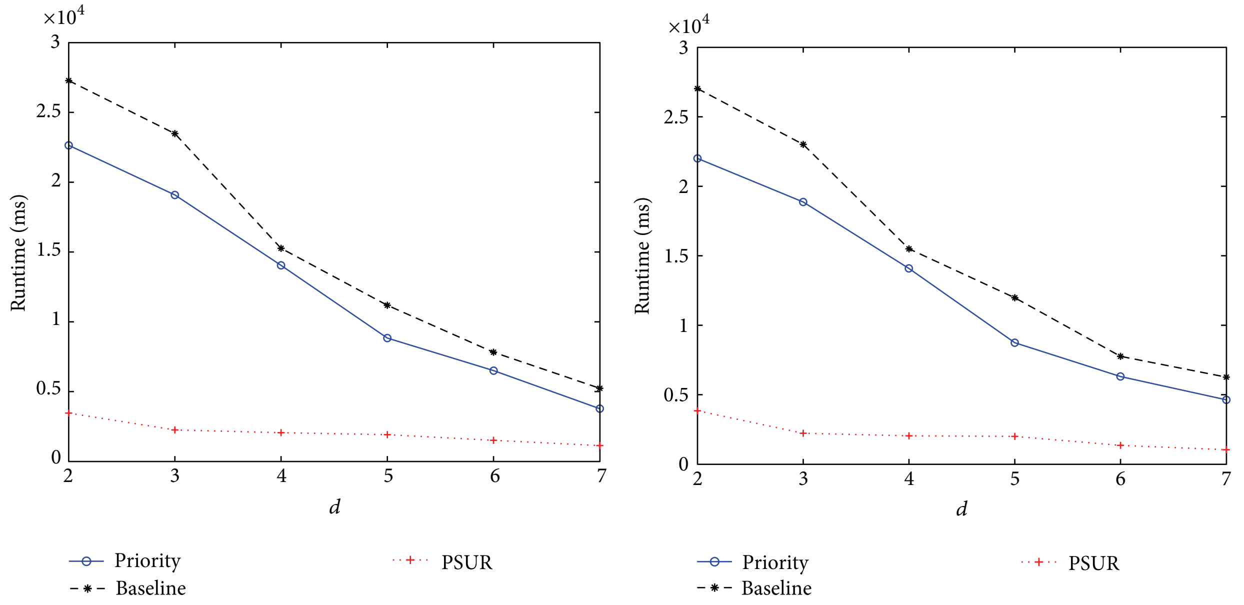
6.4. The Effect of Static Attribute Dimensionality. We will discuss the effect that multidimensional attributes impact on our algorithm over real road network. The result is shown in Figures 9 and 10. It is known that the higher dimension



(a) Gaussian distribution

(b) Uniform distribution

FIGURE 8: Segment length effects of the performance in Oldenburg city (Num = 50, d = 5, t = 20).



