Capturing Uncertainty in Spatial Queries over Imprecise Data

Xingbo Yu, Sharad Mehrotra

Shcool of Information and Computer Science, University of California
Irvine, CA 92697, USA
{xyu, sharad}@ics.uci.edu

Abstract. Emerging applications using miniature electronic devices (e.g., tracking mobile objects using sensors) generate very large amounts of highly dynamic data that poses very high overhead on databases both in terms of processing and communication costs. A promising approach to alleviate the resulting problems is to exploit the application's tolerance to bounded error in data in order to reduce the overheads. In this paper, we consider imprecise spatial data and the correlation between the data quality and precision requirements given in user queries. We first provide an approach to answer spatial range queries over imprecise data by associating a probability value with each returned object. Then, we present a novel technique to set the data precision constraints for the data collecting process, so that a probabilistic guarantee on the uncertainty in answers to user queries could be provided. The algorithms exploit the fact that objects in two-dimensional space are distributed under certain distribution function. Experimental results are also included.

1 Introduction

With the rapid development of wireless communication devices and sensor networks, wide variety of applications that require efficient access to and management of dynamic spatial-temporal data have emerged. Examples include traffic control, vehicle navigation, battle field monitoring and mobile communication analysis. Due to the dynamic nature of the data, many issues [2] arise on collection, storage and query of such data.

In many such applications, data is generated at a rapid rate(often continuously). It raises significant challenge for the database server where data is stored, as well as the communication networks through which the data flows to the server. Existing data management systems, as well as communication networks do not possess the bandwidth and capability required to sustain the data generating rate. The problem is further exacerbated when bandwidth is limited or network wireless electronics(e.g. sensors) have limited resource(e.g. power). Recent research has provided various techniques, including compression and model adaptation [3]. Deliberately accepting approximate data is another important approach to reduce data size and associated cost, if the imprecision could be tolerated by user applications posed on the collected data.

In this paper, we consider the situation in which users have specific requirements on the accuracy/certainty of the answers. The problem we address is how data precision can be set during data collection so as to meet the application requirement. Other issues related to how the spatial-temporal data can be stored at the server(see [10] and [11]) or indexing approaches to optimize query performance(see [14], [15] and [13]) are not considered.

The rest of the paper is organized as follows: In next the section, we provide a formal description on the problems we are going to address. Section 3 illustrates how uncertainty from imprecise data could be propagated to and presented in query answers. In section 4, we present our proposed algorithms on setting precision requirements to guarantee small uncertainty in answering user queries. Experimental results are presented in section 5. Section 6 concludes the paper.

2 Problem Formulation

The first problem we will discuss deals with answering range queries with imprecise cached data. Let e represent the imprecision of a object location. If we know from the database the object location at certain time instant, its real physical location may be at any point within a distance of e from that cached location. Thus, an object location could be represented graphically using a circle area(referred as "uncertain area" henceforth). In another word, a point object will be represented by an object with physical extent. The probability density distribution of the exact object location in the circle area should be decided by the data collection process. In this paper, we make the assumption that a given object is equally likely to be anywhere in its uncertain area.

Let R represent a query region. A typical range query is "return all the objects that are located in R". A $COUNT$ query is "return the number of objects that are located in R". Due to the fact that there exists uncertainty with the exact location of objects, it is impossible to provide exact answers to these queries. In section 3 we discuss how to process these queries. Here we introduce a couple of more concepts to facilitate later discussions.

$MUST$ **set:** The set of objects that "must" be located within the query range.
MAY **set:** The set of objects that "may" be located within the query range.
ANS **set:** It is the approximate answer set of objects whose cached locations are in the query region.

We further represent the number of objects in a $MUST$ set as N_s, number of objects in a MAY set as N_m, and number of objects in an ANS set as N_q.

Absolute Uncertainty δ_a**:** δ_a is the size of MAY set ($\delta_a = N_m$).
Relative Uncertainty δ_r**:** δ_r is the ratio of N_m to N_q ($\delta_r = \frac{N_m}{N_q}$).

In our second problem, we assume that there is a requirement on the degree of uncertainty that could appear in answering user queries. The task is to set

precision constraints to all location data(e.g. using a quality-aware data collection middleware), so that the specified requirement is satisfied. Formally, the problem can be stated as:

Given a requirement that the answer to a random range COUNT query on objects in a two-dimensional space should have uncertainty $\delta \leq \delta_0$ with probability $P \geq P_0$, find the largest possible imprecision value e for all location data.

