International Journal of Advanced Research in

Electronics, Communication & Instrumentation Engineering and Development

Volume: 1 Issue: 1 08-Nov-2013,ISSN_NO: 2347 -7210

GPS Tracking and Energy Saving System in Android Mobile Environments.

D.Venkateshwaran[1], V.Kumaran[2],
[1]U.G Students, B.E CSE, Alpha College of Engg, Chennai, T.N, India.

[1]alphacse138@yahoo.com

ABSTRACT— *We consider a set O of objects, a query point q, and a positive value r. We use dist(o,q) to denote the distance between an object o e O and the query q. A distance-based range query returns every object o e O that lies within distance r of the query location q, i.e., every object such that dist(o,q) <= r. Our main focus in this paper is on euclidean distance-based range queries. Since the search space around the query is a circle in this case, such queries are also called circular range queries. We also consider the case when distance (o,q) is the network distance between o and q (e.g., queries moving in a road network). Another variation of the range query, which we term "rectangular range query" (also called window query), returns the objects that lie within a rectangle around the query location. Distance-based range queries and rectangular range queries are inherently different and have different applications. When clear by context, we use the term range query to refer to the distance-based range queries. Due to availability of inexpensive position locators, cheap network bandwidth and the mobile device with high computation and storage capabilities, location-based services are gaining increasing popularity.*

Keywords-SWT, DWT, IDWT, LL, ImageResolution Synthesis.

1, INTRODUCTION

Consequently, continuous monitoring of user spatial queries has received we study the system continuous monitoring of moving range queries over static data objects, i.e., a mobile scenario where the queries are constantly moving whereas the data objects do not change their locations. Such scenario has many interesting applications. Consider the example of a family travelling by car. Suppose they need to reach their final destination by a certain time and only have up to 90 minutes available for lunch. They may want to continuously monitor restaurants within 10 km of their current location so that they can choose a restaurant that serves their favorite meals, and will not take more than 15 minutes to reach. As another example, a bomber plane might want to continuously monitor the enemy targets (e.g., airport, arms depot) that are within its attack range.

We next discuss two models to monitor spatial queries. Clientserver model. In this model, the clients issue queries and the central server is responsible for the computation of these queries. For example, a person walking down the street may issue a system query to his mobile service provider to continuously report the coffee shops within 1 km of the issuer's location. It may be assumed that the server processes the query in the main-memory, i.e., the data objects are stored in the main-memory along with other relevant information needed to efficiently update the results. However, such systems require that the server continuously maintains this information in the main- memory in order to provide the service. We neither require that the data objects are

stored in the main-memory nor do we maintain any query information in the main-memory. One advantage of this is that the service can be run on demand. Since the objects are stored in the secondary memory and no main-memory information is maintained, the server can go to sleep mode if there is no query. When a query arrives, the server computes the results and the safe zone, which are then sent to the client. The safe zone is an area such that the reported results are valid as long as the client (i.e., query) remains within the safe zone. Note that some computation models require queries to get registered at the server and report their locations after every t time units.

A query that leaves its safe zone sends an update request. The server updates the safe zone and the results and ends them back to the client Local computation model. In the first application mentioned above, the car may have a GPS navigation system with points of interest (e.g., restaurants) stored in its memory card. Since the navigation systems have limited main-memory and computational capacity, it may be challenging to compute the results of the range query whenever the query changes its location (the car is continuously moving). Our proposed approach returns a safe zone which guarantees that the results of the query do not change as long as the query remains within the safe zone. The safe zone is updated efficiently when the query leaves the safe zone. Our experimental results demonstrate that the overhead to compute the safe zone is small compared to the cost of the range query. This enables our framework to work effectively on the devices with limited main-memory and computation power. We next highlight some advantages of our proposed approach.

2, RELATED WORKS ON GPS NAVIGATION SYSTEM

The computation of the safe zone reduces the overall computation time because the query needs to be reevaluated only when it leaves the safe zone. Our experiments indicate that the cost of computing the safe zone is small compared to the cost of the range query. Although the shape of the safe zone may be arbitrarily complex, we can still efficiently check whether the query lies within it. If the query is based on network distance, the safe zone itself is a small network that is a subset of the original network and it has to be determined whether the query lies within the safe zone or not. For the circular range queries, we utilize the fact that the safe zone only depends on the so-called guard objects. Checking whether the query lies within the safe zone takes k distance computations, where kn is the user number of guard objects. Our experimental results demonstrate that the average number of guard the objects is around five. This makes our proposed approach applicable for the clients that have limited computational power. We also present theoretical analysis and give an upper bound on the expected number of guard objects for the queries with the diameter of the safe zone no more than constant times its expected value.

