Location-Based Query processing: Sensing our Surroundings

Raghunath Rajachandrasekar, Zoya Ali, Srinivas Hegde, Vilobh Meshram and Nishanth Dandapantula
Department of Computer Science and Engineering,
The Ohio State University
{rajachan, aliz, hegdes, meshram, dandapan}@cse.ohio-state.edu

***Abstract*—The ever-growing mobile user base, and the increasing number of mobile services which rely on the user's locations naturally led us to select this as our survey topic. Query-based services have been around for quite a long time. Whereas, automatic location detection based on inputs from GPS devices embedded in smart phones and other such mobile devices is a relatively new technology. The convergence of these two technologies has given rise to a wide spectrum of services and applications, which can be beneficial to the user, as well as the service provider. In this report, we look at the evolution of this convergence, try to analyze the different challenges and evaluate the solutions that have been proposed to counter these challenges. We have also proposed our own services and discussed the feasibility of implementing them, given the current state-of-the-art in location detection hardware and the limitations that the smartphone software the the network pose. We have also given a comprehensive summary of the past literature and on-going work in this area of research.**

I. Introduction

The number of smart phone users is increasing exponentially. The basic smart phone hardware has been continually evolving in sync with the software which can exploit what the hardware can provide. In the process, smart phones have become both the consumer and producer of data, which can be anything ranging from user profile information, to user location.

Location of an object or a person is its geographical position on the earth with respect to a reference point [1]. Such location data is real-time and traceable, and not a one-off value that is read statically. This information can be characterized by using a number of different representations including latitude/longitude/altitude or street address, etc. With the state-of-art in location detection hardware, the granularity with which user location can be estimated is mind-blowing. Current-day location representation standards have the capability to serve the complete granularity spectrum.

There are several mobile applications and services which are 'reactive' in nature, in the sense that they respond to the user's query with specific information. This query can have a search string as a payload, or some user-defined data on which the application acts to provide the desired result. So how different is this query, from its location dependent counterpart? It makes the service / application 'proactive'. Consider, a traditional query-based search application which provides information about grocery shops. The service provider will be maintaining a back-end global database, with information about all the grocers in the country. So a user request, will query the entire database to get information about the particular shop that the user is interested in, and gives it to the user. If interested, the user will include his city along with his search string, thereby reducing the number of results he will be returned. This doesn't change the way the data based is queries in anyway. The entire database is traversed and the information about grocers which match the input string is returned. If we were to use Location Based Queries instead, the user just searches for a particular grocery shop, and the user location is automatically detected by the smart phone and encapsulated along with the query. This also enhances the way the query is processed in the back-end. The service provider configures the database in such a way that, based on the location, the query works only on a subset of the database, which has records relevant to the location in question.

The rest of this report is organized as follows. In section II, we give a background about the key components involved in our design. Then in section III, we talk about the different types of Location Based Queries and compare them. We also talk about the Query Processor and its working principle. In section V, we present the different services that can make use of LBQs. Section VI talks about the different challenges and open issues. A summary of related work is given in Section VII and in section VIII, we discuss the conclusion and future work.

II. Background

For a better understanding of location based services and queries we need to know what a geographic domain is and what a location is in terms of providing ubiquitous data access through portable, mobile computing devices. A geographic Domain is defined as the entire area covered by the Mobile Computing Platform. Thus a mobile unit can freely move around in a geographic domain. A location us a precise point within the geographic domain. It represents the smallest identifiable position in the domain. It can be represented in terms of longitude/latitude pair. Each location is identified by a specific ID.

A. Location Dependent Data

Location-dependent Data(LDD) has great promise for mobile and pervasive computing environments. They can help in providing local and non-local news, weather, and traffic reports as well as directory services. By definition, location of an object or a person is it's geographical position on the earth with respect to a reference point and thus LDD is defined as the information whose value is determined by location to which it is related. For example, In a cross country road trip, the radio stations tuned in keep varying as the location changes. The query sent out to obtain the available radio stations is the same, but the results vary as the location keeps changing. Thus, for LDD, the semantics of a set data and their values are tightly coupled with a particular location.