δ_0 and P_0 are constants specified by users or database servers. Some times, we also refer to P_0 as the confidence level. Although in this paper we address how to solve the above problem based on $COUNT$ queries, other type of aggregate queries could be handled in a similar way. We solve this problem using probabilistic analysis in section 4.

3 Answer Range Queries over Imprecise Data

In this section, we show how to answer the two range queries described earlier. Although the process is straightforward, the format of results is important for further discussions in the later sections. Here different objects may have different imprecision e. Note although we assume a uniform probability distribution of object location across the uncertain area, any type of known probability distribution is applicable.

3.1 Return Objects in a Given Range

The task of returning objects within a given query range can be accomplished by modifying the traditional spatial query processing technique that deploys tree structures. The first modification is that the point objects are now represented by objects with non-zero extents—the uncertain area with radius e(see figure 1). To handle the fact that some objects are not fully contained in the region, another modification is needed to associate each returned object with a probability value(p_i) to indicate how likely the object can really be in the query region.

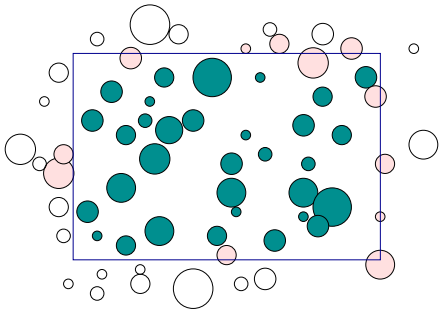

Fig. 1. Range Query over Imprecise Data (sets differentiated by colors)

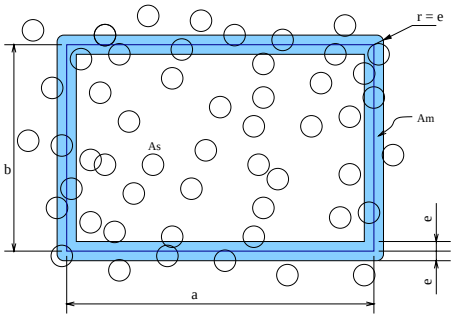

Fig. 2. Effect of Imprecision e in Range Query

Probability value p_i is 1.0 for those objects whose uncertainty area is completely contained by the given query region. For objects whose uncertainty areas overlap with query region, p_i could be computed as the cumulative probability represented by the overlapping area. Under the assumption of uniform probability distribution, we have: $p_i = \frac{A_i}{\pi e^2}$, where A_i is the overlap area that can be computed from geometric parameters.

3.2 Return *COUNT* of Objects in a Given Range

To answer $COUNT$ queries, the format of answers needs to be specified first. Possible options are $\{min, max\}$, $\{min, max, mean\}$, $\{min, max, mean, var\}$, and $\{(min, P_{min}), (min + 1, P_{min+1}), ..., (max, P_{max})\}$. Note that here we use upper case P to represent the probability that $COUNT$ takes a specific value. Lower case p is used to represent the probability with which an object could be in a query region. A server can produce all the information needed in the above answer formats, given its capability to answer the query discussed in section 3.1. Here we summarize the process to compute the above answers.

Let the first N_s objects in the answer set be the ones from $MUST$ set. Then, we know $p_i = 1$ for all $1 \leq i \leq N_s$. An immediate result is that $min = |MUST| = N_s$ and $max = |MUST| + |MAY| = N_s + N_m$. $Mean$ is summation of p_i's of all relevant objects. And $variance$ can be evaluated as:

$$variance = \sum_{k=0}^{N_s+N_m} P(COUNT = k)(k - mean)^2 \tag{1}$$

The probability for individual $COUNT$ value can be computed by summation of probabilities of events that yield the $COUNT$. For example, below is the probability that $COUNT$ takes value of $min + 1$.

$$P(COUNT = N_s + 1) = \sum_{i,j=N_s+1}^{N_s+N_m} p_i[\prod_{j \neq i}(1 - p_j)] \tag{2}$$

Among the four answer formats, the more detailed formats require more computation as well as larger answer sizes. Choosing a proper format should be a task of database server based on user requirements. In next section, we base our discussion on the second format.

