Location-aware Alert System for Mobile Devices

Philip Sibley
School of Electronic Engineering
Dublin City University
Dublin, Ireland
philip.sibley2@mail.dcu.ie

Ramona Trestian and Gabriel-Miro Muntean
Performance Engineering Laboratory
School of Electronic Engineering
Dublin City University, Ireland
{ramona, munteang}@eeng.dcu.ie

***Abstract*—Being able to react fast to campaign events such as missing persons or disaster preventions, is of paramount importance. In these situations narrowing down the search area to a targeted and accurate location is imperative. Nowadays, modern mobile devices have the location awareness capabilities that can be used to determine the users Global Positioning System (GPS) coordinates. However in order to determine if a user is located within a specific area, complex floating point calculations are required. Moreover if the area is determined by a polygon, this calculation is further complicated. In this paper we propose a novel algorithm which makes use of spatial indices to determine if a mobile is located within a predefined polygon shape area. The algorithm determines the optimal length of the spatial index such as to ensure accuracy-processing time-memory trade-off. We build a prototype system, using free and open source software, to deliver alerts to mobile devices within a predetermined geographical area. The system is assessed in terms of accuracy, processing time and memory usage.**

Keywords: location-awareness; alert system; quadtree.

I. Introduction

A very popular trend nowadays is the creation of targeting campaigns for events like missing persons or disaster preventions, in specific geographical areas. The purpose of these targeting campaigns is to react quickly to events by sending notifications to mobiles located within the geographical area where the event happened. In this context, the position accuracy is imperative. With the advances in technologies, the smartphone devices have enabled location awareness [1] capabilities that can be used to determine the users Global Positioning System (GPS)[1] coordinates.

Nowadays, location awareness has become a widely used technology. Alternatives to the GPS such as Radio Frequency Identification (RFID), cellular triangulation[2], or Skyhook[3] enabled Internet-based applications to track their users' position. Moreover, their adoption is predicted to be taken up over the next three years [2]. A number of companies have produced out-of-the-box Geographic Information System (GIS) solutions which can be enveloped into existing systems in order to produce statistical reports, using their own mapping sources. These tend to be expensive in their licensing costs, and require the use of proprietary APIs. Google invested huge amount of resources in Geospatial Visualization Capabilities [3] (Google Earth and Maps) indicating that the technology will not go away any time soon. Moreover, it allows businesses to take a local copy and utilize it behind a Firewall.

[1] Introduction to GPS www.edis.ifas.ufl.edu/in653
[2] A-GPS explained in historical context www.gps-practice-and-fun.com/a-gps.html
[3] Skyhook www.skyhookwireless.com/

The majority of research in the area of GIS has been done in the possible utilization, security, optimization, and methods of presentation of GIS information. Within the field of disaster prevention and recovery, Kekumoto et al. [4] divides the type of GIS Systems into three categories: historical, public and special purpose. They suggest the use of adaptive maps to display the temporal data, and the use of meshing techniques to display some of the data, where there is little likelihood of change (road/rivers etc.). Umit et al. [5] propose a method of using GIS in conjunction with the calculated electromagnetic field emitted by the antenna of the wireless routers. They make use of 3D GIS and conclude that this is an effective method for both the planning of the network and calculating the level of pollution caused by the electromagnetic fields. Liu et al. [6] propose a GIS orientated on emergency, plotting sets of data, like: the location of the emergency, frequency of emergencies, frequency of emergencies type in that area. They conclude that GIS is a useful tool for preparing, planning and recovering from emergencies which can be used by local government.

As there is a large amount of information both to be stored, transmitted and rendered, optimization of the GIS is very important. Zhang et al. propose in [7], the use of an optimized query index on generated GeoData objects. This involves the use of the R-tree spatial index. R-tree refers to a rectangular set of information. Each update or insert to the tree requires a re-sorting of the data structure to ensure that the object is appropriately placed. Another method for representing spatial information is Quadtree [8] which has been used to represent image data for some time [9, 10]. Quadtree could be used to represent region [11] and polygon [12] information in the spatial context. Harle et al. [13] discuss the use of a combination of R-tree (for rough grain) and Quadtree (for fine grain) spatial indices for use in location aware systems. However, for R-tree requires that the device has prior knowledge of the R-tree (for comparison on server side). Samet [14] and Zhang et al. [15] proposed methods of speeding up the Quadtree processing, but this typically involves large specification machines and is not suitable for mobile devices.