We do not require the data objects to be stored in the main-memory, which allows our approach to work on systems with limited main-memory of the user (e.g., GPS navigation systems).When an update request is received, the server computes the new safe zone and the results for the circular range of queries. After updating the results, the server only sends new information to the clients. For example, if the client was informed that an object o is within its range, the object o is not sent again in updated results if it still lies within the range. If in the future such object o ceases to be within the range, the client is informed that o is out of the range. Our experimental results demonstrate that this mobile user navigation significantly reduces the amount of data transmitted from the server to the clients. 5. In the client-server paradigm, our proposed approach does not require the server to maintain or record any information related to

the queries, yet it efficiently updates the safe zones. This enables the server to run this service on demand.

Note that some computation models require queries to get registered at the server and report their locations after every t time units. Our approach can be readily applied to such systems. In the rest of the paper, we assume a model where a query contacts the server only if it leaves the safe zone. Although there exists a safe zone-based solution for moving window queries [6], this technique is not applicable to the moving circular range queries. In Section 2, we show that it is not possible to extend this technique to the case of the distance-based range queries as the problems of monitoring moving window queries and the distance-based range queries are inherently different. We apply an aggressive approach to prune the objects/entries that cannot affect the results and/or the safe zone. Our pruning rules are tight and the performance of our solution is close to optimal.

We present an efficient and effective technique to monitor the moving circular range of user queries by adopting the concept of safe zones. We present a rigorous theoretical analysis of mobile tracking to verify the effectiveness of our user safe zone-based approach for the moving circular range queries. More specifically, we evaluate the probability that a query moves out of the safe zone within one time unit, the expected distance it travels before it leaves the safe zone, and an upper bound (which is a constant) on the expected number of guard objects for the queries with the diameter of the safe zone no more than a constant times its expected value.

Our experimental results confirm the accuracy of the presented theoretical analysis. We conduct extensive mobile experiments to show the effectiveness of our approach. We compare our algorithm with an optimal solution and a naive solution.

3, DESIGN AND IMPLEMENTATION

3.1, Outline and Architecture of the GPS

The experimental results indicate that our proposed approach is close to the optimal solution and an order of magnitude faster than the naive algorithm based on nontrivial access order and pruning rules, we present a complete frame work for answering the distance-based range queries in a road network. Experiments demonstrate that our algorithm is up to two orders of magnitude faster than a naive based algorithm the remainder of the paper is organized as follows: in Section 2, we give an overview of the related work. We introduce our framework and pruning rules for processing the moving circular queries in Section 3. In Section 4, we present our safe zone-based solution to the moving circular Range queries. Theoretical analysis is presented in Section 5. In Section 6, we present our techniques to answer moving range queries in a road network. In existing system though we have many web sites portal, and through high end mobile phones or PDAs we can get information about the needed sometimes. But these systems are not available to all and not all time. And also, this information does not reach people at the time of emergencies.

It cannot be accessed by people who don’t know to access the device. Other wise we have to make a call to some one or to make an enquiry to know about the current location. Thus it fails to provide privacy for the user. Proposed system provides more easy and efficient way to

get answer about the location and other related queries via Android. In this system we can get the answer about What, When, and how the queries are. To use this system, there is no need of using any extra devices or searching in websites. Thus it is used by people who are living in rural areas also.

3.2, LOCATION-BASED services (LBS)

This is an emerging as a major application of mobile geospatial technologies. In LBS, users with location-aware mobile devices are able to make queries about their surroundings anywhere and at any time. For example, a user can make a range query to find out all shopping centers within a certain distance of her current location, or make a kNN query to find out the k nearest gas stations. In these queries, the user has to provide the LBS server with her current location. The location-based spatial query is, thus, transformed into a region-based spatial query before being sent to the LBS server.

The LBS server then evaluates the region-based query and returns a result superset, which contains the query results for all possible location points in the cloaking region. The system submits a cloaking region R to the LBS server, which then returns the set of objects fb; c; dg that are the nearest neighbors of at least one point in R. Finally, among fb; c; dg, the system finds out the true nearest neighbor b of query location l. We conduct simulation experiments to evaluate the performance of the proposed location cloaking and query processing algorithms regarding the end-to-end system performance, MaxAccu_Cloak results in very high level system query accuracy, while MinComm_Cloak achieves a good balance between communication cost and query accuracy.