B. Location Based Queries

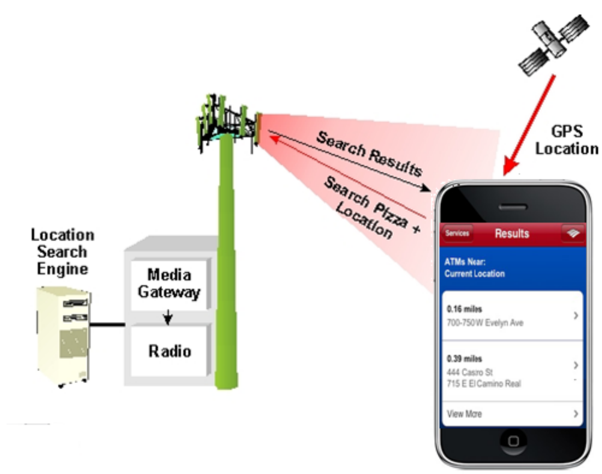

Fig. 1. Location Based Queries

Location based Queries (LBQ) provide support for location based Services. Location based applications (LBA) provide location based services (LBS) by using queries called location based queries (LBQ) The result of these queries is based on the location of the mobile user. LBQs can be categorized into four types which include Range Queries, Nearest Neighbor Queries, Navigation Queries and Geo-Fence Queries. Range queries, for example, deal with the number of entities within a certain boundary. Nearest neighbor queries deal with the closest entity to the point of query origin. Navigation queries deal with the route calculation between two geographic locations. Geo-Fence queries broadly deal with setting up a virtual fence and the concentrating on the communication with the mobile devices which trespass into that boundary.

C. Location Based Services

Location-dependent services (LDSs) are an important class of context-aware applications. They answer location-related queries, where a location is either explicit or implied. These services emerged from advances and convergence in high-speed wireless networks, personal portable devices, and location-identification techniques. With a variety of promising applications, such as local information access (traffic reports, news, navigation maps, and so on) and nearest-neighbor queries (such as finding the nearest restaurant), LDSs will soon become an integral part of our daily lives.

Although LDSs exist in traditional computing environments (Guides@Yahoo, for example), their greatest potential is in a mobile-pervasive computing environment, where users enjoy unrestricted mobility and ubiquitous information access. The Mobile environment constraints pose a major challenge in managing data in an LDS. Mobile-pervasive environments offer scarce bandwidth, low-quality communication, frequent network disconnections, and limited local resources, complicating the provision of location-dependent information to mobile users. Another challenge is that query results depend on a querys spatial properties. For a location-bound query, the query result must be both relevant to the query and valid for the bound location. The basic advantage of ubiquitous service also poses a challenge because mobile users change locations and some tasks such as query scheduling and cache management are particularly tough in an LDS.

The market has been flooded with many location based services in the contemporary society. The broad classification includes Navigation Services, Emergency Services, Advertising Applications, Games and Social Services Features, Tracking Services and Billing Services. There can be several ways to apply this technologies in all the above mentioned services.

D. Smart phones and Positioning

A smart phone is a mobile phone that offers more advanced computing ability and connectivity than a contemporary basic feature phone. Smart phones and feature phones may be thought of as handheld computers integrated within a mobile telephone, but while most feature phones are able to run applications based on platforms such as Java ME, a smart phone allows the user to install and run more advanced applications based on a specific platform. Smart phones run complete operating system software providing a platform for application developers.

The current smart phones follow the GPS based positioning system. Many emerging smart phone applications require position information to provide location-based or context-aware services. In these applications, GPS is often preferred over its alternatives such as GSM/WiFi based positioning systems because it is known to be more accurate. However, GPS is extremely power hungry. Hence a common approach is to periodically duty-cycle GPS. However, GPS duty-cycling trades-off positioning accuracy for lower energy. A key requirement for such applications, then, is a positioning system that provides accurate position information while spending minimal energy.

