International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 3 (2015) pp. 7133-7145

http://www.ripublication.com

# Answering Closest-Pair Nearest Neighbor Using Voronoi Diagram For Location Dependent Information System In Mobile Environment

**Mary Magdalene Jane.F and Ilayaraja N**

*Department of Computer Applications*
*Dr.N.G.P. Institute of Technology Coimbatore India*
*janejayakumar@gmail.com*
*Department of Applied Mathematics and Computational Sciences*
*PSG College of Technology Coimbatore India*
*ilayarajaan@yahoo.co.in*

**Abstract**

Location dependent information system (LDIS) has received more and more attention recently with the wide availability of mobile devices (smart phones, tablet, notebook, iPad etc.). Location dependent queries (LDQ) such as range query, window query and nearest neighbor (NN) query are gaining popularity. One of the most important LDQ is the closest pair nearest neighbor (CPNN) query, which is to find the closest pair of objects among the two data sets close to the user's current location where the query is issued. User may be interested in finding the closest restaurant and medical store or the closest supermarket and parking, etc. Users may request successive NN queries for theatre and restaurant. In previous works, CPNN query for LDIS, considers two items of the closest pair from two different points of interest (POI) data set indexed by R-trees. In this work, we address the closest pair nearest neighbor query using the Voronoi index structure and cache management strategies. The experiments conducted on our proposed cache strategies show an increase in cache hits which in turn help reduce communication costs and support rapid spatial query processing.

## 1 Introduction

Continuous innovation in wireless communication environment and leveraging the use of Internet on the move using handheld devices and the localization technologies (GPS, AGPS and others), mobile network operators rely on location based services (LBSs) as a key-asset to provide highly personalized services. By utilizing the user's current location, LDIS supports various types of spatial queries, which are all related on any of spatial properties of spatial objects. Nearest neighbor query is very popular

and common among the users of LDIS. For example, "Find the nearest hospital to me", is nearest neighbor query that seeks the spatial object more close to the query given point. Like the NN query, mobile users are interested in finding the spatial objects that are possibly from two different spatial data sets that have close distance between them. For examples, 'finding the closest medical store and clinical lab' and 'the closest supermarket and parking', etc. This type of special query is called as closest-pair nearest neighbor query or also known as nearest group query (NGQ) [20]. Computation of closest-pair NN queries has been studied for several decades. However, NN and CPNN queries have received considerable attention from LDIS research community [10, 5, 6, 20].

Caching is considered as one of the important techniques to relieve bandwidth constraint imposed on wireless mobile systems. Copies of remote data can be kept in the local memory of mobile devices to substantially reduce data retrievals from the original server. Caching not only reduces the uplink and downlink bandwidth consumption but also improves latency in data access. By caching the result of CPNN in the client device memory, it is possible to answer the individual NN query from the client itself when queries are issued from the same valid scope. Thus, caching frequently accessed data in mobile devices can potentially minimize communication cost.

The paper is organized as follows: Section 2 explores the existing works and motivation on CPNN and caching mechanism; Section 3 explains the closest-pair nearest neighbor query processing; Section 4 deals with the implementation of CPNN query using Voronoi index structure; Section 5 deals with the performance evaluation of the proposed system; Section 6 concludes the paper.

## 2 Related work

The related work pertinent to our proposed approach can broadly be classified into two areas, namely nearest neighbor query processing and cache management in mobile environments. A most popular and important query in LDIS and geographic information systems (GIS) is the NN query search [16, 17]. The NN query is: given a set of instances of a facility type, a distance metric and a query object *q*, find a facility instance "closest" to *q*. The distance between two objects is measured using some metric function over the underlying data space. In the existing works on NN queries, the Euclidean distance [6, 14, 20] and graph path length, e.g., road distance [3, 4, 12] are used as common distance metrics. The Euclidean distance can be applied for expressing the concepts of "neighborhood" and "closeness". The concept of "neighborhood" is related to the discovery of all objects that are "near" to a given query object. The concept of "closeness" is related to the discovery of all pairs of objects that are "close" to each other [16]. According to [3], NN queries retrieve the object of a certain class which is the closest to a certain object or location. As an example, consider the query "Retrieve the closest hospital to my location". This NN query retrieves the nearest hospital to the user's current location. A point-NN query is a conventional NN query [4, 14] (e.g., "find the nearest ATM to my hotel"). A variant to the point-NN query is a closest pair query between two POI datasets [1, 7]. For an

example, "find the pair of a gas station and a restaurant that has the smallest distance between them". Queries to prefetch items into the mobile cache also has been proposed[8,9].