4 Set Data Precision Constraints to Meet Application Requirements

As shown in the previous section, aggregate range queries can be answered in form of $\{min, max, mean\}$. For $COUNT$ query, $max - min$ is determined by information on the size of query range, object density in the area of interest, and the location data precision. Figure 2 illustrates the above factors under a typical

query scenario. Intuitively, the smaller e is, the smaller the shaded area A_m is and thus the smaller the absolute uncertainty is. In this section, we develop a precision constraint on e so that we can have a probabilistic guarantee on uncertainty. Usually, high confidence on small uncertainty is desired.

There are many factors in the real world that complicate the problem. Certain assumptions have to be made to simplify the problem. First, we assume objects are uniformly distributed on the space of interest, with density known as d. In another word, any object is equally likely to appear anywhere in a given space. In most applications, locations of moving objects may display certain pattern(e.g., many vehicles are on highway 405). However, a space could be partitioned into different areas that approximately have uniform densities. For example, density of mobile phone users in a community area can be different from that on a campus. With the partitioning, data imprecision can be set differently for different areas. For the regions that contain parts of neighboring areas, data precisions can be set with the density that can provide conservative result. It is also possible for a specific application to estimate the upper and lower bound of object densities in the application scenario and choose a conservative bound for analysis. Density can be estimated by sampling the interested area. Second, we will set the same imprecision e to all location data. Although it could be more beneficial to set different imprecision to different object location data, the problem with various data precision will become very complicated. And that remains a topic of future work. We also assume a typical range query is over a range with dimension $a \times b$. And it could be positioned anywhere in the space.

4.1 Geometric Representation and Probabilistic Properties

We can visualize the MAY set and $MUST$ set in the answer to a range query by areas in figure 2. The inner rectangle area A_s in the figure corresponds to $MUST$ set. Similarly, the shaded area A_m corresponds to MAY set. All the objects falling outside of these areas are irrelevant. These conclusions are based on the observation that any circle centered within A_m must intersect with query window and the corresponding object has non-zero probability of both being within query region and being outside the region. Also notice that ANS set is represented by the query window R. Geometric calculations yield the following results: $A_s = (a - 2e)(b - 2e)$ and $A_m = 4(a + b)e - (4 - \pi)e^2$.

Corresponding to N_s, N_m, and N_q introduced earlier, we use n_s, n_m, and n_q to denote variables (not actual outcomes) of number of objects with cached locations in A_s, A_m, and query region respectively. From probability theory [1], we know that they are Poisson variables, with means of $\lambda_s = A_s d$, $\lambda_m = A_m d$, and $\lambda_q = abd$. In the following analysis, we first make an assumption that n_m and n_q are independent variables. This is valid when the overlap(hence, correlation) between A_m and R is small. When the assumption is not valid, we use heuristic method to improve the performance.

4.2 Probabilistic Guarantee on Absolute Uncertainty

The problem to be solved here is *"Find a constraint on e, such that with probability $P \geq P_0$, a randomly positioned range($a \times b$) COUNT query will be answered with $\delta_a \leq \delta_0$."*. The desired solution should be some constraint on e, such that $P(n_m \leq \delta_0) \geq P_0$. Since n_m is a Poisson variable with mean $\lambda_m = A_m d$, its probability density function and cumulative probability distribution function are:

$$P_{n_m} = \frac{e^{-\lambda_m} \lambda_m^{n_m}}{n_m!} \tag{3}$$

$$P(n_m \leq \delta_0) = \sum_{n_m=0}^{\delta_0} \frac{e^{-\lambda_m} \lambda_m^{n_m}}{n_m!} \tag{4}$$

We observe [1] that when λ_m becomes smaller, $P(n_m \leq \delta_0)$ gets larger. This observation of monotonicity tells us that if e_0 is the value of e such that $P(n_m \leq \delta_0) = P_0$, we will guarantee $P(n_m \leq \delta_0) \geq P_0$ when $e \leq e_0$,. Now the problem becomes to find the λ_m that enables $P(n_m \leq \delta_0) = P_0$. There exist many approaches to deal with this problem, including programs implementing numerical methods or employing Poisson Cumulative Probability Table. With λ_m computed, e_0 can be obtained by solving the equation: $[4(a+b)e-(4-\pi)e^2]d = \lambda_m$.

$$e_0 = \frac{4(a+b) - \sqrt{16(a+b)^2 - 4(4-\pi)\frac{\lambda_m}{d}}}{2(4-\pi)} \tag{5}$$

We conclude that $e \leq e_0$ is the desired constraint that guarantee, with probability P_0 or higher, a random $COUNT$ query will be answered with absolute uncertainty $\delta_a \leq \delta_0$. Since we exploited the exact Poisson cumulative distribution in the above process, the e_0 so obtained is the optimal tight bound on e that satisfies the uncertainty constraint.