An alternative solution is the rate-adaptive positioning system for smart phone applications. It is based on the observation that GPS is generally less accurate in urban areas, so it suffices to turn on GPS only as often as necessary to achieve this accuracy. This technique uses a collection of techniques to cleverly determine when to turn on GPS. It uses the location-time history of the user to estimate user velocity and adaptively turn on GPS only if the estimated uncertainty in position exceeds the accuracy threshold. It also efficiently estimates user movement using a duty-cycled accelerometer, and utilizes Blue tooth communication to reduce position uncertainty among neighboring devices. Finally, it employs cell tower-RSS blacklisting to detect GPS unavailability (e.g., indoors or among skyscrapers) and avoid turning on GPS in these cases.

III. Classification of Queries

A. Range Queries (RQ)

Range Queries retrieves the objects located within a specific region. Range queries can be static range queries or moving/mobile range queries. If the region is rectangular then the range queries are known as window queries.Window queries can be static or dynamic. If the objects of the query are stationary or moving, they can take any of the query type (dq, ddt), (dq, ds), (sq, ddt) and (sq, sdt) where (d dynamic, dt data, s static, q query).

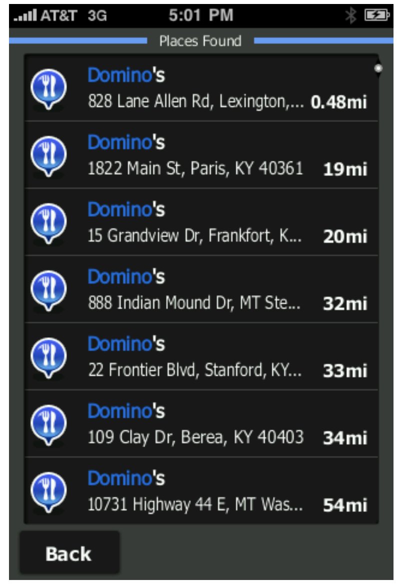

Fig. 2. Range Query

B. Nearest Neighbor Query (NNQ)

Nearest neighbor (NN) queries are responsible for getting the objects closest to a specific location. If they are capable of getting k objects other than the nearest one then they are called kNN queries. Static database is used to execute static NN queries and dynamic database is required to execute dynamic NN queries. A reverse kNN query can fetch the objects that have a specified location among their k nearest neighbors. In spatial databases, constrained NN queries are the queries with a range constraint for the objects retrieved.

C. Navigation Query (NQ)

Navigation queries can fetch the mobile user a suitable path to the destination taking the network traffic into consideration. These type of queries use either of the following forms: a predicate is applied to many objects; e.g., nearest neighbor queries or a predicate is checked for every object with respect to the other one. Continuous queries are evaluated continuously until the user decides to terminate them. Persistent queries are continuous queries that consider both the current state and the past states of the moving objects.

D. Geo-Fence Query (GFQ)

The Geo-Fence query allows users to create a virtual monitoring boundary on a geographic area on the map.

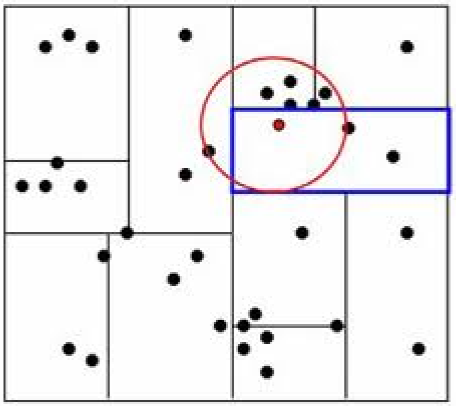

Fig. 3. Nearest Neighbor Query

When that boundary is crossed by a moving object, the intrusion is recognized as an event and the user is notified by a SMS text or email message. It is important to note that there are several commercial applications of Geo-Fence. All these applications make the simplifying assumption that the distance between the objects/points is Euclidean distance during the query processing. In real-world scenarios, however, the objects move in spatial networks, where the distance between the objects is the length of the shortest path (i.e., network distance) connecting them. See Fig. 4.