Constructing Voronoi structures [10,18] is a projected a method for answering LDQ in a mobile environment. Voronoi diagrams (VD) have been built on the data objects to serve as an index for the data objects to answer NN queries.

Temporal-based cache replacement strategies, such as LRU (least recently used) [3], LFU (least frequently used) and LRU-K[7] have been studied widely in the past. Kam-Yiu Lam [5] proposed a policy called invalid-LRU. Cache replacement decisions are made based on the distance between a client's current location and the origin location of each cached data object in Manhattan Distance-based cache replacement policy [4]. PA (Probability Area) and PAID (Probability Area Inverse Distance) [10] are both cost-based replacement policies. The Weighted Predicted Region based Cache Replacement Policy [2] is an extension to PAID which considers data size and the weighted data distance of an item along with the probability of data access. It is already proven that the replacement policy RAAR (Re-entry probability, Area of valid scope, Age, Rate of Access) works well with data on demand [10, 13].

The motivation for this work is to implement existing RAAR cache replacement mechanism for answering the CPNN query and NN query using Voronoi index diagram. Client side cache memory can be efficiently utilized for answering the user query with reduced latency, potentially minimizing the communication cost and increase the data availability.

## 3 The Closest-Pair Nearest Neighbor Query

In this paper, we extend NN query to another type of widely useful query named CPNN or 1NG query[15]. Unlike a NN query, a CPNN query finds "closest" group of objects from two data sources, where each group consists of exactly one object from each data source.

**Definition 1:** *Given two spatial data sets A and B, the* Closest Pair (CP) query *finds the pair of objects (a, b) such that* $a \in A$, $b \in B$, *and* $a' \in A$ *and* $b' \in B$, *distance*(a, *b*) $\leq$ *distance*(a*',b'* ).

Here *distance*(*s*,*t*) corresponds to the Euclidean distance of two objects *s* and *t*. A practical extension of CP is the *Range-CP* problem, where a spatial range *R* is involved and we are interested in finding the closest pair of objects inside *R*. The Range-CP query finds the closest pair of objects from a subset of objects in data sets, where the locations of subset of objects are in a given spatial range.

**Definition 2:** *Closest-pair NN query is defined as given two location dependent POI datasets* $D_A$ *and* $D_B$, *the output of the closest-pair NN query is composed of* $O_a$, $O_b$, *such that* $O_a$ *and* $O_b$ *are spatially closed to user's query point q.*

**Example : CPNN query processing**

When the user issue a query *Q(Restaurant, Medical store)* as pair query, Restaurant is considered as primary query and Medical store is considered as secondary query. The CPNN query finds a pair of restaurant and medical store from two data sets with shortest distance between them.