(a) Gaussian distribution

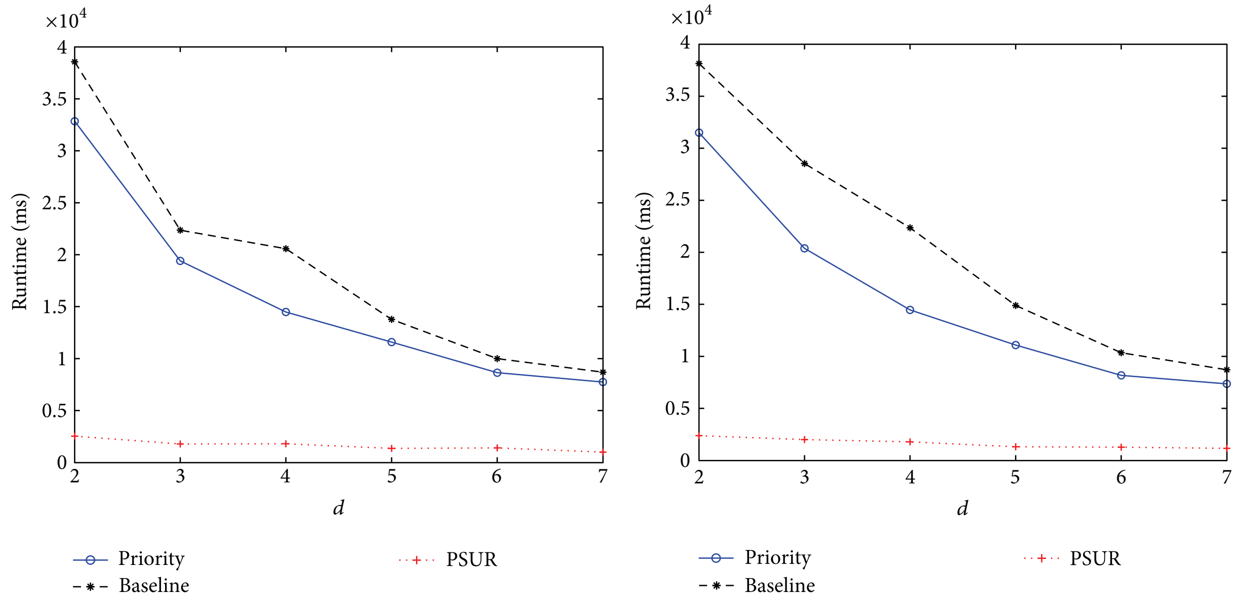
(b) Uniform distribution

FIGURE 9: Static attribute dimensionality effects of the performance in Cixi city (Num = 50, Len = 20, t = 20).

results in less run-time. This is because the computation of dominance relation on static attributes is carried out only once at the beginning. The higher the static dimensions are, the less objects that dominant o_i are. The less the time cost on computation of initializing adjacency list and event queue is, the less the total runtime is.

Tables 6 and 7 show the event size changes with static dimensionality of PSUR. Unlike preceding two experiments, in this case, when static dimensionality d increases, the event size reduces instead of increasing. This is due to the fact that, when d increases, the number of moving objects which have dominant relationship in static dimensionality decreases. The distance function will interact less, so will valid events.

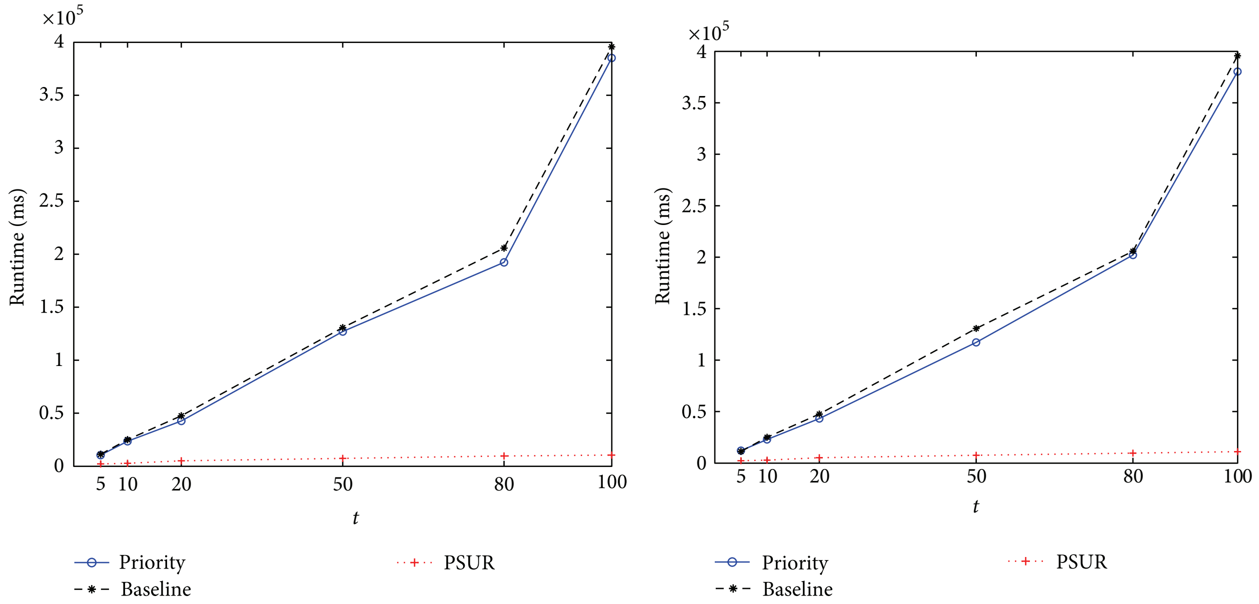
6.5. The Effect of Time Span. Baseline and priority are somewhat static algorithms. They need to recompute skyline probability for each moving object, so runtime is proportional to the time span. However, as a dynamic method, PSUR