This paper seeks to design and develop a prototype system which can deliver alerts to mobile devices in the most efficient way (e.g., amount of memory, processing time, accuracy, etc.). In order to do this an algorithm is defined to map the geographical area (a polygon of GPS coordinates) into a set of usable spatial indices by making use of Quadtree. Such a system could have multiple uses across many industries: (1) Geo-Marketing - targeting potential customers within a given geographical area; (2) missing person campaigns - sending photographic notifications to subscribers to assist in the location of the person; (3) emergencies/impending natural disasters notifications – transmitting to all persons within a geographic area.

II. SYSTEM ARCHITECTURE

This paper presents a framework for targeting campaigns and delivering alerts in specific geographical areas. The architecture of the system is illustrated in Fig. 1 and consists of four major components integrated using web services: (1) an admin web application to collect the GPS coordinates of the predefined polygon area where the event happened and details of the campaign from the administrator; (2) web application to display the campaign; (3) a mobile application to register and receive messages from the server; (4) batch routines to produce spatial indices from GPS coordinates (points and polygon) based on Quadtree.

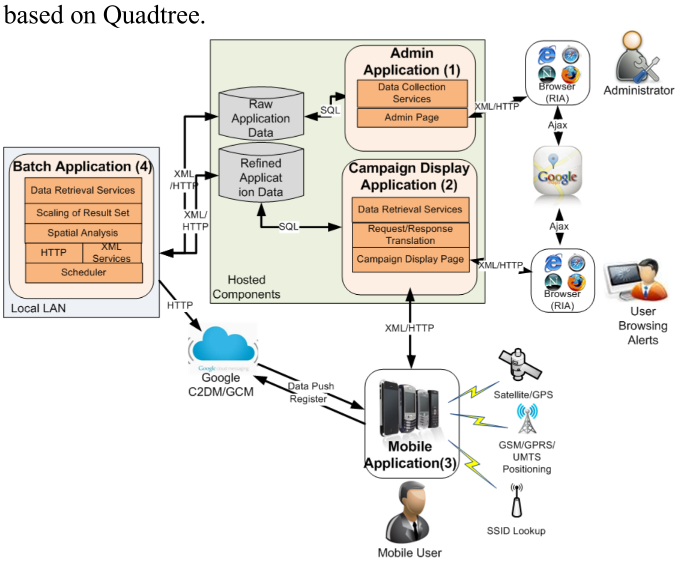


Figure 1. System Architecture

In order to create a targeted campaign, we need to: (1) register subscribers for campaigns; (2) define the campaign and the geographic area in which the campaign is to be transmitted; (3) determine if a subscriber is located within that area; (4) transmit the campaign only to the subscribers located in that specific area.

A. *Development Environment*

Eclipse IDE was used for Java development (Helios) of the batch processing. Android plug-in version 2.2 was employed for the Android development. A MySQL (version 5.0.91) was used for data persisting and hosted in the Cloud. Development licenses were obtained for both Google Maps and Cloud to Device Messaging (C2DM)[4]. C2DM is a free beta tool by Google for pushing information to registered Android devices.

The functionality of the system works as follows: Google GCM/C2DM was used to register the users for campaigns, this requires that an Android application be written to retrieve a registration ID from Google and passes this to the campaign server to sign-up the user for the receipt of messages. A simple web application using Google Maps was created to collect the GPS coordinates of a targeted polygon-shape area and the campaign information. This information is then passed to the server. At the server side an algorithm is proposed to determine if a subscriber is within the geographic area specified by the polygon. When a subscriber has been correctly identified as eligible (based on geography) they will be directed to a web page containing details of the campaign.