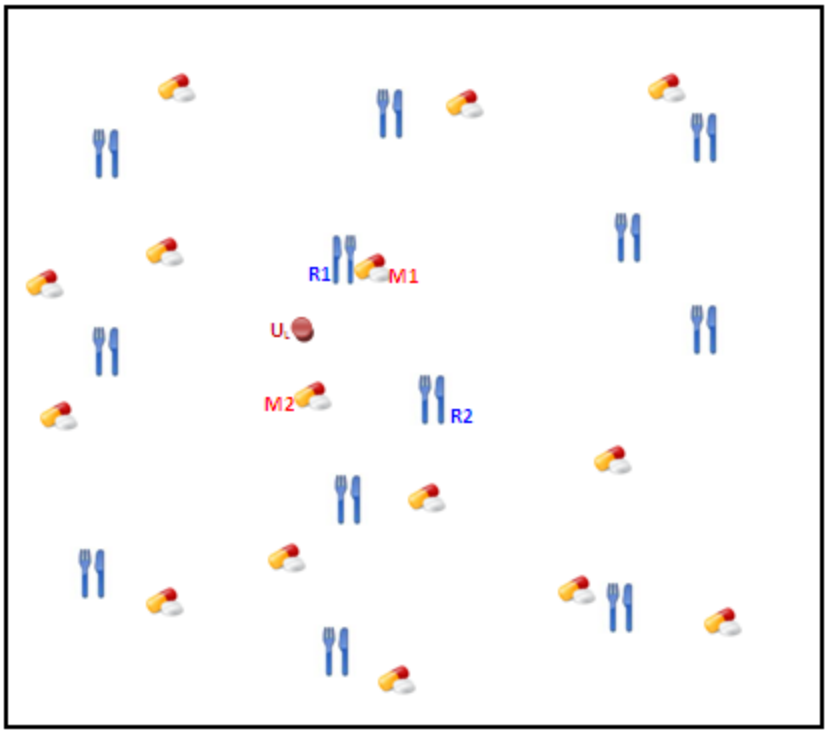


**Figure 1 Spatial representation : Restaurant and Medical store**

For example, Figure 1 shows the spatial distribution of two data sources *R* and *M*. *R* consists of restaurants and *M* represents medical stores. If a user is located at point $U_L$ and wants to find the nearest restaurant by issuing the NN query, $R_1$ will be returned as it is nearest to $U_L$ in data source *R*. If the user is interested to find the nearest medical store, then $M_2$ is the result. Here, consider this scenario, if the user wants to utilize a restaurant for lunch and a medical store for getting medicine after his lunch immediately, user has to travel lot because the distance between the returned two objects $R_1$ and $M_2$ is greater. They are not spatially close to each other.

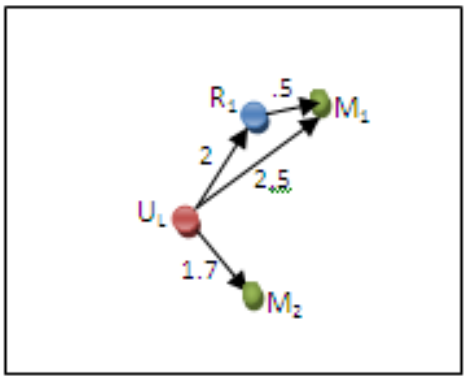


**Figure 2 Euclidean Distance between items and user**

User prefers the two locations (restaurant and medical store) to be close to each other and also both to be near his location. If we use the POI data sets as in Figure 1 to answer this query in such scenario, a group of two items ($R_1$,$M_1$) will be returned as a closest-pair nearest neighbor to the user's current location.

Given two datasets R and M, a closest-pairs query retrieves the pair ($R_1$,$M_1$), $R_1 \in R$, $M_1 \in M$ that are closest in the datasets (e.g., find the restaurant, medical store pair within the smallest driving distance). Closest pair can be selected by calculating the Euclidean distance between items and user as shown in Figure 2. We use the following formula to calculate Euclidean distance between locations of two items.

In Cartesian coordinates, if p = ($p_1$, $p_2$,.., $p_n$) and q = ($q_1$, $q_2$,..., $q_n$) are two points in Euclidean n-space, then the distance from p to q, or from q to p is given by:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} = \sum_{i=1}^{n} (q_i - p_i)^2 \quad 1$$

Distance between user's query locations $U_L$ to location restaurant $R_1$ is calculated using the formula 1.

Latitude and longitude coordinates of user is represented as $U_L(x_1,y_2)$ and restaurant is represented as $R_1(x_2,y_2)$. Among the calculated distance values LDIS system returns the closest pair $(R_1,M_1)$ as query response.

$$Dis(U_L, R_1) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2} \quad 2$$

CPNN query can be extended to get the data from more than two POI data sets is called *k*NG query. Consider the scenario, when a user has a shopping list in which each item can be purchased from multiple candidate shops, he can utilize *k*NG query to select the nearest shops to cover all the items. For example, User has to find the nearest "fuel station", "medical store", "milk store" and "Coffee shop", so that the user can fill fuel, get a medicine, get milk and then have a coffee to save time. User issues a query *Q(Fuel, Medical store, Milk store, Coffee shop)* to get nearest services.

Cache the data items returned as the response to CPNN query in client device is essential to answer any of NN query issue by the user in near future. The objective of this work is to provide a novel caching mechanism to cache the CPNN query items with the spatial and temporal valid scopes so that we increase the cache hit ratio. This causes a drastic change in the cache placement, invalidation and replacement policies when compared to those in the previous policies where no grouping of services (CPNN query) is done. The cache-hit ratio is expected to be relatively high and so users can avoid queries, resulting in unnecessary network usage. This also would reduce the rate of cache replacement as we hoard the cache with related data which is highly probable to be queried in the near future.