(a) Gaussian distribution

(b) Uniform distribution

FIGURE 10: Static attribute dimensionality effects of the performance in Oldenburg city (Num = 50, Len = 20, t = 20).



(a) Gaussian distribution

(b) Uniform distribution

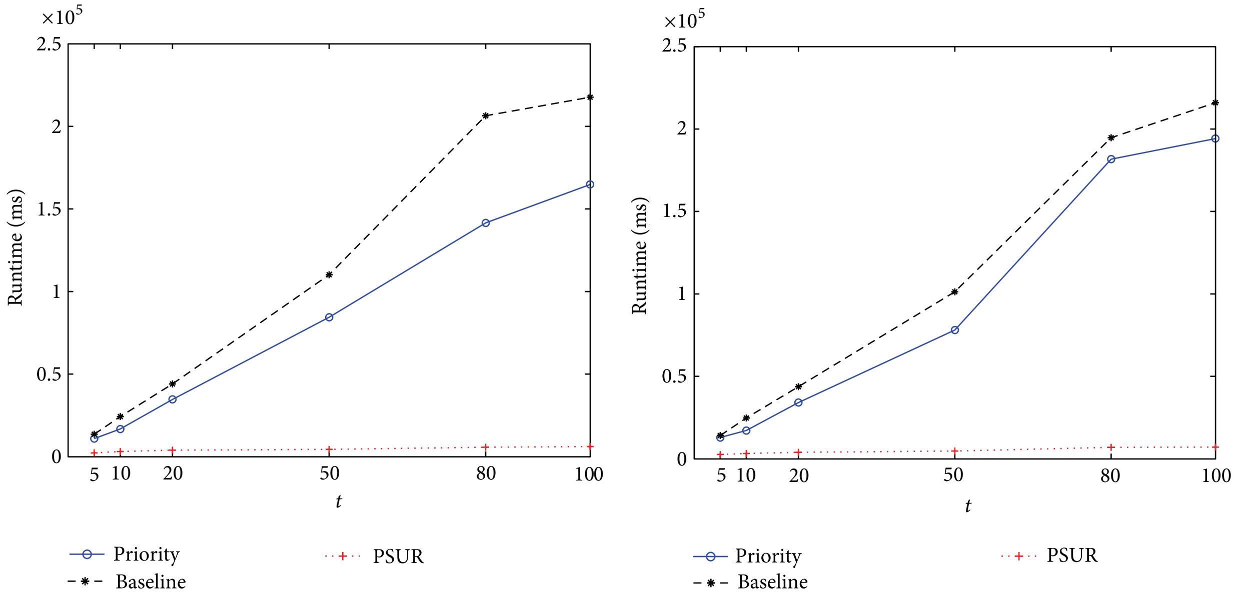
FIGURE 11: Time span effects of the performance in Cixi city (d = 5, Len = 20, Num = 100).

can update the p-skyline set by tracking event size to decide which one might vary, so the time cost is approximately the same as in each timestamp, which saves more time than other two algorithms. Figures 11 and 12 show this evidence.