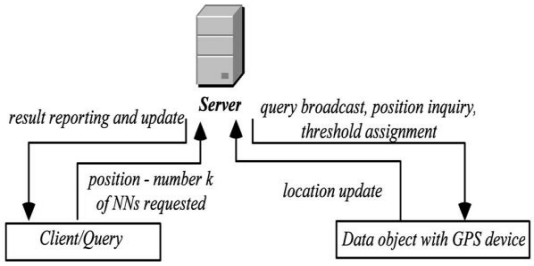

Fig .1. System Architecture

For large result supersets that require a long time to evaluate and transmit, the progressive mode achieves a shorter user-perceived response time than the bulk mode by parallelizing the query evaluation and result transmission. Assume a set of moving objects and a central server that monitors their positions over time, while processing continuous nearest neighbor queries from geographically distributed clients. In order to always report up-to-date results, the server could constantly obtain the most recent position of all objects. However, this naive solution requires the transmission of a large number of rapid data streams corresponding to location updates. Intuitively, current information is necessary only for objects that may influence some query result (i.e., they may be included in the nearest neighbor set of some client).

Motivated by this system observation, we present a threshold-based algorithm for the GPS continuous monitoring of nearest neighbors that minimizes the communication overhead

between the server and the data objects. The proposed method can be used with multiple, static, or moving queries, for any distance definition, and does not require additional knowledge (e.g., velocity vectors) besides object locations.

Whenever a clocking agent receives a query from the client it will check the query and find the safe region for the client. Safe region is calculated from the exact user location. We apply Max Cloak Algorithm, to fetch the direction of the user. If the direction of the user is towards forward then the cloaking agent will calculate the safe region with respect to the main location. For example user sends a request from Habibullah Road, user is moving towards T.Nagar, then the Safe region T.Nagar, and if the user is moving in the opposite direction then the cloaking agent will specify the safe region as Nungambakkam. After find safe region the clocking agent will send the request to the Cloud server. The Cloud server will send the result for the safe region the clocking agent receives the result from the Cloud server and then it find the nearest Location from the result and send the location to the client. The Clocking agent Manipulate the Safe region for the client and the send the query to the Cloud server.

3.3, Nearest Location for the user

The Cloud server checks the Query and Retrieve the results according to the safe region and then send the result to the clocking agent. If the user is requested for ATM Bank from Habibullah Road, first the query is sent to the Cloaking Agent. Cloaking agent will manipulate the safe region as T.Nagar, and then the query is forwarded to the Cloud Server. After getting the query result from the Cloud server, the clocking server will filter the results in accordance to the user exact location. The Cloud server will retrieve the bank information or ATM whichever is nearest to the user in accordance to T. Nagar to the cloaking agent. But the cloaking agent knows user is in Habibullah Road. So the cloaking agent will apply KNN Query Algorithm to fetch the nearest ATM or bank in accordance to Habibullah Road. So user will be receiving the exact information, as requested but then the user's Location Privacy is still maintained, because the Cloud server will update in its table as the query is from T.Nagar not from Habibullah Road. By this way we ensure Privacy in the user's location.

We neither require that the data objects are stored in the main-memory nor do we maintain any query information in the main-memory. One advantage of this is that the service can be run on demand. Since the objects are stored in the secondary memory and no main-memory information is maintained, the server can go to sleep mode if there is no query. When a query arrives, the server computes the results and the safe zone, which are then sent to the client.

The safe zone is an area such that the reported results are valid as long as the client (i.e., query) remains within the safe zone. The server updates the safe zone and the results and ends them back to the client Local computation model. In the first application mentioned above, the car may have a GPS navigation system with points of interest (e.g., restaurants) stored in its memory card.