## 4 CPNN Query Processing using Voronoi Diagram

The Voronoi Diagram (VD) is a traditional data structure in computational geometry, primarily designed for evaluating nearest-neighbor queries over two-dimensional spatial points [11, 19, 20], has raised plenty of research interest. Conceptually, the VD partitions the data space into disjoint 'Voronoi cells' (VC) with the associated point in set of points S, is called generator point, so that all locations in the same Voronoi cell have the same nearest neighbor than any other generator point[19]. A VD is a special kind of decomposition of a metric space determined by distances to a specified discrete set of objects in space [12]. Thus, preprocessing can be done to construct the corresponding VD of different services (POIs datasets).

For an example, given a set of points of S for a particular POI's, the corresponding Voronoi diagram will be generated. Each generator point $s_i$ has its own Voronoi cell $V(s_i)$, which consists of all locations closer to $s_i$ than to any other generator points. A VD is constructed on the data objects of POI type for answering the NN queries. As per [19], three data structures are used to record the preprocessed data of Voronoi diagram of POI, such as edges denoted by <*edge_id; x*1*; y*1*; x*2*; y*2>, *service object* denoted as <*service_id, Location(x,y) ,number_of_edges, list_of_edges_ids*>, edges-service denoted as <*segment id; serv object id*1*; serv object id*2>. The task of finding the nearest neighbor of a query point is then reduced to a point query.

Thus, preprocessing can be done to construct the corresponding VD of different services (POIs datasets). POIs are categorized into different kinds such as restaurants, fuel, medical store, hospitals, gas stations, etc. For each kind of POI service, a VD index is constructed based on the data structures as discussed [20]. Overall, the maintenance cost of a VD index is reasonable considering the gain in query performance.

Voronoi diagrams for restaurant and medical store are constructed as shown in Figure 3 and Figure 4. Figure 3 illustrates a Voronoi diagram of restaurant POI. The user issues a NN query to find the nearest restaurant from query point q. Since the query point q is located in the Voronoi cell of restaurant item $R_1$, $R_1$ is the nearest neighbor of q. The $R_1$'s detail is returned to client as an answer of NN query along with the spatial valid scope and temporal valid scope of an item $R_1$. Spatial valid scope is the vertices of polygon (Voronoi cell) of an item $R_1$. Temporal valid scope is

defined as operating time of an item $R_1$. Client stores the item $R_1$'s details along with the spatial valid scope and temporal valid scope.

CPNN query processing at server side is explained in algorithm 1. For an example query *Q(Restaurant, Medical store)*, find the nearest pair to user location as discussed in previous section is answered by using the Voronoi index structure by performing the iterative processing. The query first starts using VDI of primary query to find the VC, which contains the user query location *u* as shown in Figure 4.b. VC of restaurant item r1 contains the user location *u*. Once the nearest restaurant of *u* is found, now find the nearest medical store $m_1$close to the restaurant $r_1$ and calculate the distance from *u* to from $m_1$ *as* $dist(u,r_1,m_1)$. Find the next primary query neighbor of *u* (adjacent VCs of $r_1$) and check $dist(u,r_1,m_1)$ *is less than dist(u, r2)* then no need to furthermore to find the nearest medical store of $r_2$, repeat the same procedure to find next primary query neighbor of *u* and repeat the distance calculation and check the distance between user location *u* to next nearest restaurant item with $dist(u,r_1,m_1)$. If dist(u, $r_{new}$) is lesser than the $dist(u,r_1,m_1)$ *then new nearest medical store should be found.* Then check the $dist(u,r_1,m_1) > dist(u,r_{new},m_{new})$ then new pair items are $(u,r_{new},m_{new})$. Repeat this procedure for all adjacent restaurant items of VC of $r_1$, find the closest pair neighbor of *u*, having minimum distance among the all, then server returns the (<$r_1$, *spatial valid scope, temporal valid scope*>, <$m_1$, *spatial valid scope, temporal valid scope*>) as a closest pair neighbor to user location *u* to client.