Tables 8 and 9 show the event size changes with time span length of PSUR. It is obvious that the event size grows with increasing time span length. Nevertheless, the difference is not significant. The runtime of PSUR varies a little in Figures 11 and 12.

7. Conclusion

To the best of our knowledge, this is the first work to compute probabilistic skyline queries for uncertainty in road network. In this paper, we have addressed the problem of continuous probabilistic skyline query for moving objects. Firstly attributes of objects are divided into static and dynamic to define dominant probability and skyline probability with continuous data. In order to update the skyline



(a) Gaussian distribution

(b) Uniform distribution

Figure 12: Time span effects of the performance in Oldenburg city (d = 5, Len = 20, Num = 100).

Table 4: Event size in Cixi road work (d = 5; Num = 50; t = 20).

Segment length	Event size before pruning	Event size after pruning
2	1029	39
10	1597	51
20	2264	82
40	3387	113
60	4147	137
100	4450	144

Table 5: Event size in Oldenburg road work (d = 5; Num = 50; t = 20).

Segment length	Event size before pruning	Event size after pruning
2	190	2
10	245	5
20	306	6
40	461	7
60	611	7
100	890	16

Table 6: Event size in Cixi road network (Len = 20; Num = 50; t = 20).

Static dimensionality d	Event size before pruning	Event size after pruning
2	543	97
3	558	72
4	665	37
5	519	21
6	620	10
7	625	5

Table 7: Event size in Oldenburg road network (Len = 20; Num = 50; t = 20).

Segment length	Event size before pruning	Event size after pruning
2	101	41
10	118	14
20	121	7
40	124	6
60	128	4
100	122	2

set continuously, trigger events are introduced to compute varying skyline probability of objects. Two pruning ways are proposed to save search space and speed up this computation. At last, the continuous probabilistic skyline query algorithm with uncertainty in road network named PSUR is proposed. Finally, a series of experiments are devised to verify the effectiveness and efficiency of PSUR. The results of our experimental study for different scales of datasets on two real road networks are very encouraging. The investigation demonstrates that the proposed updating method is more effective and efficient than periodic methods.

Table 8: Event size in Cixi road network (Len = 20; Num = 100; $d = 5$).

Time span length	Event size before pruning	Event size after pruning
5	1029	39
10	1597	51
20	2264	82
50	3387	113
80	4147	137
100	4450	144

Table 9: Event size in Oldenburg road network (Len = 20; Num = 100; $d = 5$).

Segment length	Event size before pruning	Event size after pruning
5	359	9
10	481	13
20	583	15
50	714	20
80	830	24
100	914	28

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is partly supported by the Zhejiang Natural Science Foundation (LY12F02020), Ningbo Natural Science Foundation (2013A610063, 2012A610066), and the K. C. Wong Magna Fund in Ningbo University.

References

[1] O. Wolfson, A. P. Sistla, S. Chamberlain, and Y. Yesha, “Updating and querying databases that track mobile units,” *Distributed and Parallel Databases*, vol. 7, no. 3, pp. 257–287, 1999.

[2] G. S. Iwerks, H. Samet, and K. Smith, “Continuous k-nearest neighbor queries for continuously moving points with updates,” in *Proceedings of the International Conference on Very Large Data Bases (VLDB ’03)*, pp. 512–523, 2003.

[3] H. Mokhtar, J. Su, and O. Ibarra, “On moving object queries,” in *Proceedings of the 21st ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS ’02)*, pp. 188–198, June 2002.

[4] Z. Huang, H. Lu, B. C. Ooi, and A. K. H. Tung, “Continuous skyline queries for moving objects,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 12, pp. 1645–1658, 2006.

