# Integrated Security Solution for Moving Object Tracking System

**Anand Nayyar**
*Assistant Professor, Department of Computer Applications & IT,*
*KCL Institute of Management and Technology, Jalandhar*
*anand_nayyar@yahoo.co.in*

**Abstract:** *Tracking the moving objects in high traffic environment is really a complex task. For location tracking system Spatial Data values are used. The information regarding the location of the objects is maintained by Database Server. To process Spatial Query values the application server is used. Location update is the highly bandwidth and resource constrained operation in the moving object tracking system. Location update is initiated and constantly refreshed with respect to the spatial query values. In this Research Paper a system is designed with integrated security and privacy for the moving object tracking systems. Location privacy is added using region based location update technique. Authentication is provided using regional data centers. In order to ensure confidentiality RSA (Rivest, Shamir and Adelman) algorithm; the secure hash algorithm is used for the integrity analysis process. The system also enhances the query processing schemes with improved spatial query execution mechanisms.*

**Index Terms:** Object Tracking, PAM, Security, Query Processing

## I. Introduction

In Mobile and Spatiotemporal databases, monitoring continuous spatial queries over moving objects is required in various applications like Public Transportation, Logistics and Location Based Services. Diagram 1 demonstrates typical monitoring system which comprises of a base station, a database server, application servers and large number of moving objects i.e. Mobile Clients. The Database Server manages the location information of the objects. The Application Server captures monitoring requests and register spatial queries at the Database Server which in turn automatically updates the query results till the queries are deregistered.

Monitoring System determines three principles performance measures of monitoring i.e. Accuracy, Efficiency and Privacy. Accuracy determines to which extent the monitored results are correct and this factor heavily depends on the frequency and accuracy of location updates. In terms of Efficiency, there exist two dominant costs: Wireless Communication Cost for location updates and Query Evaluation cost for Database Server; these both dominant costs depend on the frequency of location updates. And for Privacy, the accuracy of location updates determines to what extent the client's privacy is exposed to the Server.

Recently some efforts were attempted to overcome the privacy issue and a technique called Location Cloaking was proposed to blur the exact client positions into bounding boxes by assuming a third party centralized server that store all exact client positions. Various location cloaking algorithms were proposed to build the bounding boxes while achieving the privacy measure like K-Anonymity. But the results were not at all unique. A common approach was there to assume that the probability distribution of the exact client location in the bounding box is known and well formed. Therefore, the results are defined as the set of all possible results together with their probabilities. However, all these approaches focused on one-time cloaking or query evaluation; they cannot be applied to monitoring applications where continuous location update is required and efficiency is a critical concern.


**Figure 1: System Architecture of Traditional Tracking System**

In order to optimize privacy or efficiency issues, some alternative strategies were deployed. As compared to previous work, the new solution called PAM framework has the following advantages:-

1. In terms of Efficiency, the PAM Framework significantly reduces location updates to only when an object is moving out of the safe region and in turn, very likely to alter the query results.
2. In terms of accuracy, the Framework offers correct monitoring results at any time, as opposed to only at the time instances of updates in systems that are based on periodic or deviation location update.
3. The PAM frame work is called generic in terms that it is not designed for a specific query type but is provides a common interface for monitoring various types of spatial queries like range queries and kNN queries. In addition to this, the PAM framework doesn't presume any mobility pattern on moving objects.

4. The framework is flexible in that by designing appropriate location update strategies, accuracy, privacy, or efficiency can be optimized.

## II. Related Work

There is a large body of research work on spatial temporal query processing. Early work assumed a static data set and focused on efficient access methods and query evaluation algorithms. Recently, a lot of attention has been paid to moving-object databases, where data objects or queries or both of them move.

The work on monitoring continuous spatial queries can be classified into two categories. The first category assumes that the movement trajectories are known. Continuous kNN monitoring has been investigated for moving queries over stationary objects and linearly moving objects. Iwerks et al. extended to monitor distance semi joins for two linearly moving data sets. However, as pointed out in, the known-trajectory assumption does not hold for many application scenarios

(e.g., the velocity of a car changes frequently on road).

Uncertainty and privacy issues have been recently studied in moving object monitoring. To protect location privacy, various cloaking or anonymizing techniques have been proposed to hide the client's actual location. Among them are the spatiotemporal cloaking, the Clique- loak, the Casper anonymizer, hilbASR, and peer-to-peer cloaking. In spatiotemporal cloaking, for each location update, the server divides the space recursively in a quad-tree-like format till a suitable subspace is found to cloak the updated location. The Clique Cloak algorithm constructs a clique graph to combine some clients who can share the same cloaked spatial area. The Casper anonymizer is associated with a query processor to ensure that the anonymized area returns the same query result as the actual location. In hilbASR, all user locations are sorted by Hilbert space-filling curve ordering, and then, every k users are grouped together in this order. Besides, location cloaking, pseudonym, dummy, and transformation were also proposed for privacy preservation. Pseudonym decouples the mapping between the user identity and the location so that an untrusted server only receives the location without the user identity. Dummy generates fake user locations (called dummies) and mixes them together with the genuine user location into the request. Transformation utilizes certain one-way spatial transformations (e.g., a space filling curve) to map the query space to another space and resolves query blindly in the transformed space.