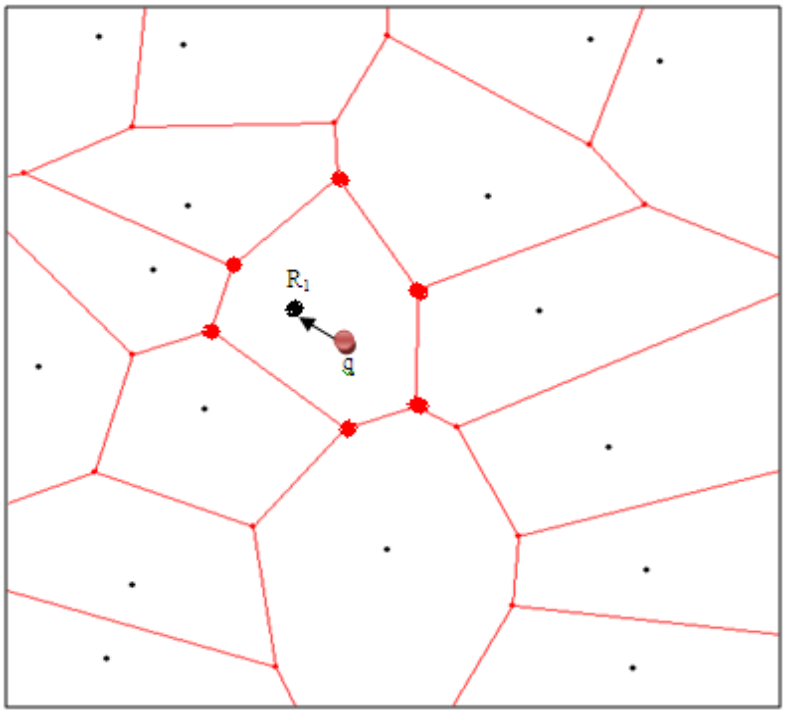


**Figure 3 Voronoi diagram for Restaurant**

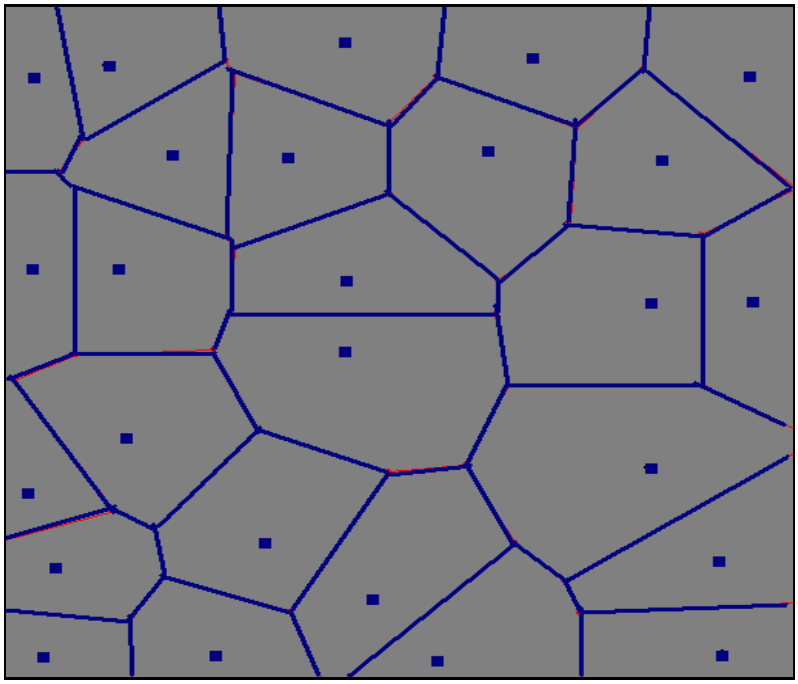

**Figure 4.a. Voronoi diagram for Medical store**

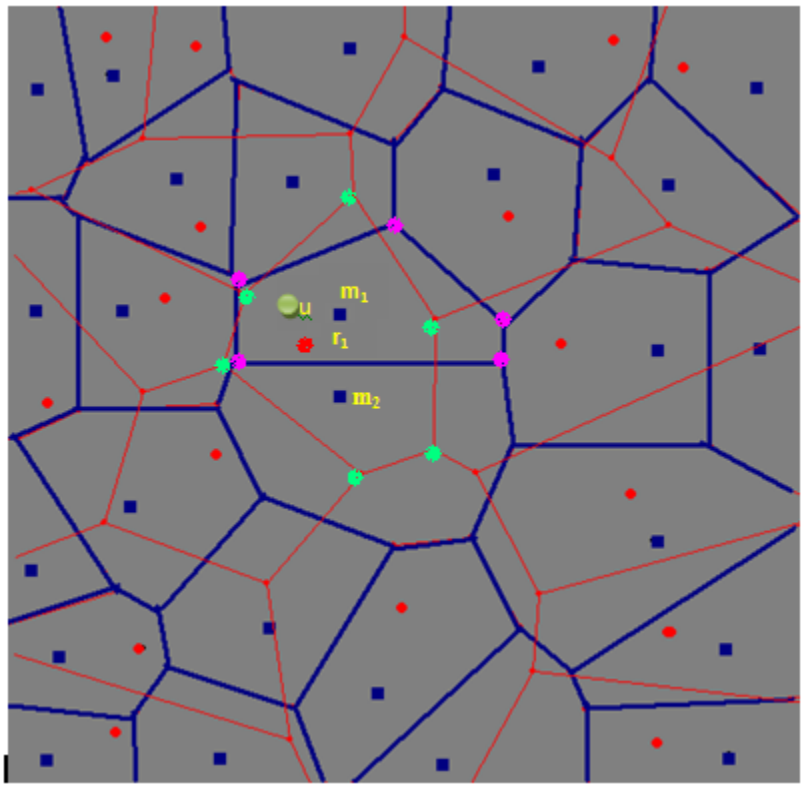


**Figure 4.b Voronoi diagram chaining for Restaurant and Medical store**

### 4.1 Algorithm for CPNN query

***Algorithm: CPNN-Query***

***Input:*** *User location($U_{x,y}$), Q(Primary item query(P), secondary item query(S)), POI datasets(POIDS), preprocessed Voronoi index structure(VDI) for each category of POI*

***Output:*** *(($P_j$,SVS($P_j$),TVS($P_j$)), ($S_j$,SVS($S_j$),TVS($S_j$))) as CPNN(Q)*

*1 Let U be the query point*
*2 Find the Voronoi Cell VC($P_1$) = contains (U) in VDI(P)*
*3 Find the generator point $P_1$ as first nearest primary query item of U*
*4 Find the generator point $S_1$ in VC($S_1$) contains $P_1$ as first nearest secondary item to location of ($P_1$) // point NN query*
*5 Compute the dist(U,$P_1$,$S_1$) = dist(U,$P_1$) + dist($P_1$,$S_1$)*
*6 j=1, min=1*
*7 For i=2 to all generator points of adjacent VCs of ($P_1$)*
*8 If( dist(U,$P_{min}$,$S_{min}$) < dist(U,$P_i$))*
*9 goto step 7*
*10 else*
*11 {*
*12 j=i;*
*13 Find the generator point $S_j$ in VC($S_j$) which contains $P_j$ as first nearest*
*Secondary query item $S_j$ to location of generator point of $P_j$*
*14 Compute the dist(U,$P_j$,$S_j$) = dist(U,$P_j$) + dist($P_j$,$S_j$)*
*If(dist(U, $P_{min,}$ $S_{min}$) > dist(U,$P_j$, $S_j$))*
*min=j;*
*End if*
*}*
*15 End if*
*16 End for*
*17 Return (($P_{min}$, SVS($P_{min}$), TVS($P_{min}$)), ($S_{min}$, SVS($S_{min}$) ,TVS($S_{min}$) )) as CPNN(Q)*

## 5. Analysis

The network is considered as a single, large service area within which the mobile clients request location-dependent information services. The service area is represented by a rectangle with a fixed size of 4000 * 4000 m. A "wrapped around" model for the service area is assumed where a client leaves one edge of the service area to enter the service area from the opposite edge at the same velocity. The database contains *ItemNum* items or services which are requested by the clients and every item has different values. For example, an item "hospital" may contain several values which are the names of all the hospitals in the service area. Every item is

assumed to contain *ScopeNum* different values within the service area. Each data value has a size of *DataSize*. In the simulation, the scope distributions of the data items are generated based on Voronoi Diagrams . The client and server execution models are discussed in [10]. The cache size is measured as a percentage of the data base size. The proposed cache replacement policy is RAAR+ as discussed in [10]. The secondary items should not be evicted from the cache while the intervening queries are issued. Hence the secondary items are retained in the cache till user has issued intervening queries whose number equals the semantic distance.

If the client requests the secondary item before the semantic distance, then the secondary item gets the status of on demand item and competes with the primary items for eviction. If the item is not explicitly requested by the user even after the maximum number of intervening queries, then it could be considered for eviction and it is left to compete with the on demand items.