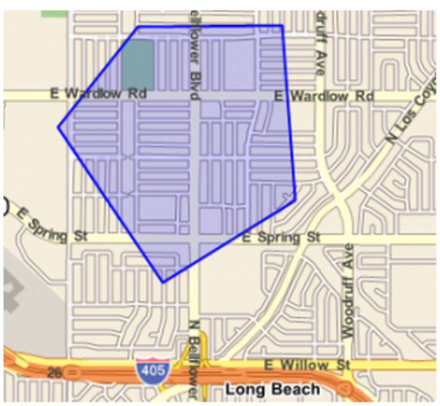

Fig. 4. Geo-Fence Query

IV. Query Processor

The mobility that is made possible by the usage of car-based or handheld GPS devices in metro cities results in two fundamental system requirements: distance computations based on a (road) network and processing of moving Points of Interest (POIs). There exist an increasing number of applications that require query processing of moving POIs based on an underlying network. Several techniques have aimed at solving location-based (Continuous) k Nearest Neighbor (kNN/C-kNN) queries in spatial networks. These methods assume that the positions of POIs are fixed (e.g.,gas stations or bus stops). While the ability to process moving POIs is challenging, it also enables new applications and in the most general case the query points and the POIs are interchangeable, i.e., users or vehicles equipped with GPS devices are able to report their positions and hence themselves become POIs. We present our novel design of a system to process location-based queries on moving objects in road Networks (MOVNet). MOVNet focuses on executing snapshot range and nearest neighbor queries over moving POIs in a stationary road network. Fig. 5 illustrates the system infrastructure and components of MOVNet. Specifically, MOVNet combines an on-disk R-tree [8] structure to store the connectivity information of the road network with an in-memory grid index to efficiently process moving object position updates. A feature of MOVNet is the bi-directional mapping between the two structures that enables the retrieval of a minimal set of data for query processing. We introduce the concept of affected cells that form the set of grid cells overlapping with a given edge and provide an efficient algorithm to compute these cells.

A road network (or network for short) is a directional weighted graph G consisting of a set of edges (i.e., road segments) E, and a set of vertices (i.e., intersections,dead ends) V [2]. A network in MOVNet is transformed into a modeling graph in memory. Specifically, graph vertices represent the following three cases: (i) the intersections of the network, (ii) the dead end of a road segment, and (iii) the points where the curvature of the road segment exceeds a certain threshold so that the road segment is split into two to preserve the curvature property. The modeling graph is a piecewise approximation of the network. Fig. 6(a) shows an example road network, and Fig. 6(b) demonstrates the corresponding modeling graph.

There are different objects (e.g., cars, taxis, and pedestrians) moving along the road segments in a network. These objects are identified as the set of moving objects M. A moving object m in M is a POI located in the network. The location of m at time t is defined as loct(m) = (xm, ym), where xm and ym are the x and y coordinates of m at time t, respectively. A query point q is a moving object, a subset of M, issuing a location-based spatial query at any time. MOVNet assumes that

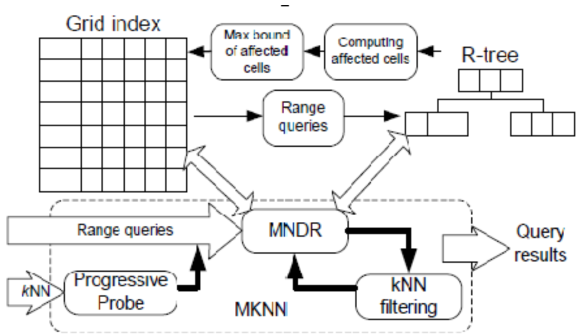

Fig. 5. Index Structures and MOVNet query processing modules

periodic sampling of the positions of moving objects is used to represent their locations as a function of time. A spatial query submitted by a user at time t1 is computed based on loct0 (M), where the system has the last snapshot of moving objects at t0 with t0 t1, t1 t0 ¡ delta(t), delta(t) is a fixed time interval, and the result is valid until t0 + delta(t).