4.3 Probabilistic Guarantee on Relative Uncertainty

In this subsection, we will try to *"find a constraint on e, such that with probability $P \geq P_0$, a randomly positioned range($a \times b$) COUNT query will be answered with $\delta_r \leq \delta_0$."*. In a random query, δ_r is represented by $\frac{n_m}{n_q}$. The problem becomes to find a constraint on e such that $P(\frac{n_m}{n_q} \leq \delta_0) \geq P_0$. Since n_m and n_q are Poisson variables, we can develop cumulative probability function for $\frac{n_m}{n_q}$:

$$P(\frac{n_m}{n_q} \leq \delta_0) = \sum_{n_q=0}^{\infty} \sum_{n_m=0}^{\delta_0 n_q} p(n_m, n_q) \tag{6}$$

[1] One easy way to look at this is to use figures of cumulative functions with different means.

When n_m and n_q are independent, we can express $p(n_m, n_q)$ as $p(n_m)p(n_q)$:

$$P(\frac{n_m}{n_q} \leq \delta_0) = \sum_{n_q=0}^{\infty} \left(\frac{e^{-\lambda_q} {\lambda_q}^{n_q}}{n_q!} \cdot \sum_{n_m=0}^{\delta_0 n_q} \frac{e^{-\lambda_m} {\lambda_m}^{n_m}}{n_m!} \right) \tag{7}$$

Since n_q goes up to infinity, it is impossible to evaluate for exact cumulative probability $P(\frac{n_m}{n_q} \leq \delta_0)$. But we can approximate the value by observing the fact that when n_q becomes large, the element to be summed in the outer summation becomes very small. It is smaller than Poisson probability at value n_q. Obviously, the larger the upper bound of n_q, the more accurate the approximation is. Also note that the standard deviation of Poisson variable is square root of its mean $\sigma = \sqrt{\lambda}$. Then using $c\lambda_q$ as upper bound, where c is a constant larger than 1, should yield a good approximation that is *smaller* than exact value.

$$P(\frac{n_m}{n_q} \leq \delta_0) \cong \sum_{n_q=0}^{c\lambda_q} \left(\frac{e^{-\lambda_q} {\lambda_q}^{n_q}}{n_s!} \cdot \sum_{n_m=0}^{\delta_0 n_q} \frac{e^{-\lambda_m} {\lambda_m}^{n_m}}{n_m!} \right) \tag{8}$$

With this approximation, we can find λ_m that will guarantee $P(\frac{n_m}{n_q} \leq \delta_0) \geq P_0$. And the corresponding $e = e_0$ can be evaluated using formula 5. Again, numerical method should be deployed for finding $\lambda_m(e_0)$. In our experiment, we search for $\lambda_m(e_0)$ starting from $\lambda_m = \delta_0 \lambda_q$ when $P_0 \geq 0.5$. This is enabled by the fact that the cumulative probability function is a monotonic function of λ_m(hence, e_0).

We have so far showed how to set imprecision constraints on location measurements so that a probabilistic guarantee could be provided on the answers to random user queries. Between the two uncertainties we have defined, since approximation is applied and correlation plays a role in developing bound with relative uncertainty, the bound is not as tight as the one for absolute uncertainty. On the other hand, relative uncertainty guarantee could still be preferred by users, since people tend to have a percentage concept in mind.

5 Empirical Evaluation

In this section, we study the performance of the proposed algorithms for providing probabilistic guarantees through simulation. We compare the simulation results of probability that a random query will be answered under given uncertainty constraints with desired confidence values.

Data: In the simulation, we set the space of interest to be 100×100 in two dimensions. The unit is not specified and it could vary from application to application. Queries are randomly positioned in the 100×100 space and they are all in same size—10×10. However, we change the object density(which is uniform) from 0.05/square unit to 10/square unit. From statistics point of view, changing query window size and changing object density have the same effect on theoretical analysis, since both result in a change in the mean value.