[5] R. Cheng, D. V. Kalashnikov, and S. Prabhakar, “Querying imprecise data in moving object environments,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 9, pp. 1112–1127, 2004.

[6] D. Pfoser and C. Jensen, “Capturing the uncertainty of moving-objects representations,” in *Proceedings of International Conference on Scientific and Statistical Database Management*, vol. 1651 of *Lecture Notes in Computer Science*, pp. 111–131, 1999.

[7] X. Ding and Y. Lu, “Indexing the imprecise positions of moving objects,” in *Proceedings of the ACM SIGMOD Ph.D. Workshop on Innovative Database Research*, pp. 45–50, Beijing, China, 2007.

[8] C. Böhm, F. Fiedler, A. Oswald, C. Plant, and B. Wackersreuther, “Probabilistic skyline queries,” in *Proceedings of the 18th ACM International Conference on Information and Knowledge Management (CIKM ’09)*, pp. 651–660, Hong Kong, November 2009.

[9] S. Borzsony, D. Kossmann, and K. Stocker, “The skyline operator,” in *Proceedings of the 17th International Conference on Data Engineering*, pp. 421–430, Heidelberg, Germany, April 2001.

[10] J. Chomicki, P. Godfrey, and J. Gryz, “Skyline with presorting: theory and optimization,” in *Proceedings of the 14th International Conference on Intelligent Information System*, pp. 593–602, 2005.

[11] D. Kossmann, F. Ramsak, and S. Rost, “Shooting stars in the sky: an online algorithm for skyline queries,” in *Proceedings of the International Conference on Very Large Data Bases (VLDB ’02)*, pp. 275–286, Hong Kong, 2002.

[12] D. Papadias, Y. Tao, G. Fu, and B. Seeger, “Progressive skyline computation in database systems,” *ACM Transactions on Database Systems*, vol. 30, no. 1, pp. 41–82, 2005.

[13] Y. Tao and D. Papadias, “Maintaining sliding window skylines on data streams,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 3, pp. 377–391, 2006.

[14] L. Tian, P. Zou, A.-P. Li, and Y. Jia, “Grid index based algorithm for continuous skyline computation,” *Chinese Journal of Computers*, vol. 31, no. 6, pp. 998–1012, 2008.

[15] M. Kontaki, A. N. Papadopoulos, and Y. Manolopoulos, “Continuous k-dominant skyline computation on multidimensional data streams,” in *Proceedings of the 23rd Annual ACM Symposium on Applied Computing (SAC ’08)*, pp. 956–960, March 2008.

[16] Y. Ishikawa, Y. Iijima, and J. X. Yu, “Spatial range querying for Gaussian-based imprecise query objects,” in *Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE ’09)*, pp. 676–687, April 2009.

[17] F. Fiedler, *Probabilistic skyline queries on uncertain data [Ph. D. dissertation]*, 2006.

[18] J. Pei, B. Jiang, X. Lin et al., “Probabilistic skylines on uncertain data,” in *Proceedings of the 33th International Conference on Very Large Databases(VLDB ’07)*, pp. 253–264, Vienna, Austria, 2007.

[19] X. Lian and L. Chen, “Monochromatic and bichromatlc reverse skyline search over uncertain databases,” in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD ’08)*, pp. 213–226, June 2008.

[20] W. Zhang, X. Lin, Y. Zhang, W. Wang, and J. X. Yu, “Probabilistic skyline operator over sliding windows,” in *Proceedings of the 25th IEEE International Conference on Data Engineering (ICDE ’09)*, pp. 1060–1071, April 2009.

[21] M. J. Atallah and Y. Qi, “Computing all skyline probabilities for uncertain data,” in *Proceedings of the 28th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems (PODS ’09)*, pp. 279–287, July 2009.

[22] X. Ding, X. Lian, L. Chen, and H. Jin, “Continuous monitoring of skylines over uncertain data streams,” *Information Sciences*, vol. 184, no. 1, pp. 196–214, 2012.