The distance between two moving objects highly depends on the length of edges and the connectivity of vertices as well as the current locations of the objects. To efficiently manage both stationary network connectivity and dynamic object position updates, MOVNet utilizes a dual-index structure. First, an on-disk R-tree stores the stationary network data. Second, an in-memory grid index supports the position updates of moving objects. We use a vertex list to store the coordinates of vertices in the graph. For each vertex, a linked list records the edges going out from it. To quickly locate a vertex in the list, MOVNet manages a hash table to map the coordinate of a vertex into its index in the vertex list. Without loss of generality, we assume the service space is a square. We can partition the space into a regular grid of cells with a size of c c. We use c(column, row) to denote a specific cell in the grid index (assuming the cells are ordered from the bottom left corner of the space). At time t, a moving object m has loct(m) = (xm, ym), therefore it overlaps with cell c. Each cell maintains an object list containing the identifiers of enclosed objects. The objects coordinates are stored in an object array, and the object identifier is the index into this array. Fig. 7 shows a part of the network of Fig. 6(b) that is managed by a grid index with 8 8 cells. An example object on e(v2, v4) is enclosed by c(5, 5). Accordingly, the object list of c(5, 5) records the object identifier and hence we can retrieve the coordinate of the object from the object array. We define the set of cells {c1, c2, ...cn}, where an edge e(v1, v2) consecutively overlaps from v1 to v2, as the set of affected cells.

V. LOCATION-BASED SERVICES

LBSs are information services accessible with mobile devices through the mobile network and utilizing the ability to make use of the location of the mobile device. They are wireless-IP services that uses geographic information to serve a mobile user. Any application service that exploits the position of a mobile terminal can also be called LBS.

These definitions describe LBS as an intersection of three technologies, see Fig. 8. It is created from New

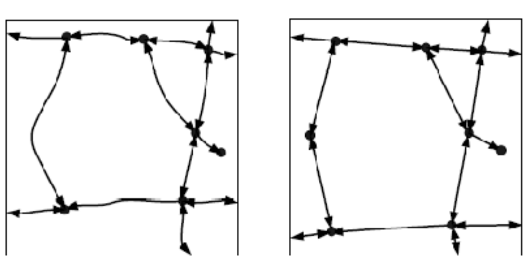

Fig. 6. Example road network and model graph

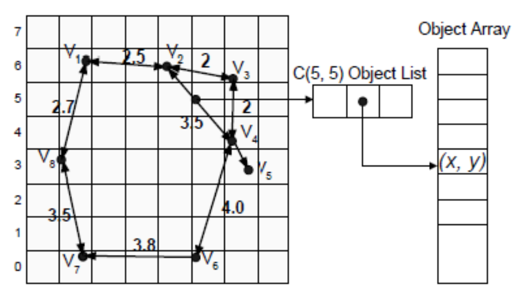

Fig. 7. Network indexed by grid index and its data storage

Information and Communication Technologies (NICTS) such as the mobile telecommunication system and hand held devices, from Internet and from Geographic Information Systems (GIS) with spatial databases.

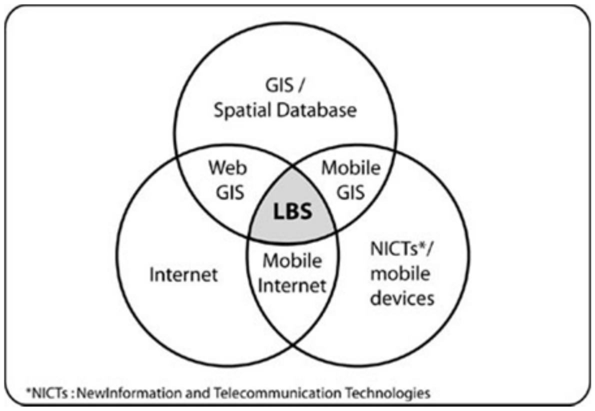

Fig. 8. Location Based Services

A. Navigation Services

Navigation and information services are services like digital travel assistants that provide driving directions based on time taken to travel, distance covered. Some other examples are location-dependent web pages [3] and mobile yellow pages that provide information about nearby points of interest [4].

