Spatial Query Processing Based on Uncertain Location Information

Yoshiharu Ishikawa

Information Technology Center, Nagoya University
ishikawa@itc.nagoya-u.ac.jp
http://www.itc.nagoya-u.ac.jp/~ishikawa

Abstract. Location information acquired by sensors and other devices is not necessarily accurate and has vagueness. In the research field of spatial databases, query processing techniques based on uncertain location information are highly interested in recent years. In this paper, we overview the trend of this field and describe our related projects and future prospects.

1 Introduction

In recent years, due to the development of mobile network technology and GPS devices, information services and applications based on location information are repidly increasing. Location information acquired by GPS usually contains noise and we may not be able to obtain accurate location information. Uncertainty of location occurs not only in mobile computing but also in a location estimation of a mobile robot. In robotics, the location of a mobile robot is often estimated by sensors and movement histories [1]. Based on such a background, research on spatial database queries based on uncertain location information has become an active research field.

Our research group has been investigating query processing techniques for spatial databases based on uncertain information [2,3]. The feature is to use the *Gaussian distribution* to express location vagueness. Gaussian distribution is one of the basic probability distributions and widely used in statistics and pattern recognition [4]. Our research mainly focuses on the situation when the user's position obeys a Gaussian distribution and we have proposed algorithms for range queries [3] and nearest neighbor queries [2]. In this paper, we overview related studies in spatial query processing for uncertain location information then briefly introduce the research activities of our group.

2 Overview of Related Work

There are various ways for representing uncertainty of locations. The simplest approach is to assume the location of an object obeys a *uniform distribution* (e.g., [5]). Another approach is to assume a location is generally described by some *probabilistic density function* (*PDF*). The assumed types of PDFs are different

S. Kikuchi, S. Sachdeva, and S. Bhalla (Eds.): DNIS 2010, LNCS 5999, pp. 311–316, 2010.

depending on the studies. Most of the approaches [6,7,8] treat a PDF as a black box and do not consider its details. Their approaches aim at generic algorithms which work well for arbitrary PDFs. On the other hand, some other approaches consider special types of PDFs [2,3]. Using specific properties of a PDF, we would be able to develop efficient and effective algorithms.

We can classify types of spatial queries. There are various types of queries in spatial databases [9]. We can also consider several types of queries for uncertain location information. Most of the proposals focus on the algorithms for range queries [7,10,3,5,8] and nearest neighbor queries [6,11,10,12]. Some studies focus on clustering methods based on uncertain location information [13].

We can consider other criteria by considering which objects are uncertain. Roughly speaking, there are three cases: data objects are uncertain [11,10,5,8], query objects are uncertain [3], and both of the objects are uncertain [6,7,12]. In addition, the proposed algorithms are different in several factors such as the types of object shapes (e.g., points, rectangles) and the number of dimensions (e.g., 1-D, 2-D, arbitrary dimensions).

The concept of an *uncertainty region* is often used in the algorithms for spatial querying based on uncertain location information. An uncertainty region for an object is a region such that the region in which the object is located with the specified probability. It roughly represents the area in which the object locates and used for fast processing when a spatial query is given. An uncertainty region may be contained in the description of the object itself, or it may be derived from the properties of the object. The latter approach is often taken when it is costly to treat the vague representation of an object directly.

When the location of an object is given by a PDF, integration of the PDF is often required to process queries. If the PDF cannot be integrated analytically, it is necessary to use numerical integration such as the Monte Carlo method, but the processing cost is quite huge. In this case, a filtering process to decrease the number of target objects for integration becomes extremely important, and the concept of an uncertainty region is especially effective.

In addition to these approaches, some studies incorporated sampling methods into query processing algorithms [12]. Moreover, [14] extended the notion of the *Voronoi diagram* [15] for uncertain locations to process nearest neighbor queries.

3 Summary of Our Research

3.1 Uncertain Query Locations

In our research, we assume that the Gaussian distribution is used for representing uncertainty of locations. In the previous studies [2,3], we assume that the location of a query object is imprecisely specified by a Gaussian distribution. The location of a query object is formally defined as follows.

Definition 1. *Assume that $\boldsymbol{x}$, the location of a query object q, is represented by a d-dimensional Gaussian distribution* [4]

$$p_q(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{q})^t\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{q})\right], \tag{1}$$

where $\boldsymbol{q}$ *is the average of the distribution,* $\boldsymbol{\Sigma}$ *is a* $d \times d$ *covariance matrix, and* $|\boldsymbol{\Sigma}|$ *represents its determinant.*

As described in the introduction part, the Gaussian distribution is quite popular in statistics and pattern recognition [4] and is also used in estimating the location of moving robots [1]. We have developed efficient spatial query processing algorithms considering the specific properties of a Gaussian distribution.