As for location uncertainty, a common model for characterizing the uncertainty of an object is a closed region with a predefined probability distribution of this object in the region. Based on this probabilistic model, query processing and indexing algorithms have been proposed to evaluate probabilistic range queries and kNN queries. While in these studies, By adopting the notion of "safe region," the frequency of query reevaluation on uncertain location information is reduced, and hence, the system efficiency and scalability are improved.

Distributed approaches have been investigated to monitor continuous range queries and continuous kNN queries. The main idea is to shift some load from the server to the mobile clients. Monitoring queries have also been studied for distributed Internet databases, data streams, and sensor databases. However, these studies are not applicable to monitoring of moving objects, where a two-dimensional space is assumed.

## III. Overview of Pam Framework

As shown in Diagram 2, the PAM framework consists of components located at both the database server and the moving objects. At the database server side, we have the moving object index, the query index, the query processor, and the location manager. At moving objects' side, we have location updaters. Without loss of generality, we make the following assumptions for simplicity:

1. The database server handles location updates sequentially; in other words, updates are queued and handled on a first-come-first-serve basis. This is a reasonable assumption to relieve us from the issues of read/write consistency.

2. The moving objects maintain good connection with the database server. Furthermore, the communication cost for any location update is a constant. With the latter assumption, minimizing the cost of location updates is equivalent to minimizing the total number of updates

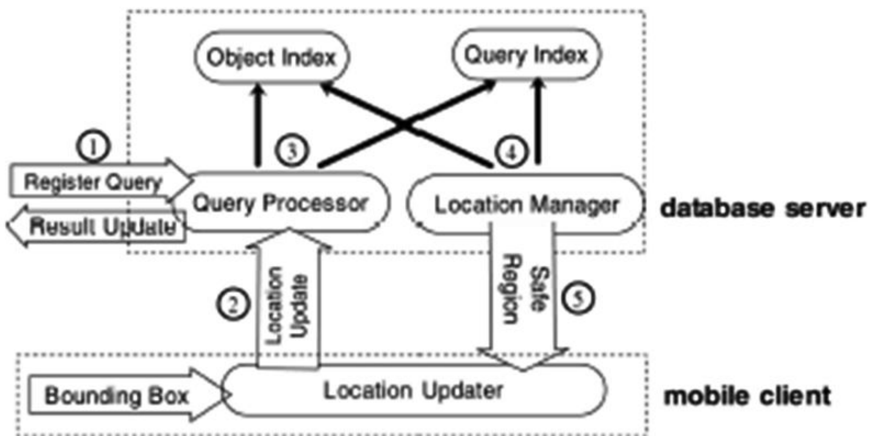


**Diagram 2: Pam Framework Overview**

PAM framework works as follows (see Fig. 2): At any time, application servers can register spatial queries to the database server (step 1). When an object sends a location update (step 2), the query processor identifies those queries that are affected by this update using the query index, and then, reevaluates them using the object index (step 3). The updated query results are then reported to the application servers who register these queries. Afterward, the location manager computes the new safe region for the updating object (step 4), also based on the indexes, and then, sends it back as a response to the object (step 5). The procedure for processing a new query is similar, except that in step 2, the new query is evaluated from scratch instead of being reevaluated incrementally, and that the objects whose safe regions are changed due to this new query must be notified.

Algorithm 1 summarizes the procedure at the database server to handle a query registration/deregistration or a location update is possible. It is noteworthy that although in this paper, the most probable result is used, this framework can also adapt to other query result definitions such as over a probability confidence (e.g., “returns objects that have 90 percent probability inside the query range”). The only changes needed to reflect the new result definition are the query evaluation algorithms in the query processor and safe region computation in the location manager. In the rest of this paper, we stick to the definition of the most probable result and leave the modification details for other definitions to interested readers.

## IV. Query Processing

In this section, we present the detailed algorithms to evaluate or reevaluate a spatial query Q in terms of the most probable result. Aside from the definition of the query result, we know that Q also differs from a conventional spatial query in that the object locations are in the form of -square (for updating objects) or bbox (for other objects), both of which are rectangular. Instead of regarding Q as a special query type, we take an alternative approach by regarding the space where the object locations are defined as a special euclidean space. In this space, spatial relations such as overlapping, containment, or even distance are implemented differently from a conventional euclidean space. By using the new implementations of spatial relations, existing spatial query processing algorithms can be applied directly to the new space. In the following sections, we implement two relations that are required for spatial queries, namely, containment and closer.