When a query is sent to obtain the location of the user from a mobile device, it is an iterative process. For example, consider a person moving from point A to point F and consider that points B, C, D, E are between points A and F. A query is fired to obtain the nearby places of interest at point A. But the person is already at point B by the time the results from the query sent at point A are obtained. Thus another query to obtain the nearby places of interest is sent at point B and a set of results more stabilized than the previous results are obtained. An improved estimation of the location of the user is also obtained. This iterative process continues at all the intermediate points until we obtain a stabilized set of results for the places of interest near the user and until the error in the obtained estimation of the user's location is less then a particular delta(x).See Fig. 9.

B. Emergency Services

Some LBS provide roadside assistance by tracking the GPS location of a vehicle and provides assistance in locating the individual who is unaware of his/her location in an emergency situation. A technique called geographic-based messaging [1] is used to warn about danger to the users within a certain area. See Fig. 10.

C. Advertising Services

In mobile commerce, the concept of location dependent advertising, or proximity triggered advertisements, is an interesting and efficient way of inviting users to shopping. Many people can be attracted by advertising e-discounts that motivate them to perform purchases inside the shopping malls.

D. Games and Social Services

In Location-based games the players move in the real world and interact with each other and with the environment depending on their locations. Some of them are as follows racing in space, Swordfish, etc. These games depend on positioning systems. Some of the other types of location based services are location-based message boards through which user can post messages within there vicinity The list of available LBSs is constantly growing, and there are many other services that have not been mentioned above.

E. Tracking Services

Tracking services provides assistance to the users in tracking the present locations of moving objects like vehicles, friends, animals in a sanctuary, for parents to track their children, old and retarded people, products, or personnel, tracking suspected criminals and specific employees, pets, monitoring taxis in an area). The flexibility in use of GPS devices plays an important role in the development of these services. The GPS devices can be worn on clothing or can be fixed in bracelets or watches.

F. Geo-Fence Services

The Geo-Fece methodology has many applications in the real world scenario. It can be used effectively by emergency services by setting up a geo fence around an area to avoid people from getting into that area. For

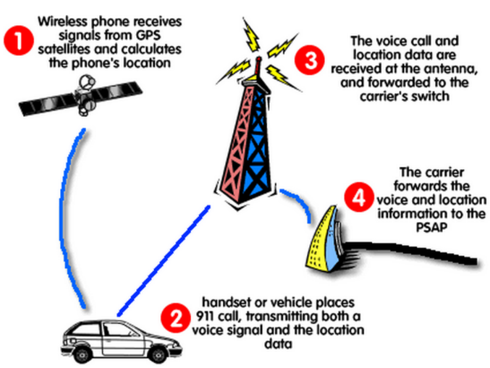

Fig. 10. Emergency Services

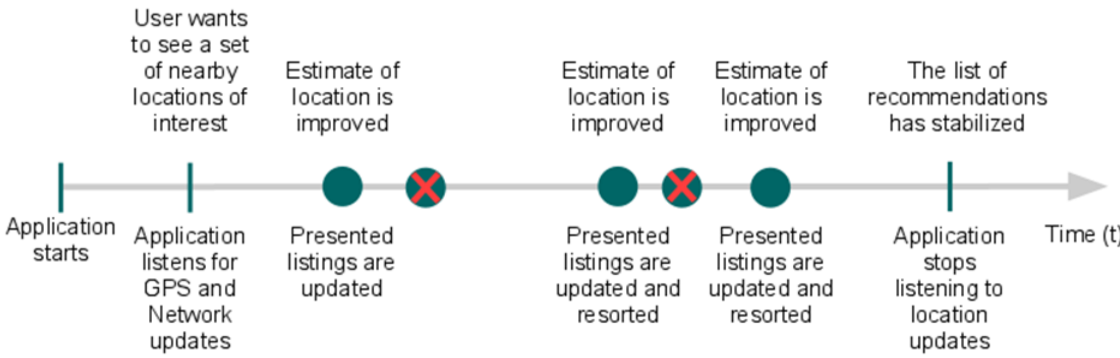

Fig. 9. Navigation Services

example, If there is a flash flood in an area or if there is a tornado approaching an area, a geo fence could be set up to warn unaware people at the earliest to avoid that route and take an alternate route. Another example could be that, If you are traveling on a freeway and if there is a huge accident ahead which has blocked the road, A geo fence could warn people about this congestion of traffic and could advise them regarding alternate routes to avoid this.