The experimental results show that caching improves item hits and the cache replacement policy which to suit the closest pair where the secondary items were retained in the cache for an expected period of time. Results demonstrate a better cache hit ratio when prefetching is used along with RAAR+.

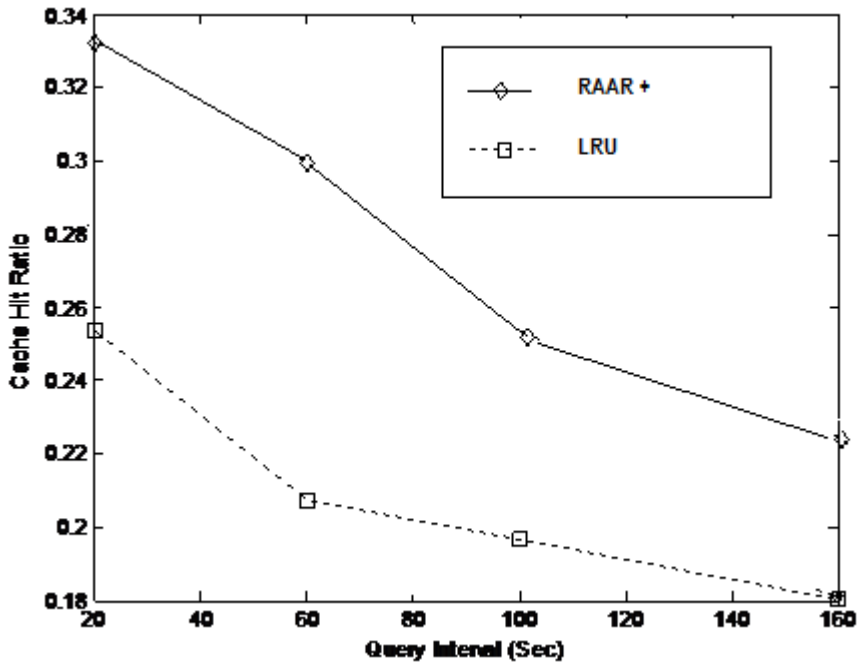


**Figure 5 Cache hit ratio Vs Query Interval**

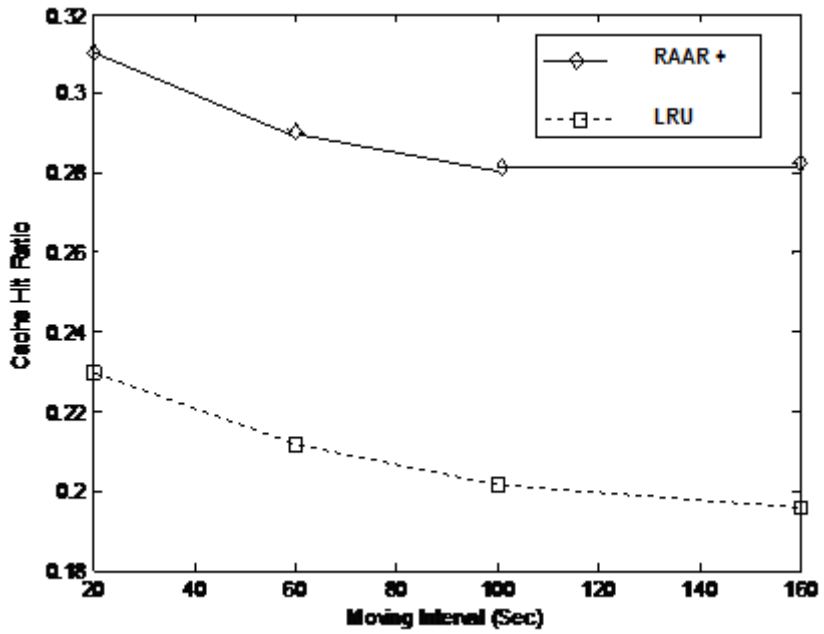


**Figure 6 Cache hit ratio Vs Moving Interval**

## 6. Conclusion and Future work

A cache management scheme has been proposed to store the results of the spatial queries in the client cache. Results demonstrate a better cache hit ratio when replacement was done using the proposed replacement policy where the secondary items are allowed to reside in the cache for an expected period of time. In this work, it is assumed that the data items are of fixed size and read-only, the possibilities of extending the policy to accommodate data of varying size and subject to regular updates would be taken up for further research.