Experiments on Various Densities: Our first experiment is done by varying the density, while fixing the uncertainties at 10 or 10% and fixing the confidence requirement at $P_0 = 0.90$, and computing tolerated imprecision value e. Then simulations are conducted to count the numbers of returned objects in different sets, N_m and N_q, for each randomly generated query. The number of trials is set to be 10000. This simulation gives us insight about how the algorithms behave with different densities(thus, mean λ_q). Figure 3 shows the computed

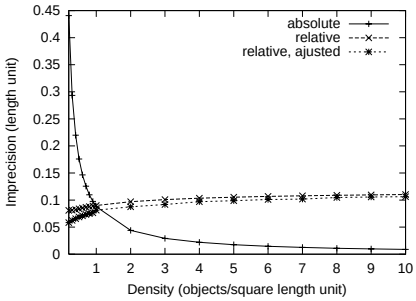

Fig. 3. Imprecision Values

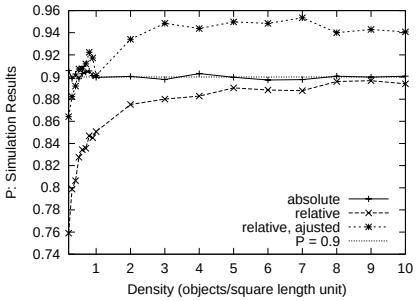

Fig. 4. Simulation Results

imprecision values and figure 4 shows the simulation results. It is easy to see that the algorithm performs extremely well for absolute uncertainty constraint. But for relative uncertainty constraint, the result is good only for large density values and deteriorate when the density becomes smaller and smaller. This can be understood, however, with a re_examination of the independence assumption between n_m and n_q. The assumption is valid only when the overlap between MAY region and query region R is small and there are large number of objects in query region. But when λ_q is small, the correlation between n_m and n_q becomes too significant to be ignored. We adopt heuristic methods to improve the performance. The curve "relative, adjusted" in above figures show the effect of an adjustment, which is to use $P_0 \geq 0.95$ for constraint requirement $P_0 \geq 0.90$. After this adjustment, for most of the densities larger than 0.5/square unit yield good results. Experimental results show that a very small decrease in e will produce significant improvement on the confidence level. Although the results from adjustments are not optimal, they are very close to optimal (which should be a curve between the two "relative" curves) as shown in figure 3.

Experiments on Different Confidence Levels: Figure 5 and figure 6 show us empirical results on different confidence levels ranging from 0.80 to 1 with fixed uncertainties ($\delta_a = 10$ or $\delta_r = 10\%$) and density ($d_0 = 1$). Again, results for absolute uncertainty are very good. The adjusted experiments are done by increasing the confidence level by 5% percent for P_0 values below 0.90 and increasing less(up to $0.96, 0.97, 0.98, 0.99, 0.995, 1$, respectively) for other confidence values. And the results are conservative after the adjustments.

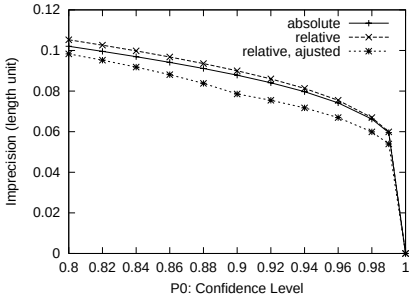

Fig. 5. Imprecision Values

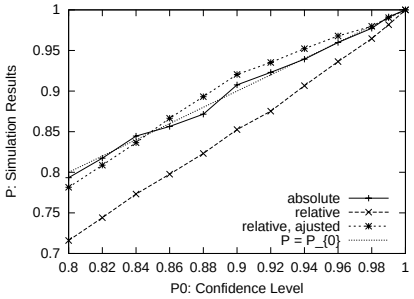

Fig. 6. Simulation Results

6 Related Work

The authors are not aware of any work on calibrating data for probabilistic uncertainty guarantees in the domain of spatial range query. There are two closely related areas, uncertainty management in information system and quality-aware data collection, where substantial amount of research work has been conducted.

Uncertainty management in information system([5], [4]) deals with the uncertainty management and reasoning in an information system. The work in section 3 of this paper can be viewed as a specific example of uncertainty management in the domain of spatial range queries using probabilistic method. As a contrast, [12] handled uncertainty in spatial database domain using method of fuzziness. But the major contribution of this paper is on how to control the introduction of uncertainty into an information system, which has not been addressed in the area.

Quality-aware data collection is the most closely related work to this paper. In this area, quality/precision requirements of query answers and quality in raw data are connected, with the ultimate goal of satisfying query requirements while minimize certain cost. For example, Yu et al. [6] addressed network monitoring problems in which certain aggregate values in a network are approximated. Olston et al. [9] provided an adaptive data collection protocol to collect data with certain precision requirements so that the application quality requirements of a set of continuous queries could be met and the total communication cost for collecting the data is minimized. However, most of the works in this area have been concentrated on deterministic guarantees of user requirements. Usually in the context of real time applications, more accurate data is available with more cost. In contrast, in our problem, queries are on data collected before and thus the precisions have been fixed.

There are also some other uncertainty-related works in literature. Schneider [7] introduced fuzziness into the modeling of spatial data. Pfoser et al. [8] addressed a problem in which the uncertainty is caused by low sampling rate. These works are all different from our approach or our problem setting.

7 Conclusion

In this paper, we first described a query processing technique for aggregate range queries over imprecise data. Then we presented algorithms to set precision constraints on spatial data collection process to meet uncertainty constraints when the data are used to answer user queries. The guarantees are probabilistic and are discussed under the scenarios of using either absolute uncertainty or relative uncertainty. Both theoretical analysis and experimental results showed that the probabilistic guarantee based on absolute uncertainty yields a tighter bound which enables larger tolerated data imprecision. And simple adjustments on relative uncertainty method can be applied to improve its performance.

Acknowledgments. Our work was supported by the National Science Foundation (Awards IIS-9996140, IIS-0086124, CCR-0220069, IIS-0083489) and by the United States Air Force (Award F33615-01-C-1902).

References

1. S. Ross. A First Course in Probability. *Prentice Hall*, 2002.
2. O. Wolfson, B. Xu, S. Chamberlain, L. Jiang. Moving Objects Databases: Issues and Solutions. *SSDBM* 1998: 111-122.
3. I. Lazaridis, S. Mehrotra. Capturing Sensor-Generated Time Series with Quality Guarantees. *International Conference on Data Engineering (ICDE)*, March, 2003.
4. F. Sadri. Modeling Uncertainty in Databases. *Proceedings of the Seventh IEEE International Conference on Data Engineering*, April, 1991.
5. C. E. Dyreson. A Bibliography on Uncertainty Management in Information Systems. *Uncertainty Management in Information Systems: From Needs to Solutions, Kluwer Academic Publishers*, 1997: 415-458.
6. H. Yu, A. Vahdat. Efficient Numerical Error Bounding for Replicated Network Services. *VLDB* 2000: 123-133.
7. M. Schneider. Uncertainty Management for Spatial Data in Databases: Fuzzy Spatial Data Types. *SSD* 1999: 330-351.
8. D. Pfoser, Christian S. Jensen. Capturing the Uncertainty of Moving-Object Representations. *SSD* 1999: 111-132.
9. C. Olston, J. Jiang, and J. Widom. Adaptive Filters for Continuous Queries over Distributed Data Streams. *To appear: SIGMOD* 2003.
10. M. Erwig, R. Hartmut, M. Schneider, M. Vazirgiannis. Spatio-Temporal Data Types: An Approach to Modeling and Querying Moving Objects in Databases. *GeoInformatica 3(3)*, 1999: 269-296.
11. M. Vazirgiannis, O. Wolfson. A Spatiotemporal Model and Language for Moving Objects on Road Networks. *SSTD* 2001: 20-35.
12. M. Vazirgiannis. Uncertainty Handling in Spatial Relationships. *SAC (1)* 2000: 494-500.
13. S. Saltenis, C. S. Jensen, S. T. Leutenegger, M. A. Lopez. Indexing the Positions of Continuously Moving Objects. *SIGMOD* 2000: 331-342.
14. Y. Theodoridis, T. K. Sellis, A. Papadopoulos, Y. Manolopoulos. Specifications for Efficient Indexing in Spatiotemporal Databases. *SSDBM* 1998: 123-132.
15. G. Kollios, D. Gunopulos, V. J. Tsotras. On Indexing Mobile Objects. *PODS* 1999: 261-272.