4, INTERFACE OF GPS-ANDROID DEVICE

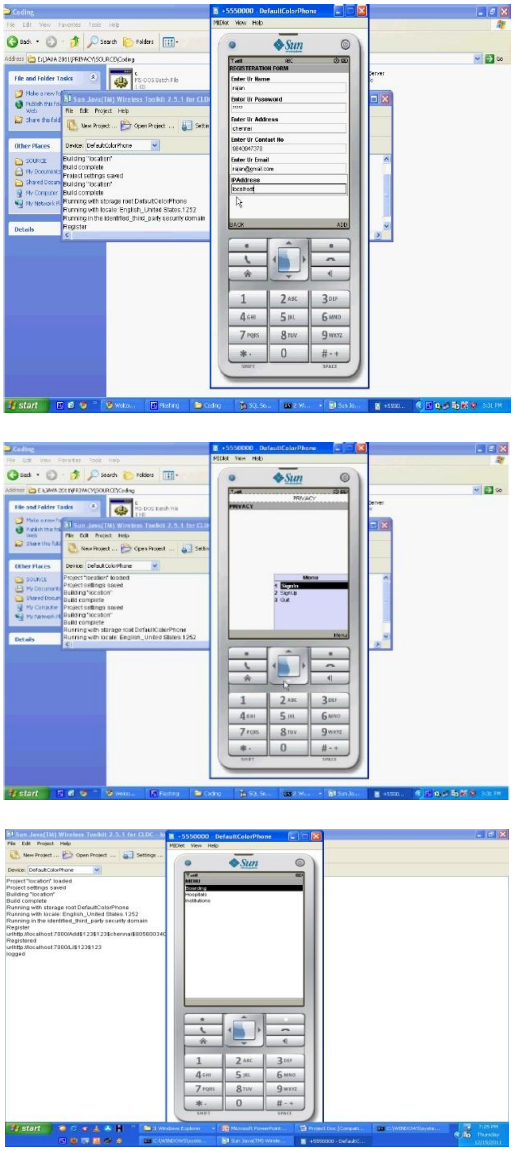

Fig. 2. Mobile Hardware Interface.

5, CONCLUSION & FUTURE WORK

The representation of cloaking regions and circular region leads to a small result superset. Experimental result shows that the user mobility aware cloaking algorithms are robust against system trace analysis attack without compromising much on query latency or communication cost.The progressive query processing algorithm generally achieves system shorter response time than the bulk algorithm. The clocking agent will get user location and then it will find the user is

moving towards the location or moving outwards the location. The current location is obtained using GPS from the mobile user. The mobile user will carry with the GPS for getting the longitude & latitude values. These values are obtained via satellite communication. So once the user sends the query to the Cloaking Agent, the Cloaking agent will get the exact location of the user via GPS values of the user.

In the Location base query system we have to register the user for his archive and future query search. Without registering a user can't access the clocking agent. For register the user should give his details such as his name, address, age, sex, ext., once a user register his details he can get useful information from the clocking server. Each user will identify by a unique username and password. If a client want to arise a query first he should be authenticated by the server for this he have to login by his user name and password after he got sign-in he can arise query to the server.

REFERENCES

[1] L. Yi-bo, X. Hong, and Z. Sen-yue,"The wrinkle generation method for facial reconstruction based on extraction of partition wrinkle line features and fractal interpolation," in *Proc. 4th Int. Conf. Image Graph.*, Aug. 22–24, 2007, pp. 933–937.

[2] Y. Rener, J. Wei, and C. Ken, "Downsample- based multiple description coding and post-processing of decoding,"in*Proc. 27th Chinese Control Conf.*, Jul. 16–18, 2008, pp. 253–256.

[3] H. Demirel, Anbarjafari, and S. Izadpanahi, "Improved motionbased localized super resolution technique using discrete wavelet transform for low resolution video enhancement," in *Proc. 17th Eur. Signal Process. Conf.*, Glasgow, Scotland, Aug. 2009, pp. 1097–1101.

[4] Y. Piao, I. Shin, and H. W. Park, "Image resolution enhancement using inter-subband correlation in wavelet domain," in *Proc. Int. Conf. Image Process.*, 2007, vol. 1, pp. I-445–448.

[5] H. Demirel and G. Anbarjafari, "Satellite image resolution enhancement using complex wavelet transform," *IEEE Geoscience and Remote Sensing Letter*, vol. 7, no. 1, pp. 123–126, Jan. 2010.

[6] C. Atkins, C. A. Bouman, and J. P. Allebach, "Optimal image scaling using image pixel classification," in *Procedingds. Int. Conf. Image Process.*, Oct. 7–10, 2001, vol. 3, pp. 864–867.

[7] W. K. Carey, D. B. Chuang, and S. S. Hemami,"Regularity-preserving decimated image interpolation," *IEEE Trans. Image Process.*, vol. 8, no. 9, pp. 1295–1297, Sep. 1999.

[8] K. Kinebuchi, D. D. Muresan, and R. G. Baraniuk, "Waveletbased statistical signal processing using hidden Markov models," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2001, vol. 3, pp. 7–11.

[9] G. Anbarjafari and H. Demirel, "Image super resolution based on interpolation of wavelet domain high frequency subbands and the spatial domain input image," *ETRI J.*, vol. 32, no. 3, pp. 390–394, Jun. 2010.

Volume: 1 Issue: 1 08-Nov-2013,ISSN_NO: 2347 -7210

BIOGRAPHY

D.Venkateshwaran, V.Kumaran, [1]U.G Students, B.E Computer Science Engineering from Alpha College of Engineering, Chennai, Tamil.Nadu, India.