**Algorithm 1:** Evaluating a new kNN Query

**Input:** root: root node of object index q: the query point

**Output:** C: the set of kNNs

**Procedure:**

1: initialize queue H and H;
2: enqueue {root, d(q; root)} into H;

3: **while** |C| < k and H is not empty **do**

4: u = H.pop ();

5: **if** u is a leaf entry **then**

6: **while** d(q, u) > D(q, v) **do**

7: v = H.pop ();

8: insert v to C;

9: enqueue u into H;

10: **else if** u is an index entry **then**

11: **for** each child entry v of u **do**

12: enqueue hv; dðv; qpi into H;

To reevaluate an existing kNN query that is affected by the updating object p, the first step is to decide whether p is a result object by comparing p with the kth NN using the "closer" relation: if p is closer, then it is a result object; otherwise, it is a nonresult object. This then leads to three cases: 1) case 1: p was a result object but is no longer so; 2) case 2: p was not a result object but becomes one; and 3) case 3: p is and was a result object.4 For case 1, there are fewer than k result objects, so there should be an additional step of evaluating a 1NN query at the same query point to find a new result object u.

**Algorithm 2:** Reevaluating a kNN Query

**Input:** C: existing set of kNNs p: the updating object

**Output:** C: the new set of kNNs

**Procedure:**

1: **if** p is closer to the k-th NN then

2: **if** p C then

3: p* = the rank of p in C;

4: **else**

5: p*= k;

6: enqueue p into C;

7: **else**

8: if p C then

9: evaluate 1NN query to find u;

10: p*= k;

11: remove p and enqueue u into C;

12: relocate p or u in C, starting from p*;

## V. Safe Region Computations

The location manager computes the optimal safe region for an individual query Q, which is the inscribed rectangle with the longest perimeter (Ir - lp) of Q's inner or outer region, separated by the quarantine line. Therefore, the safe region is obtained in two steps: finding the quarantine line, and then, finding the Ir - lp. It is noteworthy that the safe region must contain the updating object p (i.e., its centroid), because otherwise, this object has to send an immediate location update after it receives this safe region. In this section, we present the detailed algorithms to compute the quarantine line, and hence, the safe region for various types of queries.

## VI. Integrated Security Mechanism for Object Tracking

The object location tracking system is enhanced with the security and privacy mechanism. The location updates are secured with authentication, confidentiality and integrity mechanisms. The regional data centers are used for the authentication process. Confidentiality is used to secure the location update data values. The RSA algorithm is used to secure the data. The secure hashing algorithm is used to provide message integrity to the system.

## VII. Conclusion

This paper proposes a framework for monitoring continuous spatial queries over moving objects. The framework is the first to holistically address the issue of location updating with regard to monitoring accuracy, efficiency, and privacy. We provide detailed algorithms for query evaluation/ reevaluation and safe region computation in this framework. We also devise three-client update strategies that optimize accuracy, privacy, and efficiency, respectively. The performance of our framework is evaluated through a series of experiments. The results show that it substantially outperforms periodic monitoring in terms of accuracy and CPU cost while achieving a close-to-optimal communication cost. Furthermore, the framework is robust and scales well with various parameter settings, such as privacy requirement, moving speed, and the number of queries and moving objects.

**References**

[1]. Chen.J and R.Cheng, (2007) "Efficient Evaluation of Imprecise Location-Dependent Queries," Proc. IEEE Int'l Conf. Data Eng., pp. 586-595.

[2]. Gedik.B and L. Liu,(2005) "Location Privacy in Mobile Systems: A Personalized Anonymization Model," Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS), pp. 620-629.

[3]. Hjaltason.G.R and H. Samet, (1999)"Distance Browsing in Spatial Databases," ACM Trans. Database Systems, vol. 24, no. 2, pp. 265-318.

[4]. Hu.H, J. Xu and D.L. Lee,(2005) "A Generic Framework for Monitoring Continuous Spatial Queries over Moving Objects," Proc. ACM SIGMOD, pp. 479-490.

[5]. Haibo Hu, Jianliang Xu and Dik Lun Lee (2010)"PAM: An Efficient and Privacy-Aware Monitoring Framework for Continuously Moving Objects" IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 3.

[6]. Jensen.C.S, D. Lin and B.C. Ooi, (2004)"Query and Update Efficient B+- Tree Based Indexing of Moving Objects," Proc. Int'l Conf. Very Large Data Bases (VLDB).

[7]. Saltenis.S, C.S. Jensen and M.A. Lopez, (2000)"Indexing the Positions of Continuously Moving Objects," Proc. ACM SIGMOD.

[8]. Tao.Y, D. Papadias, and J. Sun, (2003)"The TPR*-Tree: Optimized Spatio-Temporal Access Method for Predictive Queries," Proc. Int'l Conf. Very Large Data Bases (VLDB).

[9]. Tao.Y, D. Papadias and Q. Shen, (2002) "Continuous Nearest Neighbor Search," Proc. Int'l Conf. Very Large Data Bases (VLDB).

[10]. Yu.X, K.Q. Pu and N. Koudas, (2005)"Monitoring k-Nearest Neighbor Queries over Moving Objects," Proc. IEEE Int'l Conf. Data Eng. (ICDE).

* * * * *