Other applications of geofence include child locations services, vehicle tracking systems and law enforcement security devices. Geofences are also used to mark the perimeter of commercial areas, residential areas, business sites and security-sensitive locations. Sometimes, immobilization equipment are attached to geofencing devices to secure equipment, doors, windows, vehicles and even people. Geofence notifications are important components of some security systems, sending notifications to alert security officials, business owners or individuals in the event of trespassing or theft of vehicles or equipment. See Fig. 11.

Tracking certain vehicles or people are also integral in the importance of geofencing technology. Vehicle routes can be established, with notifications being sent to authorized individuals or groups when these routes are not followed. Locations of children and security-sensitive people can easily be determined with this technology. Once they go outside a designated area, the proper authority figures like law enforcement officials and parents will be alerted. This kind of tracking technology is often debated as unethical and inhumane. There are more advantages than disadvantages for geofencing, though, and that is why more and more organizations are starting to use it.

VI. Challenges

A. Challenges in determining User location

- Multitude of location sources : GPS, Cell-ID, and Wi-Fi can each provide a clue to users location. Determining which to use and trust is a matter of trade-offs in accuracy, speed, and battery-efficiency [3].
- User movement : Because the user location changes, you must account for movement by re-estimating user location every so often.
- Varying accuracy : Location estimates coming from each location source are not consistent in their accuracy. A location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source.

B. Challenges in processing LBQs

1) Distributed Query Processing: While the inherent distribution of the moving objects themselves suggests a distributed approach for the query processing, many existing works reduce the problem to a centralized query processing.For centralized query processing system like DOMINO scalability, performance, and feasibility reasons, it is not convenient to assume that there exists a

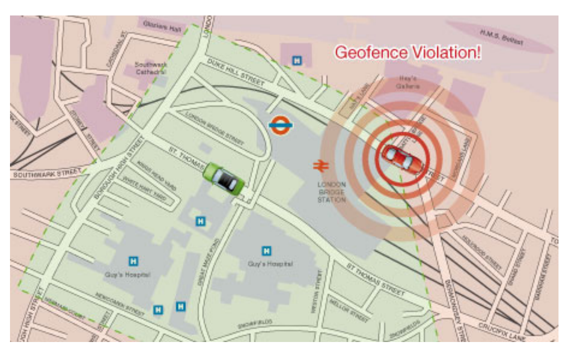

Fig. 11. Geo-Fence Application

centralized computer which is aware of the locations of all the moving objects of interest in the scenario (there could be millions of moving objects spread out over a very large geographic area).For distributed system like MobiEyes distribute the query processing, they suffer several disadvantages. For example, their approach can overload user devices and increase their wireless communications efforts, as they rely on certain availability of processing and memory capabilities of the user devices in order to process queries on them [5].

2) Catering Live Queries: This difficulties arise from the need of keeping the answer to continuous queries up to date while optimizing the wireless communications. If we consider a continuous query as just a sequence of instantaneous queries that are periodically evaluated, we will probably not be able to refresh the answer with the required frequency, due to the involved remote data access delays and initialization overhead.

VII. Related work

Most of the works that deal with some aspects of the processing of location-dependent queries assume a centralized architecture where a single computer is in charge of providing location information about all the interesting objects. In such a centralized architecture like DOMINO a single computer is in charge of providing location information about all the interesting objects.Their main concern is when to transmit results to a mobile host in order to minimize communications while providing an up to date answer to the user. However, they are not well adapted to deal with continuous queries that ask for locations of moving objects; they also assume that objects trajectories are known by the query processor.