## References

1. Ajey Kumar, Manoj Misra, A. K. Sarje 2007, A Weighted Cache Replacement Policy for Location Dependent Data in Mobile EnvironmentsSAC'07, March 11–15, 2007, Seoul, Korea. Copyright 2007 ACM 1-59593-480-4/07/0003
2. K. Lai, Z. Tari and P. Bertok, "Location–Aware Cache Replacement for Mobile Environments", IEEE Globecom 2004, pp. 3441-3447
3. E. O'Neil and P. O'Neil, "The LRU-k page replacement algorithm for database disk buffering", Proceedings of the ACM SIGMOD, 296-306, 1993
4. Q. Ren and M. Dunham, 'Using semantic caching to manage location dependent data in mobile computing', In Proceedings of the International Conference on Mobile Computing and Networking, pages 210–221, 2000.

5. B. Zheng, J. Xu, and D. Lee, 'Cache invalidation and replacement strategies for location-dependent data in mobile environments', IEEE Transactions on Computers, 51(10):1141– 1153, October 2002.
6. Tao, Feng, Fionn Murtagh, and Mohsen Farid, 'Weighted association rule mining using weighted support and significance framework', Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2003.
7. Wang, Wei, Jiong Yang, and Philip S. Yu. "Efficient mining of weighted association rules (WAR)." Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2000.
8. Cai, Chun Hing, et al. "Mining association rules with weighted items." Database Engineering and Applications Symposium, 1998. Proceedings. IDEAS'98. International. IEEE, 1998.
9. Tait, C., H. Lei, S. Acharya & H. Chang (1995). Intelligent file hoarding for mobile computers. Proc. First International Conf. Mobile Computing and Networking (MobiCom'95), Berkeley, California, United States: 119 – 125.
10. Mary Magdalene Jane F., Ilayaraja N., Safar M. and Nadarajan R. (2009), 'Cache Pre-fetching and Replacement strategies for Location Dependent Data in Mobile Environments', Journal of Digital Information Management, Vol. 7, No. 3, pp.185-190.A. Okabe, B. Boots, K. Sugihara, and S. Chiu 2000, 'Spatial Tessellations: Concepts and Applications of Voronoi Diagrams', Wiley, second edition.
11. J. Zhang, M. Zhu, D. Papadias, Y. Tao, and D. L. Lee 2003,'Location-based spatial queries', In SIGMO.
12. Mary Magdalene Jane F., Ilayaraja N., Safar M. and Nadarajan R. (2009), 'Cache Pre-fetching and Replacement strategies for Location Dependent Data in Mobile Environments', Journal of Digital Information Management, Vol. 7, No. 3, pp.185-190.
13. Ilarri, Sergio, Eduardo Mena, and Arantza Illarramendi. "Location-dependent query processing: Where we are and where we are heading." ACM Computing Surveys (CSUR) 42.3 (2010): 12.
14. Ilayaraja N, Mary Magdalene Jane F. et al (2009), 'Service type based Cache Replacement Policy for location dependent data in mobile environments', International Conference on Mathematical and Computational Models, pp.214-222.
15. Liria, Antonio Leopoldo Corral 2002, 'Algorithms for Processing of Spatial Queries using R-trees. The Closest Pairs Query and its Application on Spatial Databases', PhD Thesis.
16. Zhang B, J., Zhu, M., Papadias, D., Tao, Y., & Lee, D. L. (2003, June), 'Location-based spatial queries', In Proceedings of the 2003 ACM SIGMOD international conference on Management of data (pp. 443-454). ACM.
17. Sharifzadeh, M., & Shahabi, C. (2006, September), 'Additively weighted Voronoi diagrams for optimal sequenced route queries', In of the Third Workshop on Spatio-Temporal Database Management STDBM 06 (p. 33).

18. Xuan, K., Zhao, G., Taniar, D., Safar, M., & Srinivasan, B. (2011). Voronoi-based multi-level range search in mobile navigation. Multimedia Tools and Applications, 53(2), 459-479.
19. Zheng, B., & Lee, D. L. (2001), 'Semantic caching in location-dependent query processing', In Advances in Spatial and Temporal Databases (pp. 97-113). Springer Berlin Heidelberg.
20. Zhang, D., Chan, C. Y., & Tan, K. L. (2013, July), 'Nearest group queries', In Proceedings of the 25th International Conference on Scientific and Statistical Database Management (p. 7). ACM.