[4] C2DM Developers Guide, www.developers.google.com/android/c2dm/

B. *Web-based Campaign Administration System*

The Graphical User Interface (GUI) of the admin system is illustrated in Fig 2. The GUI is a HTML page hosted in the cloud which uses an XMLHTTP object to communicate with three REST Services written in Perl, which in turn access the campaign database. The admin GUI integrates the Google Maps, which allows the administrator to create a polygon on the map by clicking on the points which encompass the region and then selection 'Draw Polygon' (Fig. 2). Once the area has been defined the administrator can enter a description, E-mail address and campaign name, before selecting an image to be displayed with the campaign and clicking 'Save Campaign'.

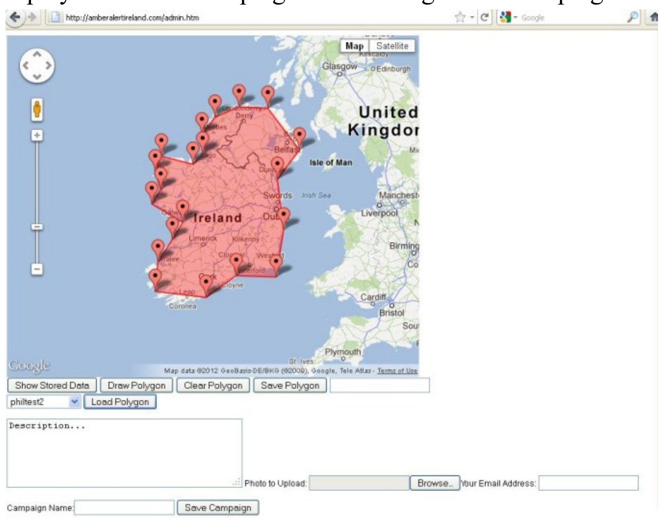


Figure 2. Admin System GUI

C. *Web Application to Display Campaigns*

The GUI of the campaign display application is illustrated in Fig. 3. There are two possible methods of accessing the Web Application to display the campaigns, either by passing in a parameter (on the URL e.g. alerts.html?campName=????) to display a specific campaign or by simply visiting the URL. The HTML is rendered to be mobile device specific, the line <meta name="viewport" content="width=device-width, user-scalable=no" /> in the HTML headers is setting this up.

D. *Android Application for message processing*

When a user initially launches the application the GPS activity is started. If the user has not been previously registered for campaign messages, a registration request is sent to GCM. The user is validated and Google returns a separate "REGISTRATION_CALLBACK_INTENT" message. On receipt of this message broadcast the receiver will locally store the registration ID and will send an addRegistration request which registers the device for campaign messages.

E. *Batch Routine*

The batch routine is the core of the overall system, that does the mapping between the 2D GPS coordinats to the polygone and the spatial indices that will be used to match the location of the subscriber to the campaign. There are two modes of the batch program. The first mode (BATCH) schedules the batch program to run every 10 seconds, and queries the database to see if any new campaigns have been added. On a new campaign the details are loaded and processed, the updated spatial indices are written to the database, the further reduced set of indices are computed and all the information is sent to all the subscribers. The second mode (KMLFILE) of the batch program allows the loading of KML files to produce the spatial indices. Collecting the information from a local KML format file rather than from the raw_polygon_data table. In this instance the program runs only once and adds the spatial indices created to the database.

Figure 3. Campaign Display GUI

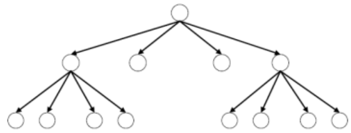
III. Spatial Indices Algorithms

There are a number of methods available to determine if a subscriber is within a geographic area. For example, on the client side, a listener could check for changes in GPS coordinate and create events which contain the current GPS location. Once this location is retrieved the coordinate could be sent to the server for determination. However, this would require the user sending their location with every campaign irrespective of whether the campaign is relevant or not. This could be perceived as a method of tracking the subscribers and would not be desirable for the user. Another possibility is to pass the coordinates of the polygon to the device, but given the limitation of the size of the message passed (4KB for GCM, 1KB for C2DM) this would be restrictive and may require data simplification. Moreover, it would also require a large amount of processing on the client side to determine whether the GPS coordinate falls within the polygon.

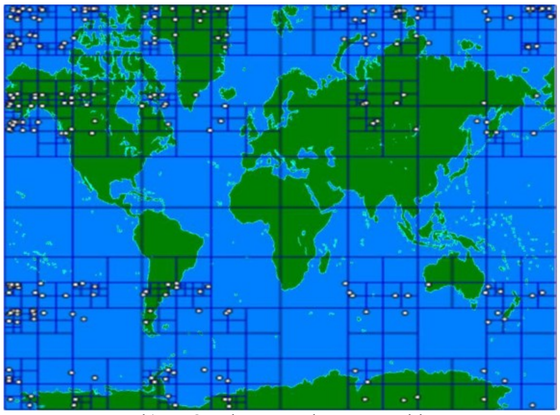
In order to overcome these drawbacks we propose the use of Quadtree to map the GPS coordinate to spatial indices on the client side. The principle behind Quadtree is illustrated in Fig. 4. Quadtree defines a data structure for which each node has four child nodes (Fig. 4a). It is used to define a discrete 2D space into a series of index able areas of varying sizes.

For a selected polygon-shape area, the GPS coordinates of the vertices of that polygon are used to compute the spatial indices. For example, if the map of the world is break down into 2^n tiles of a given size then we can perform a Quadtree on the map using the spatial indices to indicate the coverage of the polygon (Fig. 4b). The size of n can be determined based on the accuracy required for the representation.

To facilitate the generation of the campaign, the polygon describing the geographic area of the campaign must be converted to a set of spatial indices. The device's GPS coordinate have to be converted to a spatial index in the same space to validate its location within the region.



a) Quadtree Nodes



b) Quadtree overlay on World Map

Figure 4. Quadtree Principle

Using the point where the equator and prime meridian meet as the origin (0,0), we can calculate the 1st digit of the spatial index. Taking the mid-point of each sub-quadrant we can then calculate the 2nd digit and so on until we reach the desired length of spatial index. The ideal grid would by 360,000,000 x 180,000,000 Booleans, but memory constraints prevent this from being achievable. The formula for calculating the Quadtree index is given in (1), considering the range from 0 to 2^p in x and y directions (derived from [3]).

$$index = \sum_{n=1}^{p}\left(\left\lfloor \frac{MOD(x,2^n)}{2^{n-1}} \right\rfloor + 2*\left\lfloor \frac{MOD(y,2^n)}{2^{n-1}} \right\rfloor\right) \tag{1}$$

If the spatial index length is known to the device we can calculate the equivalent spatial index for the GPS coordinate of the device. It is also possible to convert a spatial index back to a GPS coordinate (2) and (3) which will give the coordinate of the top left corner of the block represented by that spatial index, where code[n] is the n^{th} character of the spatial index.

$$x = \sum_{p=1}^{p} 2^{n-1}(MOD(code[p-n+1],2)) \tag{2}$$

$$y = \sum_{p=1}^{p} 2^{n-1}\left\lfloor \frac{code[p-n+1]}{2} \right\rfloor \tag{3}$$

In order to calculate the spatial indices for the polygon associated with the campaign, two algorithms were defined and assessed.

A. *Algorithm One-Back and Forth Polygon Translation*

Algorithm one, first calculates the corner point for the translation by using (1) to determine the spatial index of the ($latitude_{max}$, $longtitude_{min}$) of all points of the polygon and then using (2) and (3) to determine the corresponding Terra Linda (TL) coordinate. This is then used to translate all the points in the polygon to the origin. An image is created using the resulting set of coordinates bounded by an image which is rounded up to the next size of the power of 2 (e.g. Image size 90x90 is rounded up to 256x256). Quadtree processing is then performed on the resulting image and a set of spatial indices is produced. The spatial indices are expanded such that every possible spatial within is represented as shown in Fig. 5. Equation (2) and (3) are used to determine the coordinate for each generated spatial index and this point is then translated back to the original place. The translated coordinate is then rendered back to a spatial index using (1). The resulting spatial indices are then processed to remove unnecessary values (e.g. 00, 01, 02, 03 can be replaced by 0).

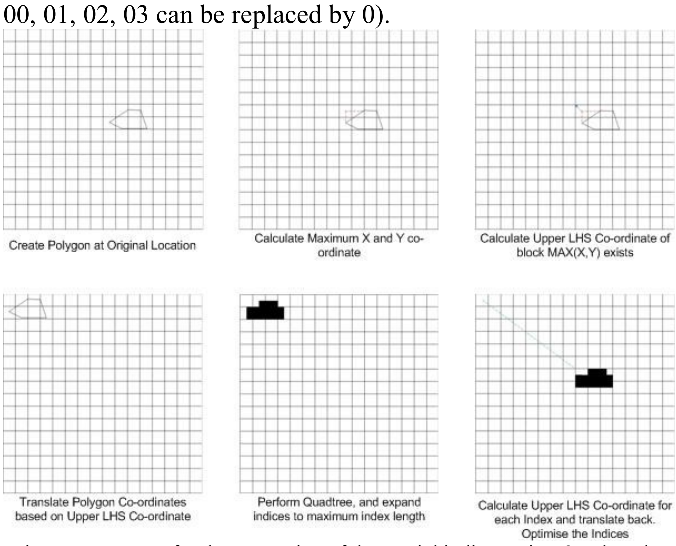


Figure 5. Process for the generation of the spatial indices using Quadtree by translating the polygon before and after the generation of indices

B. *Algorithm Two-Virtual Computation of Quadtree*

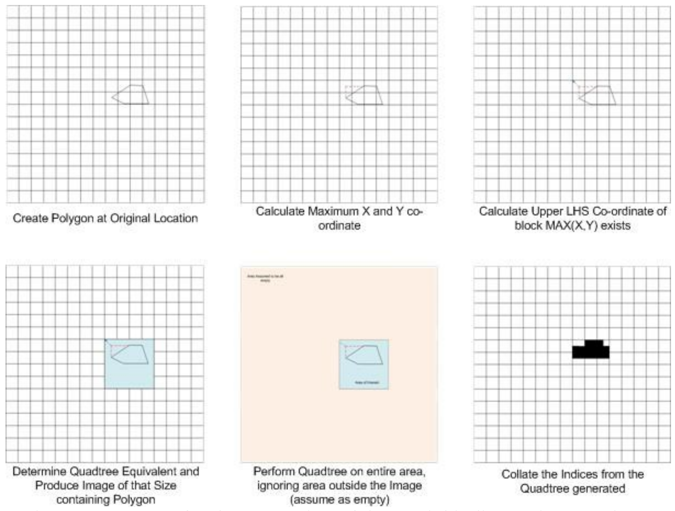


Figure 6. Process for the generation of the spatial indices using Quadtree, ignoring points which are not of interest

In Algorithm two, an image is created of the next block size up from the maximum size of the image generated, as illustrated in Fig. 6. The image is then virtually positioned within the Quadtree grid and then Quadtree is run. If a value falls outside of the area of the image, it is ignored by the Quadtree process. This reduces the amount of memory required and the processing time. The spatial indices are then derived from the Quadtree object.

Both algorithms generate a set of spatial indices to represent the polygon. The indices are further reduced to produce a discrete set of indices representing the polygon and the space around the polygon. This list of spatial indices is sent to the device to check if the device is located within the polygon. If matching, the resulting spatial index generated by the mobile device is sent to the server for a more definitive match.

IV. ANALYSIS AND RESULTS

The end to end process of generating a campaign was completed and the results are found below. The focus of this section is on the performance of the batch process in producing the spatial indices for the alert, and its analysis. Google maps were used to provide six decimals coordinates, the decimal places in GPS coordinates were varied. The data accuracies obtained are presented in Table I. This was done in the latitudinal direction, which is longer than the longitudinal given the ellipsoidal shape of the earth.

TABLE I. ACCURACY LOSS DUE BASED ON DECIMAL ACCURACY USED

Decimal Places Used	INDEX LENGTH
6	no loss
5	up to 1 meters inaccuracy
4	up to 12 meters inaccuracy
3	up to 130 meters inaccuracy
2	up to 13 km inaccuracy
1	up to 6000 km inaccuracy

The decimal place accuracy of the system is derived from the size of the grid using (4), where n is the length of the spatial index used and MSD is the index of the most significant decimal place.

$$DecimalPlaceAccuracy = MSD\left[\frac{2^{n}}{360*10^{6}}\right] \quad (4)$$

Table II shows the derived accuracy based on the spatial index lengths that we used to give an acceptable accuracy. For instance a spatial index length of 20 results in a block size of approx. 38m^2 and gives a result accurate to within 50m of the GPS coordinate when round of the decimal places is also considered.

TABLE II. CALCULATED ACCURACY BASED ON SPATIAL INDEX LENGTH

Index Length	Tile width (m)	Tile Height (m)	Decimal Accuracy	Accuracy (+/- m)
12	9783.9746	9767.5781	2	22,784
13	4891.9873	4883.7891	2	17,892
14	2445.9937	2441.8945	2	15,446
15	1222.9968	1220.9473	2	14,223
16	611.49841	610.47363	3	741
17	305.74921	305.23682	3	436
18	152.8746	152.61841	3	283
19	76.437302	76.309204	4	88
20	38.218651	38.154602	4	50

Several runs were completed using the test data encompassing several regions as a reference for the batch process. The maximum length of the spatial index was increased, which in turn reduced the size of the tile used for Quadtree and increased the precision of the match as illustrated in Table II. The results of the performance comparison of both algorithms are listed in Table III and IV as well as in Fig. 7 and 8. The results shown are gauged by the memory required by the batch process to perform the operation as well as the time taken to produce the spatial indices. These results show that the virtual Quadtree algorithm performs far better than the translation algorithm. This is explained by the additional processing that is required to produce the expanded, then translated and optimized indices.

TABLE III. PERFORMANCE RESULTS FOR ALGORITHM ONE

Map	Index Length	Duration (sec)	Memory Use (B)	Indices	Area (km^2)
Dublin	12	9	1.21MB	27	920
Ireland	12	15	4.4MB	174	84421
UK	12	82	10.73MB	3399	245000
Dublin	13	9	1.49 MB	86	920
Ireland	13	22	14.18 MB	4562	84421
UK	13	28	38.39 MB	3845	245000
Dublin	14	9	2.54 MB	73	920
Ireland	14	35	52.30 MB	4926	84421
UK	14	73	149.05 MB	5658	245000
Dublin	15	11	1.23 MB	1121	920
Ireland	15	103	206.48 MB	7545	84421
UK	15	235	584.94 MB	6287	245000
Dublin	16	38	10.30 MB	3224	920
Ireland	16	635	815.56 MB	8440	84421
UK	16	2530	234.88 MB	6567	245000

TABLE IV. PERFORMANCE RESULTS FOR ALGORITHM TWO

Map	Index Length	Duration (sec)	Memory Use (B)	Indices	Area ((km^2))
Dublin	12	7	1087776	8	920
Ireland	12	7	1109952	142	84421
UK	12	8	1150408	343	245000
Dublin	13	6	1088312	11	920
Ireland	13	7	1135192	282	84421
UK	13	8	1225224	687	245000
Dublin	14	7	1090400	24	920
Ireland	14	8	1191160	508	84421
UK	14	10	1397176	1222	245000
Dublin	15	8	1097168	61	920
Ireland	15	10	1342152	903	84421
UK	15	13	1920408	2188	245000
Dublin	16	14	1107864	95	920
Ireland	16	18	1831088	1596	84421
UK	16	32	3722416	3852	245000
Dublin	17	31	1148728	186	920
Ireland	17	46	3620384	2848	84421
UK	17	372	10539848	6909	245000
Dublin	18	101	1274160	339	920
Ireland	18	161	10291328	5051	84421
UK	18	1243	36100272	11895	245000
Dublin	19	378	1723696	661	920
Ireland	19	1078	36316720	10015	84421
Dublin	20	1499	3399512	1253	920

Comparing Algorithm two with the Oracle® Spatial results [14] which use raster loading of an image and translation of that image, the creation of the spatial indices were considerably faster. By adapting the software to allow the loading of KML files for processing and using the TIGER (Topologically Integrated Geographic Encoding and Referencing)[5] workloads as reference data sets, a comparison of the load times, as shown in Table V, with the results in [15] for some of the other currently popular Spatial Databases could be achieved.

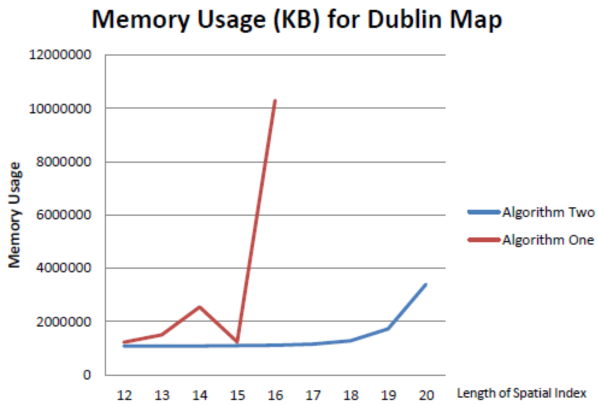


Figure 7. Comparison of Algorithms for Memory Usage (Dublin Map)

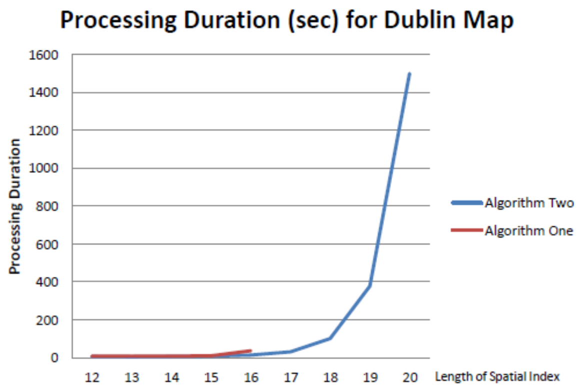


Figure 8. Comparison of Algorithm for Processing Duration (Dublin Map)

TABLE V. COMPARISON OF RESULTS WITH OTHER SPATIAL DATABASES

DataSet	MySQL	PostgreSQL	Infomix	Virtual Quadtree Algorithm For Index Length 14	15	16
arealm_merge	9.099s	2.355s	19.1s	8.339s	22.681s	min 31.458s
reawater_merge	min 56.061	1min 48.983s	20min 2.1s	43.046s	3min 54.802s	5min 10.706s

The algorithms were also repeated over several other maps including the US, Greenland and Russia, as illustrated in Table VI. In all cases a definite spike in both the memory utilization and processing time above a certain spatial index length; which also coincides with a drop in the error margin (accuracy). Tabulating these values for all the maps processed we see that this relates to the area of the map itself.

[5] "TIGER Products", http://www.cencus.gov/geo/www/tiger

TABLE VI. OPTIMAL VALUES FOR MAP TESTED

Map	Index Length	Duration (sec)	Memory Used (KB)	Accuracy (+/-m)
Dublin	16	14	1107864	95
Ireland	16	18	1831088	1596
United Kingdom	16	32	3722416	3852
United States	15	34	9961328	3577
Greenland	15	29	9886520	3068
Russia	14	81	35949176	9613

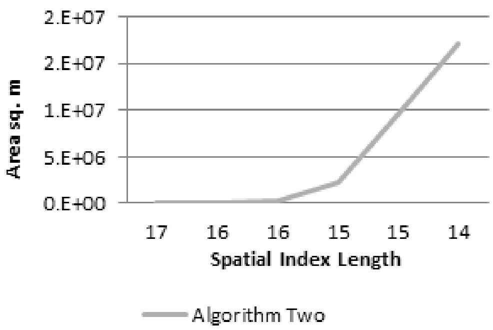


Figure 9. Graph of Area vs. Spatial Index Length to represent optimal Spatial Indices for Algorithm Two

V. CONCLUSIONS

This paper presents a location-aware alert system for mobile devices located in a specific geographical area. The system functionality is as follows: (1) first, subscribers have to register for campaigns; (2) a campaign is created and the geographic area in which the campaign will be transmitted is defined; (3) algorithm to detect if a subscriber is located within that area is proposed; (4) finally the campaign is transmitted to the subscribers located in that specific area only. In order to facilitate the generation of the campaign, the polygon describing the geographic area of the campaign and the device's GPS coordinates must be converted to a set of spatial indices. For this purpose the proposed algorithm makes use of Quadtree. The performance of the overall system was analyzed in terms of memory consumption, processing time, and accuracy. The results show the efficiency of using Quadtree for the generation of spatial indices.

ACKNOWLEDGEMENTS

This work was supported by Irish Research Council (www.research.ie).

REFERENCES

[1] P. DeBeasi, "Real-Time Location: Where are we now?" Burton Research, June 2008.

[2] W. Clark, "Context-Aware Computing Innovation Key Initiative Overview", Gartner Research, July 2011.

[3] T. Austin, A. Bitterer, "Google Improves Geospatial Visualisation Capabilities for Business", Gartner Research, August 2007.

[4] S. Kakumoto, Y. Kosugi, M. Hatayama, and H. Kameda, "Development of spatial temporal geographic information system and risk adaptive regional management information system - toward development of GIS based on Asian culture for disaster prevention," in *Proceedings of the 41st SICE Annual Conference*, vol. 1, pp. 352 – 357, 2002.

[5] M-U. Gumusay, A. Sen, U. Bulucu, A. Kavas, "Electromagnetic Coverage Calculation in GIS", *International Symposium on Mobile Mapping Technology*, Padova, Italy, 2007.

[6] L. Xiao-dong, Z. Yi, S. Li-jian, and L. Chen, "The Study of Geographic Information Service by Emergency-Oriented," in *International Conference on Management and Service Science (MASS)*, 2010.

[7] W. Zhang, X. Yang, W. Wu, and G. Xiang, "An Optimized Query Index Method Based on R-Tree," in *Proceedings of the 2011 Fourth International Joint Conference on Computational Sciences and Optimization*, Washington, DC, USA, pp. 1007–1010, 2011.

[8] H. Samet, "Data structures for quadtree approximation and compression", *Communications of the ACM*, 28(9):973-993, 1985.

[9] M. A. Oliver and N. E. Wiseman, "Operations on Quadtree Leaves and Related Image Areas," *The Computer Journal*, vol. 26, no. 4, pp. 375–380, 1983.

[10] W. I. Grosky and R. Jain, "Optimal Quadtrees for Image Segments," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 5, no. 1, pp. 77–83, 1983.

[11] H. Samet, A. Rosenfeld, C. Shaffer, R. Webber, "Quadtree Region Representation in Cartography: Experimental Results," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 13, pp. 1148–1154, 1983.

[12] H. Samet and R. E. Webber, "Storing a collection of polygons using quadtrees," *ACM Trans. Graph*., vol. 4, no. 3, pp. 182–222, 1985.

[13] R. K. Harle, "Spatial Indexing for Location-Aware Systems," *in Fourth Annual International Conference on Mobile and Ubiquitous Systems: Networking Services (MobiQuitous),* pp. 1 –8, 2007.

[14] S. Ravada, "New Performance Enhancements in Oracle Spatial" [Available Online], www.oracle.com. 2012. 23 May 2012

[15] S. Ray, B. Simion, and A. D. Brown, "Jackpine: A benchmark to evaluate spatial database performance," in *IEEE 27th International Conference on Data Engineering (ICDE)*, pp. 1139 –1150, 2011.