There are some distributed approaches such as MobiEyes that rely on the processing capabilities of moving objects to detect changes in the answer presented to the user. These solutions have three main disadvantages which limit their applicability to a general environment:

- They are not completely distributed
- They impose strong requirements on the moving objects, as the objects must be able to do some query processing tasks.
- They assume that the user is only interested in the set of objects that satisfy certain conditions, but not in the geographic coordinates of these objects.

The general architecture proposed has advantages where a variety of location dependent queries can be performed in an efficient way.It also considers the scalability of query processing.

VIII. Conclusions

With increasing intensive research and development in mobile based applications and services, location of user and their interactions has evolved drastically. Wireless communications has given rise to information services that can predict, identify and adapt to the location of the mobile users. The location dependent query processing is the building block for the location based services. Location based services consist various applications which are bound to grow exponentially in coming years and would be involved in every aspect of humans. These services are helpful in many ways such as they provide mobile users with the information rapidly and accurately about the immediate surroundings that are required by a specific user in a specific context. The research challenges in this field are incomplete and in exhaustive.

Currently we are in phase where lot of Smart phone applications based on user location are coming up, hence making mobile devices smart in the way they query things around us and expose us to them and vice versa. With increasing demand and usage several other queries will be devised in future making the user experience more knowledgeable and intelligent. They would indeed be able to sense things around them which they can use to their advantage and liking. This does concern another aspect that is the privacy of the user but it is another research topic all together. The future of these location dependent services is highly sensitive for rapid development that requires promising investments, but at the same time the future of well-being of these services is significantly critical.

IX. Acknowledgment

We would like to thank Dr.Dong Xuan and our grader Adam Champion for all their suggestions, ideas and encouragement.

References

[1] A. Y. Seydim, M. H. Dunham, and V. Kumar, “Location dependent query processing,” in *Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access*, ser. MobiDe ’01. New York, NY, USA: ACM, 2001, pp. 47–53. [Online]. Available: http://doi.acm.org/10.1145/376868.376895

[2] H. Wang and R. Zimmermann, “Processing of continuous location-based range queries on moving objects in road networks,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. PP, no. 99, pp. 1 –1, September 2010.

[3] K. Tabassum, M. Hijab, and A. Damodaram, “Location dependent query processing - issues, challenges and applications,” in *Proceedings of the 2010 Second International Conference on Computer and Network Technology*, ser. ICCNT ’10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 239–243. [Online]. Available: http://dx.doi.org/10.1109/ICCNT.2010.39

[4] S. Ilarri, E. Mena, and A. Illarramendi, "Location-dependent queries in mobile contexts: Distributed processing using mobile agents," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1029–1043, August 2006. [Online]. Available: http://dx.doi.org/10.1109/TMC.2006.118

[5] N. Marsit, A. Hameurlain, Z. Mammeri, and F. Morvan, "Query processing in mobile environments: A survey and open problems," *Distributed Frameworks for Multimedia Applications, International Conference on*, vol. 0, pp. 150–157, 2005.

[6] Y. S. Ayse, H. D. Margaret, and K. Vijay, "An architecture for location dependent query processing," in *In 4th International Workshop on Mobility in Databases and Distributed Systems (MDDS01) in 12th International Conference on Database and Expert System Applications (DEXA 2001*. IEEE Computer Society, 2001, pp. 549–550.

[7] S. Ilarri, E. Mena, and A. Illarramendi, "Location-dependent query processing: Where we are and where we are heading," *ACM Comput. Surv.*, vol. 42, pp. 12:1–12:73, March 2010. [Online]. Available: http://doi.acm.org/10.1145/1670679.1670682

[8] S.-Y. Wu and K.-T. Wu, "Effective location based services with dynamic data management in mobile environments," *Wirel. Netw.*, vol. 12, pp. 369–381, May 2006. [Online]. Available: http://dx.doi.org/10.1007/s11276-005-5280-0