3.2 Probabilistic Range Queries

As an example, let us consider a neighborhood information retrieval for a moving robot. We assume that a moving robot sequentially estimates its location based on sensor information and own movement histories (Fig. 1). If we represent and track the movement by using a Kalman filter [1] assuming a Gaussian probabilistic process, the location of the robot in every moment is be vaguely represented by a Gaussian distribution.

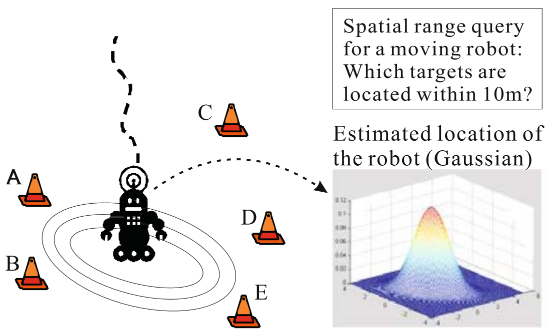

Fig. 1. Spatial range query issued by a moving robot

Consider that we want to retrieve the information of the target objects (e.g., obstacles or items to be obtained) within 10m from the robot. We assume that the data for the target objects (their locations are represented by points) are maintained in the database embedded within the robot. This query cannot be processed as a normal spatial range query—since the location of the robot is uncertain, the distance between the robot and each target is also uncertain and is given probabilistically.

To solve this problem, we have extended the concept of a spatial range query [3]. We defined a *probabilistic range query* (PRQ) such as "retrieve the target objects such that the probabilities such that the locations of the objects are within 10m from the moving robot are less than the threshold". It is formally given as follows.

Definition 2. *Given the probability density function* $p_q(\boldsymbol{x})$*, the distance threshold* δ $(\delta > 0)$*, and the probability threshold* θ $(0 < \theta < 1)$*, a* probabilistic range *query $PRQ(q, \delta, \theta)$ returns all the objects such that the probabilities that their distances from the query object q are less than or equal to δ are greater than or equal to θ. It is formally defined as follows:*

$$PRQ(q, \delta, \theta) = \{o \mid o \in \mathcal{O}, \Pr(\|\boldsymbol{x} - \boldsymbol{o}\|^2 \leq \delta^2) \geq \theta\}, \tag{2}$$

where $\mathcal{O}$ is the set of the target objects, $\|\cdot\|$ is the length of a vector, and $\|\boldsymbol{x}-\boldsymbol{o}\|^2$ represents the squared Euclidean distance between $\boldsymbol{x}$, the location of the query object q, and $\boldsymbol{o}$, the location of the object o.

The naive approach to process this query is as follows. For each target object, we consider a circle which is centered at the target point and has the radius δ. If the probability that the query object (moving robot) locates inside of the circle is greater than or equal to θ, the target object satisfies the query condition. However, we need to integrate the Gaussian distribution shown in Eq. (1) for evaluating the probability—this is quite costly because it requires numerical integration such as the Monte Carlo method.

To solve the problem, we proposed three query processing strategies to reduce the number of target objects for which numerical integration should be performed [3]. Using three strategies, we first derive three regions in which the candidate target objects are located. Then we retrieve the objects (the targets of numerical integration) which is inside of the intersection area of the three regions. In addition, the proposed algorithm can effectively use a spatial index such as an R-tree.

Let us show an experimental result. Figure 2 shows the points data obtained from the cross roads of Montgomery County of Maryland, U.S.A. We have normalized the data within 1000×1000 units. The figure shows an example query when $\delta = 50$ and $\theta = 1\%$. The ellipse shown in the figure represents the iso-surface of the Gaussian distribution. Figure 3 shows the resulting points for this query.

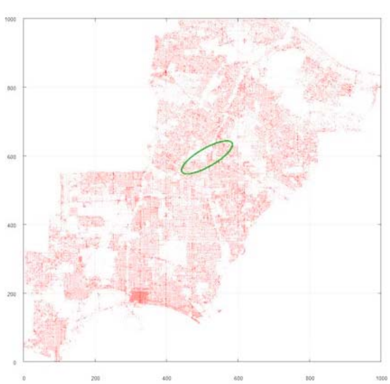

Fig. 2. Example data and query

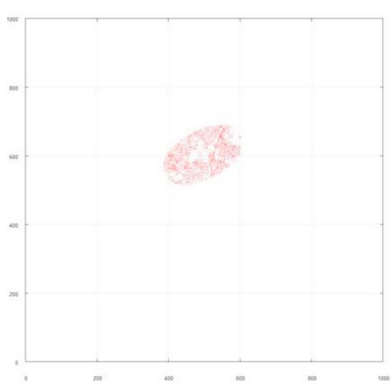

Fig. 3. Example query result

3.3 Probabilistic Nearest Neighbor Queries

We also proposed the extended version of nearest neighbor queries for our context [2]. A *probabilistic nearest neighbor query* (*PNNQ*) is defined as follows.