[23] X. Lin, J. Xu, and H. Hu, "Range-based skyline queries in mobile environments," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 835–849, 2013.

[24] X. Huang and C. S. Jensen, "In-route skyline querying for location-based services," in *Proceedings of the 4th International Workshop on Web and Wireless Geographical Information Systems (W2GIS '04)*, vol. 3428 of *Lecture Notes in Computer Science*, pp. 120–135, November 2004.

[25] K. Deng, X. Zhou, and H. T. Shen, "Multi-source skyline query processing in road networks," in *Proceedings of the 23rd International Conference on Data Engineering (ICDE '07)*, pp. 796–805, April 2007.

[26] S. M. Jang and J. S. Yoo, "Processing continuous skyline queries in road networks," in *Proceedings of the International Symposium on Computer Science and Its Applications (CSA '08)*, pp. 353–356, October 2008.

[27] H.-P. Kriegel, M. Renz, and M. Schubert, "Route skyline queries: a multi-preference path planning approach," in *Proceedings of the 26th IEEE International Conference on Data Engineering (ICDE '10)*, pp. 261–272, March 2010.

[28] W. Lee, C. S.-H. Eom, and T.-C. Jo, "Path skyline for moving objects," *Web Technologies and Applications*, Springer, vol. 7235, pp. 610–617, 2012.

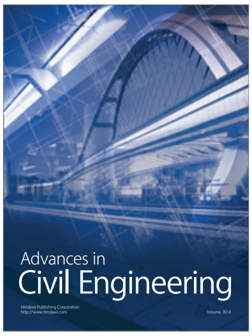
Journal of
Engineering

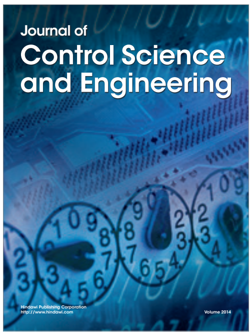
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

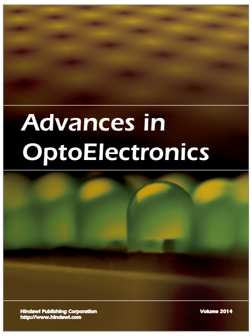
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

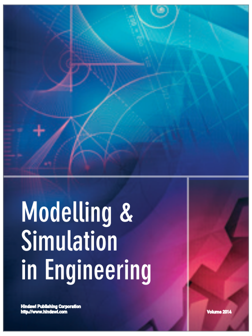
Journal of
Robotics

Journal of
Electrical and Computer Engineering

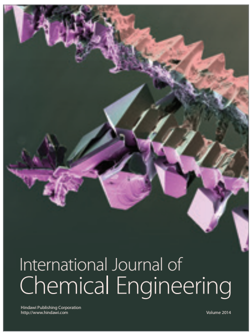

Advances in
OptoElectronics

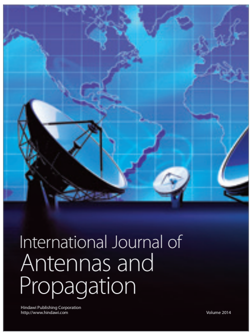
VLSI Design

International Journal of
Navigation and Observation

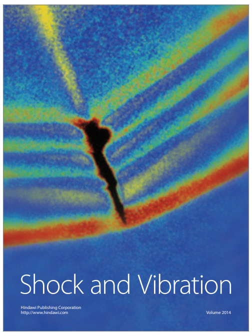

Modelling &
Simulation
in Engineering

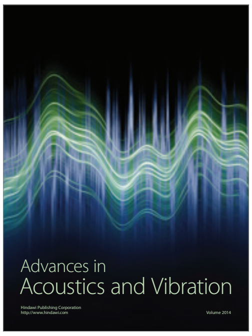
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration