GPS Location Alert System

Saravana Kumar J[1], Veeramani .R[2]
[1]*(PG Student, Department of Computer Science and Engineering, SRM University, Ramapuram, India)*
[2]*(Department of Information Technology, SRM University, Ramapuram, India)*

Abstract : *Traditionally alerts are based on time. But there are many situations the alerts are based on your current location and not based on time. It is difficult to find the persons in a particular location where some disaster is going to happen. In this situation fixing the search area and accurate location is difficult. In this modern era we have mobile devices capable of finding the user's current location using the Global Positioning Systems (GPS). In this paper we use our own algorithm "Extended Polygon Match Algorithm using Quadtree" to find whether a mobile is within a defined polygon shaped area using their GPS coordinates. This algorithm determines the spatial indices of the polygon region and mobile user's Current location. We build a prototype system using Android software to send alerts to mobile devices within the defined geographical area. This mobile application will alert you if you are in that defined geographical region and you can know the distance between you from the defined reminder location.*

Keywords: *Location-alert, GPS, Quadtree, Spatial, Android*

I. INTRODUCTION

The purpose of the Paper is to provide the system to find the missing persons or to guard the people located in the defined geographical area where the disaster is going to happened. Previous System uses Skyhook or cellular triangulation to find the users' location. Everyday technology is growing, so we are in a situation to update the system [1]. By using current technologies in smart phones like GPS [8], we can find more accurate location of the mobile user in a faster manner. In Existing system, Geographical Information System (GIS) is used to produce reports using their own maps. There are some disadvantages in existing system like expensive licensing cost, more memory needed for rendering, not getting very accurate and optimized results. The smart phones are equipped with GPS technology. Using their GPS coordinates (Longitude and Latitude) we can find their location. In our paper we propose to develop a new prototype which delivers alerts to mobile devices which are in the defined geographical area. To achieve this we define an algorithm to locate the geographical region using polygon shaped area which covers the entire defined geographical region [2]. We use Quadtree data structure to store the polygon area in the database [3]. When a disaster like earthquake or tsunami, it is difficult to define the exact geographical area which is going to be affected. So we extend the defined region such a way that the information passed to all the persons who are closer to the defined geographical area. We set a limit for the extension according to the seriousness of the disaster. The advantages of our system are less expensive, less amount of memory needed, quicker processing time and more accurate than earlier system. The Cost is reduced since we use most of open source software like Android and MySQL [7] and we use GCM(Google Cloud messaging for Android) [9] which is a service provided by Google that allows us to send or receive the message between our server and mobile devices equipped with android. This application can be implemented in indentifying the Missing Persons Location and Emergencies / Natural disaster notifications.

II. SYSTEM ARCHITECTURE

This Paper presents a framework for defining geographical region and delivering alerts in that geographical region. The architecture of the system is given in Fig. 1. It consists of four major components integrated (1) An admin application to get the GPS coordinates of the defined polygon area where the disaster may occur; (2) An alert display web application to find the target mobiles in the defined region and push message; (3) A mobile application used to register mobile with server; (4) Batch process routines to find GPS coordinates of mobile device and defined polygon region using Quadtree.

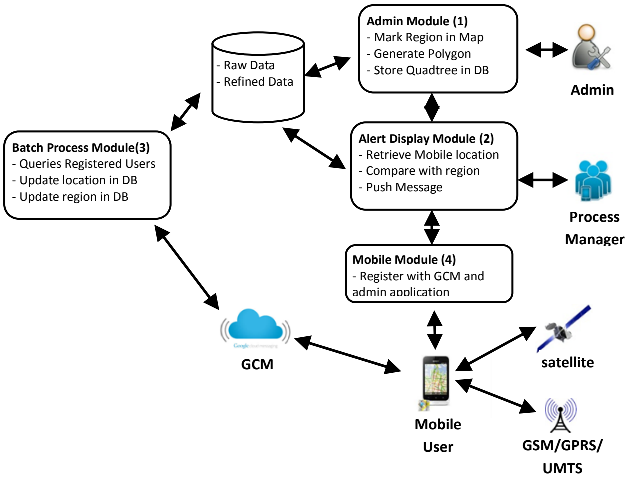

Fig. 1: System Architecture

Development Environment

Eclipse 4.3 IDE is used for Java development of the batch processing. Android plug-in version 4.0 is use for the Android development. A MySQL (version 5.2.3) is used for data storage. Development licenses are obtained for both Google Maps and GCM. GCM is a service that allows you to send data from your server to your users' Android-powered device, and also to receive messages from devices on the same connection [10]. GCM is used to register the mobile users with our application; this requires an application developed in Android to retrieve a registration ID from Google and passes this to the server. A web application using Google Maps is 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 webpage containing details of the campaign.

2.1 Admin Module

This module is used to mark targeted region in map, generate polygon region for marked area, find Spatial co-ordinates [1] for the corner points of the polygon, store the region in a data structure called Quadtree, and using Quadtree data structure store the co-ordinates in spatial database (MySQL)[6]. The spatial data are stored in database as raw data.

2.1.1 Setting Polygon Region

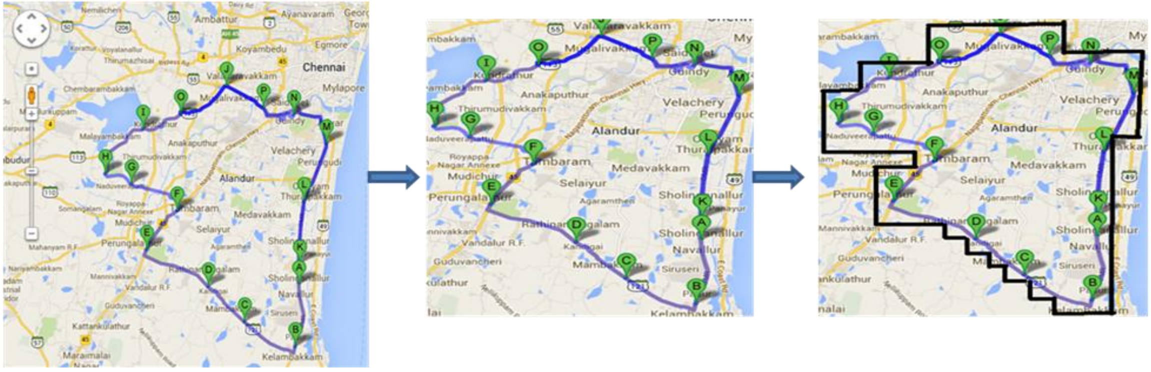

Fig. 2(a) Fig. 2(b) Fig. 2(c)

Fig. 2(a) Part of Map Contains the target region, 2(b) Minimum Square Contains the Target Region and 2(c) Polygon Region Contains the Target Region.

2.2 Alert Display Module

This module is used to retrieve all Registered mobile users from the database, find the current location of the registered users, get their longitude, latitude of their location, convert longitude and latitude into spatial co-ordinates, compare the spatial co-ordinates with the polygon region, if it is within the region push a alert message to that user’s mobile and if the Mobile user requests for traffic information push the message contains traffic information to the mobile user

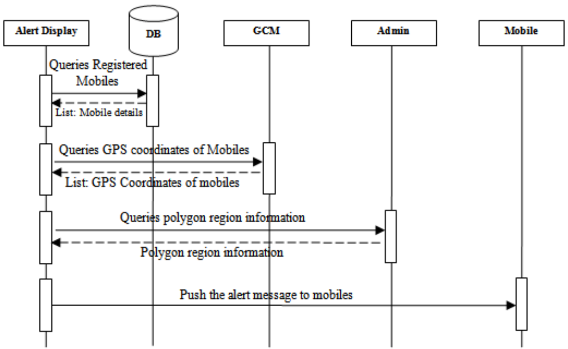
Alert Display
DB
GCM
Admin
Mobile
Queries Registered Mobiles
List: Mobile details
Queries GPS coordinates of Mobiles
List: GPS Coordinates of mobiles
Queries polygon region information
Polygon region information
Push the alert message to mobiles

Fig.3. Sequence flow of Alert Display module

2.3 Batch Process Module

It is an automated module which runs automatically at regular interval fixed by admin (For Example 30 Seconds). It Queries the database for registered mobile users and retrieves their current location from GCM. It updates the registered mobile users' current location in the database. It checks the changes in polygon region and updates it in the database. The Raw data of polygon region are processed in this module and stored as refined data that is in Quadtree tree data structure [5].

2.4 Mobile Registration Module

This module provides the registration screen to the android mobile users, Registration request is sent to Admin, Admin register the mobile user with GCM (Google Cloud Messaging) and the details are stored in the database, User Id and Password will be created for mobile user.

III. ALGORITHMS AND IMPLEMENTATION

In this paper we first mark the target area using Google Maps. Then the region is fitted into a square region that fits the marked region. The entire square region co-ordinate is stored in the data structure Quadtree using our algorithm Extended Polygon Match Algorithm using Quadtree.

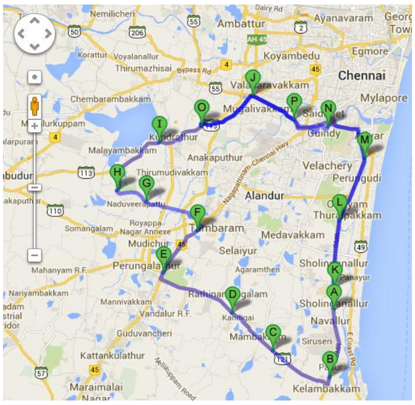

Fig.4. Marked target region on the map

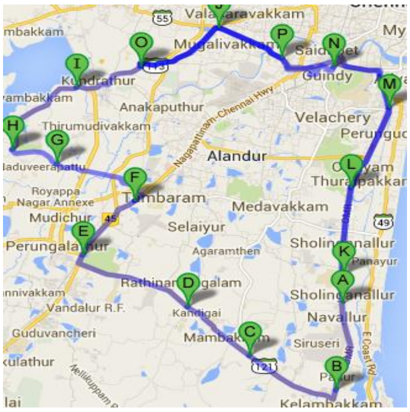

Fig.5. Minimum square region that fits entire marked region

Find the GPS Co-ordinates Longitude and Latitude of the top left corner and bottom right corner of the square region. Using the below mention formula we can find the distance between the two opposite corners.

dlon = lon2 - lon1 ; (1)
dlat = lat2 - lat1 ; (2)
a = (sin(dlat/2))^2 + cos(lat1) * cos(lat2) * (sin(dlon/2))^2 ; (3)
c = 2 * atan2(sqrt(a), sqrt(1-a)) ; (4)
d = R * c (where R is the radius of the Earth 6373 km) (5)

Using Initial GPS coordinates of Square region and distance d , we can further divide the square region in to four quadrants until it reaches minimum defined size of square.

A quadtree is a tree data structure in which each internal node has exactly four children. Quadtrees are most often used to partition a two-dimensional space by recursively subdividing it into four quadrants or regions [11]. Sample Quadtree Structure depicted in below figure 5. In below figure Grey colored box represents partially filled region, white represents the empty region and wherein black represents fully filled region [4]

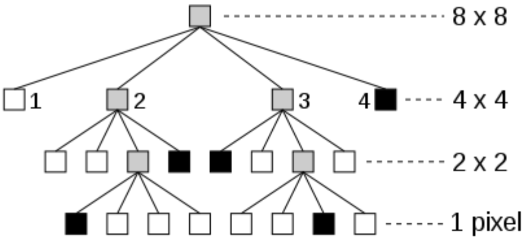

Fig.6. Structure of quadtree

Let us consider overall square region as root node Split the overall Square in to four equal quadrature “NW”, “NE”, “SW” and “SE”. Sub divide the regions until the minimum pixel range reached or the division is enough to represent the full marked region.

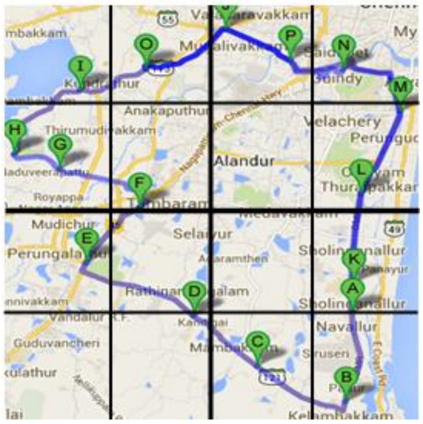

Fig.7. Splitting the square regions into quadrants

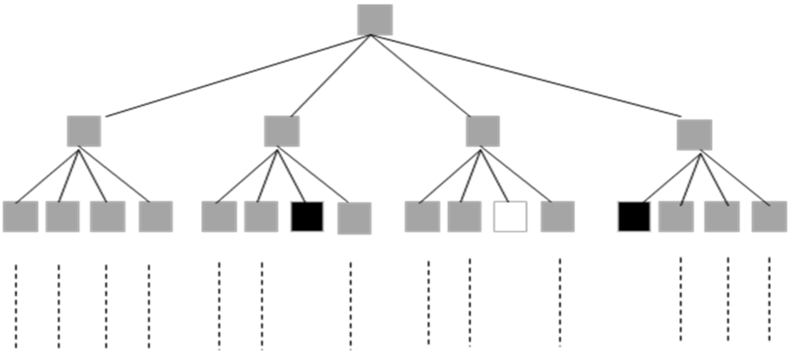

Fig. 8. Representation of Quadtree for our region

Here we extend the region at leaf node, which is partially filled or fully filled, to a fixed value ‘e’ so that the alert message will reach the registered mobile users who are not in defined geographical region but they are nearby the region up to the distance ‘e’.

```
/* Pseudo code for quadtree creation and comparison of mobile location with quadtree */
structure coordinate
/* define the structure to store the GPS coordinates Longitude and latitude */
begin
  type double long;
  type double lati;
end

/* Global variables
  type coordinate quadtop[4], quadbottom[4];

/* to find the distance between two points in the map */
function type double calculate_distance(type coordinate p1, type coordinate p2)
/* p1 is the top left corner of the square and p2 is the bottom right corner of the square */
begin
  type double distlong ← p2.long – p1.long
  type double distlati ← p2.long – p1.long
  type double a ← $\left[\sin\left(\frac{distlati}{2}\right)\right]^2 + \cos(p1.distlati) * \cos(p2.distlati) * \left[\sin\left(\frac{distlong}{2}\right)\right]^2$
  type double c ←   2 * atan2 (sqrt(a), sqrt(1-a))
  double d ← R * c
/* R is the radius of the earth 6373 km */
  return d
end

/* compare the location of mobile with polygon region */
function type boolean compare_location( type coordinate mob, type coordinate p1, type coordinate p2)
begin
  if (mob.long >= p1.long and mob.lati<= p1.lati and mob.long<=p2.long and mob.lati>=p2.lati) then
  begin
    return true;
  end
  else
  begin
    return false;
  end
 end

/* function to split the quadtree in four quadrant */
function split_quadtree( type coordinate p1, type coordinate p2)
begin
  type coordinate quadtop[4], quadbottom[4];
  quadtop[0].long ←p1.long;
  quadtop[1].long ←(p1.long + p2.long)/2;
  quadtop[2].long ←p1.long;
  quadtop[3].long ←(p1.long + p2.long)/2;
  quadtop[0].lati ←p1.lati;
  quadtop[1].lati ←p1.lati;
  quadtop[2].lati ←(p1.lati + p2.lati)/2;
  quadtop[3].lati ←(p1.lati + p2.lati)/2;
  quadbottom[0].long ←(p1.long + p2.long)/2;
  quadbottom[1].long ←p2.long;
  quadbottom[2].long ←(p1.long + p2.long)/2;
  quadbottom[3].long ←p2.long;
  quadbottom[0].lati ←(p1.lati + p2.lati)/2;
  quadbottom[1].lati ←(p1.lati + p2.lati)/2;
  quadbottom[2].lati ←p2.lati;
  quadbottom[3].lati ←p2.lati;
end

/* function to create additional square region to push message to neighbor mobile of defined region */
function create_additional_square(type coordinate p1, type coordinate p2)
begin
  if node's NW is empty create NW square with ADDITIONAL_COVERATE_DISTANCE;
  if node's NE is empty create NE square with ADDITIONAL_COVERATE_DISTANCE;
  if node's SW is empty create SW square with ADDITIONAL_COVERATE_DISTANCE;
  if node's SE is empty create SE square with ADDITIONAL_COVERATE_DISTANCE;
end

/* main function for comparison process*/
function main
begin
  define SQURE_MIN_SIZE ← 1;
  define ADDITIONAL_COVERAGE_DIST ← 1;
  type coordinate mob ←retrieve mobile's GPS coordinates;
  type coordinate initialtop, initialbottom;
  if (compare_location(mob, initialtop, initialbottom) == true) then
  begin
    if ( square_is_partialy_covered and calculate_distance(initaltop,initalbottom)<SQURE_MIN_SIZE)
    begin
      while(true)  /* infinite loop */
      begin
        split_quadtree(initialtop, intialbottom)
        split until either we reach minimum square size reached or fully covered square or empty square area
        if( the square is either in minimum square size or fully covered square or empty square area)
        begin
          if (the square is empty) no need to push the message and the mobile is not in the defined region
          if (the square is fully covered)
          begin
            push the message to mobile
            call create_additional_square
          end
          if (the square is in min size)
          begin
            push the message to mobile
            call create_additional_square
          end
      end
    end
  end
end
```

IV. CONCLUSION

Prevention and preparedness of Natural Disaster is the main goal of our paper. But the area which is going to be affected in a natural disaster cannot be defined exactly. In this paper we propose and implement the solution for this problem by extending the search area to certain limit according to the seriousness of the disaster. Our Paper provides the solution to find the location of mobile users in a defined region in a faster and efficient manner. But there are some limitations in our system, that we can use it only in a Android Powered and GPS enabled mobile device. To overcome these limitations we have planned to implement this concept in all the mobiles which is equipped with GPS in our future work.

REFERENCES

Journal Papers:

[1] 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.

[2] Gang Fu, Hongrui Zhao, Cong Li and Limei Shi, Segmentation for High-Resolution Optical Remote Sensing Imagery Using Improved Quadtree and Region Adjacency Graph Technique, *MDPI International Journal, Remote Sens. 2013, 5, 3259-3279; doi:10.3390/rs5073259*

[3] Hanan Samet, University Of Maryland And Robert E. Webber, Rutgers University, *Storing A Collection Of Polygons Using Quadtrees*, ACM Transactions On Graphics, Vol. 4, No. 3, July 1985, Pages 182-222.

[4] Abdullah Çavuşoğlu1, Baha Şen1,*, Caner Özcan2 and Salih Görgünoğlu, Cropped quad-tree based solid object colouring with cuda, *Mathematical and Computational Applications*, Vol. 18, No. 3, pp. 301-312, 2013

[5] Jean-Marc Laferté, Patrick Pérez, and Fabrice Heitz, Discrete Markov Image Modeling and Inference on the Quadtree, IEEE Transactions on Image Processing, VOL. 9, NO. 3, MARCH 2000

[6] Cyrus Shahabi, *Introduction to Spatial Database Systems*, Ralf Hart Hartmut Guting's VLDB Journal v3, n4, October 1994

Books:

[7] Evangelos Katsikaros , Towards the universal spatial data model based indexing and its implementation in MySQL , *Kongens Lyngby 2012*

[8] Jorge R. Rey, An Introduction to GPS, *University of Florida, IFAS Extension*

[9] Rajkumar Buyy, James Broberg, Andrzej Goscinski, "Cloud Computing Principles and Paradigms", A John Wiley & Sons, Inc., Publications, 2011

Web Reference:

[10] http://developer.android.com/google/gcm/index.html

[11] http://en.wikipedia.org/wiki/Quadtree

BIOGRAPHY

Author 1: **J.Saravanakumar** , received B.Sc. Mathematics degree from Madurai Kamarajar University, Madurai, Tamilnadu, India, in 1998, M.C.A., from University of Madras, Chennai, Tamilnadu, India, in 2004 and doing M.Tech., Computer Science and Engineering in SRM University, Chennai, Tamilnadu, India. He is working as an Assistant Professor in Department of Computer applications, Maamallan Institute of Technology, Sriperumpudhur, Tamil nadu, India. His areas of interest in research are Mobile Computing and Cloud Computing.

Author 2: **R. Veeramani**, received B.E degree in Computer Science and Engineering from University of Madras, Chennai, Tamil Nadu, India, in 1998 and M.E in Computer Science and Engineering from Annamalai University, Chidambaram, Tamil Nadu, India in 2006. He is currently working as Assistant Professor in SRM University, Chennai, Tamil Nadu, India. He has more than 10 years experience in teaching and development of e-commerce applications. His research interests are in Data Mining, Distributed Computing, Cloud Computing, Parallel System and Programming languages. He is member of Indian Science Congress Association.