Definition 3. *Given the probability density function* $p_q(\boldsymbol{x})$ *and the probability threshold* θ $(0 < \theta < 1)$*, a* probabilistic nearest neighbor query $PNNQ(q, \theta)$ *returns all the objects each of which satisfy the following condition: the probability that the object becomes the nearest neighbor of* q *is greater than or equal to* θ*. Let* $\mathcal{O}$ *be the set of data objects and let* $\mathrm{Pr}_{NN}(o)$ *be the probability that an object* $o \in \mathcal{O}$ *becomes the nearest neighbor of* q*:*

$$\mathrm{Pr}_{NN}(q, o) = \mathrm{Pr}(\forall o' \in \mathcal{O}, o' \neq o, \|\boldsymbol{x} - \boldsymbol{o}\|^2 \leq \|\boldsymbol{x} - \boldsymbol{o}'\|^2), \tag{3}$$

A probabilistic nearest neighbor query *is defined as follows:*

$$PNNQ(q, \theta) = \{n \mid n \in \mathcal{O}, \mathrm{Pr}_{NN}(q, n) \geq \theta\}. \tag{4}$$

We have proposed two query processing strategies both of which are based on the Voronoi diagram [15]. Figure 4 shows the result of a query. The data set is same as the former example. In the proposed method, the candidate objects (the targets of numerical integration) correspond to the intersection of the Voronoi regions which overlap with the rectangle shown in the figure and the Voronoi regions with bold borders. The shaded Voronoi regions correspond to the resulting nearest neighbor objects.

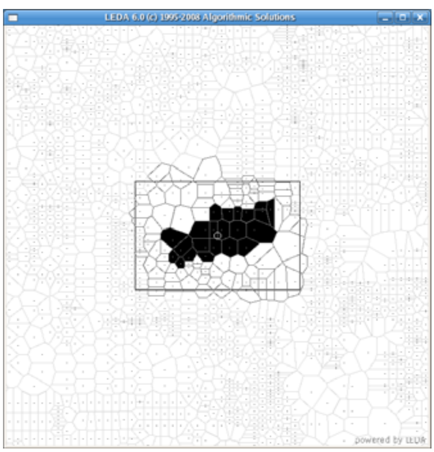

Fig. 4. Result for probabilistic nearest neighbor queries

4 Conclusions

In this paper, we described the approaches to spatial database query processing based on uncertain location information. We introduced the background and several existing studies and then described the research performed by our group. Currently, our group is working for extending the previous work. For example, we are developing query processing method when both of a query object and target objects obey Gaussian distributions and a general indexing method for processing location-based queries based on Gaussian-based distributions. We would like to report their results in the future publications.

Acknowledgments

This research is partly supported by the Grant-in-Aid for Scientific Research on Priority Areas, Japan (21013023).

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press, Cambridge (2005)
2. Iijima, Y., Ishikawa, Y.: Finding probabilistic nearest neighbors for query objects with imprecise locations. In: Proc. MDM, pp. 52–61 (2009)
3. Ishikawa, Y., Iijima, Y., Yu, J.X.: Processing spatial range queries for objects with imprecise Gaussian-based location information. In: Proc. ICDE, pp. 676–687 (2009)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. Wiley, Chichester (2000)
5. Pfoser, D., Jensen, C.S.: Capturing the uncertainty of moving-object representations. In: Güting, R.H., Papadias, D., Lochovsky, F.H. (eds.) SSD 1999. LNCS, vol. 1651, pp. 111–131. Springer, Heidelberg (1999)
6. Beskales, G., Soliman, M.A., Ilyas, I.F.: Efficient search for the top-k probable nearest neighbors in uncertain databases. In: Proc. VLDB, pp. 326–339 (2008)
7. Chen, J., Cheng, R.: Efficient evaluation of imprecise location-dependent queries. In: Proc. ICDE, pp. 586–595 (2007)
8. Tao, Y., Xiao, X., Cheng, R.: Range search on multidimensional uncertain data. ACM TODS 32(3) (2007)
9. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases with Application to GIS. Morgan Kaufmann, San Francisco (2001)
10. Cheng, R., Kalashnikov, D.V., Prabhakar, S.: Querying imprecise data in moving object environments. IEEE TKDE 16(9), 1112–1127 (2004)
11. Cheng, R., Chen, J., Mokbel, M., Chow, C.Y.: Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. In: Proc. ICDE, pp. 973–982 (2008)
12. Kriegel, H.P., Kunath, P., Renz, M.: Probabilistic nearest-neighbor query on uncertain objects. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 337–348. Springer, Heidelberg (2007)
13. Ngai, W.K., Kao, B., Chui, C.K., Cheng, R., Chau, M., Yip, K.Y.: Efficient clustering of uncertain data. In: Proc. ICDM (2006)
14. Cheng, R., Xie, X., Yiu, M.L., Chen, J., Sun, L.: UV-diagram: A Voronoi diagram for uncertain data. In: Proc. ICDE (2010)
15. Aurenhammer, F.: Voronoi diagrams: A survey of a fundamental geometric data structure. ACM Computing Surveys 23(3), 345–405 (1991)