GIS Based Navigation System of Dhaka City

Mufti Shahriar Haque

ID# 02101009

Tapan Biswas

ID#01201003

Department of Computer Science and Engineering

BRAC University, Dhaka, Bangladesh

January 2007

DECLARATION

I Mufti Shahriar Haque, student of computer science and engineering department, BRAC University represent my thesis work "GIS Based Navigation System of Dhaka City" as requirement of completion of bachelor degree. This thesis project was performed under supervision of Mr. Shadid Haque, Lecturer, BRAC University, Dhaka, Bangladesh.

This is to declare that the thesis work was done by me and it has not been submitted before. The help that was taken from internet and books was mentioned at references.

Signature of   Signature of

Supervisor   Author

---------------------   ----------------------

Shadid Haque   Mufti Shahriar Haque

ACKNOWLEDGEMENT

I want to remind Almighty most gracious, most merciful to finish my thesis work successfully and in time.

With a heart full of appreciation I desire to mention my thesis supervisor's name Mr. Shadid Haque for his brilliant supervision in every step of performing this work. He not only inspired me to work on this topic but also helped supplying me relevant resources. Within his rigid schedule he made up time for me. He suggested various solutions that I could not solve myself. My knowledge outside the topic was enriched by reviewing his supplied documents. The pleasure was all mine working with him.

I would also like to thank Tapan Biswas for helping me to collect data. It would take a lot of precious time if I did the same work.

ABSTRACT

This thesis project's objective is to implement a navigation system for Dhaka city. The system uses real road map of Dhaka city. It shows users current position, shortest path based on distance to the expected destination and destination point on the map. It also notifies users when they are off road or should take a turn. The system alerts users before a certain distance from the point to take turn. Users' current position is taken form GPS device and is shown on GPS activation. Users only input their destination and the path is dynamically updated with the change of their position. The map is properly scaled and coordinates used for calculation are taken from Google earth and other points on the map perfectly coincide with coordinate value from Google earth.

TABLE OF CONTENT

CHAPTER I: INTRODUCTION

In Dhaka the roads were not designed perfectly as it is in the modern cities. Roads are scattered as well as houses are numbered randomly. So it is very tough to find a location or house for a new comer as well as an inhabitant of this city too. Moreover many of the roads are blocked due to heavy traffic jam. As a result travelers suffers from a wasting their precious time on the road. So they need an assistant to guide them to proper way.

The proper way may be based any on of shortest distance, traffic condition of roads, users point of interest(POI), road quality, road safety, users preferred road or number of red lights. This "GPS Navigation System of Dhaka city" can navigate the traveler along the shortest path based on distance. User's current position is located using GPS device connected with computer. For the shake of perfection Graphical Information System (GIS) based map of Dhaka city has been used.

Navigation system exists in many countries but a real time navigation system does not exist for our country. But some research work has been done before but no one works as perfect assistant on road.

To develop this project programming language Java is used. Standard Widget Toolkit (SWT), a special library in Eclipse (Version: 3.2) has been used to design Graphical User Interface (GUI).

CHAPTER II: SIMILAR SYSTEM ON OTHER COUNTRIES

In USA and Canada a system called CoPilot is used. It has a lot of features like it can search by phone number, POI. It shows if the user has to pay toll, RV Routing with 12'6" & 13' height restriction options. It also contains voice notification and many other options.

There is also software called Gpsylon. It has all the features of CoPilot along with capability of adding map of new places. Layers can be added upon the map such as position layer, shape layer, location layer and scale layer.

CHAPTER III: DATA COLLECTION

Data collection is the most difficult and tedious job in this project. As road can be compared to an undirected graph (as most of the roads are two way in our country) so nodes are join of roads and edges between nodes represent roads. As Dhaka is a big city and it has a lots of roads and nodes so I have worked with a small area. Using Google earth within that area I have marked the nodes by place marks and gave them unique identifier. Later the coordinate (longitude and latitude) of each node and its name are taken in a text file along with its identifier. The data in first file is collected in following format:

1	lat=23.7591133528, lon=90.3897237302	FarmGate(Rokea Sarany)
2	lat=23.7584523581, lon=90.3900125877	Farm Gate (Over Bridge)
3	lat=23.7578185426, lon=90.3902259419	Farm Gate (Cinema Hall)
4	lat=23.749917827, lon=90.3932022645	Karwan Bazar
5	lat=23.7509767008, lon=90.3870802514	Green Road

Figure: Part of Nodes.txt file

There are total 154 nodes taken.

After that, pairs of nodes who have a connection (Road) between them are taken in another text file. The data in second file is collected in following format:

```
1        2
1        9
1       10
2        3
2      151
3        4
3        5
```

Figure: Part of Roads.txt file

Two numbers in a row means they are connected.

The map of Dhaka city used has been found by searching in Google.

CHAPTER IV: SYSTEM DESIGN

It is the most important phase of performing the project work. In this phase the shortest path problem is matched with a graph problem. Database design, system engine design and GUI design are discussed in following subsections.

4.1 DATABASE DESIGN

There are two entities. One is Node and the other one is Road.

Node has several attributes that are: id (which is primary key), longitude, latitude, pixel value on x-axis, pixel value on y-axis and its name.

A road is constructed by two nodes. So we express a road by a straight lines equation. We know,

Ax+By+C = 0 is the equation of a straight line.

So to obtain value of A, B and C we use the formula:

If P1(x1, y1) and P2(x2, y2) are two points given then the equation of straight line between them will be,

$$y-y1/y1-y2 = x-x1/x1-x2$$

=> (y-y1)(x1-x2) = (x-x1)(y1-y2)

=> y(x1-x2) - y1(x1-x2) = x(y1-y2) − x1(y1-y2)

=> -x(y1-y2) + y(x1-x2) + x1(y1-y2) − y1(x1-x2) = 0

So we get

A = - (y1-y2)

B = (x1-x2)

C= x1(y1-y2) – y1(x1-x2),

Where

x1 = longitude of point 1

y1 = latitude of point 1

x2 = longitude of point 2

y2 = latitude of point 2

Another attribute of a road is its length which is necessary to calculate shortest path. To calculate distance we use this formula:

```
DISTANCE=3963.1*ARCOS[SIN(A)SIN(C)+COS(A)COS(C)COS(B-D)];[Distance in Mile]

DISTANCE= DISTANCE*1.609*1000; [Converting into meter]



Where,

A = LAT1, B = LONG1

C = LAT2, D = LONG2 (all converted to radians: degree/57.29577951)
```
Distance Calculation

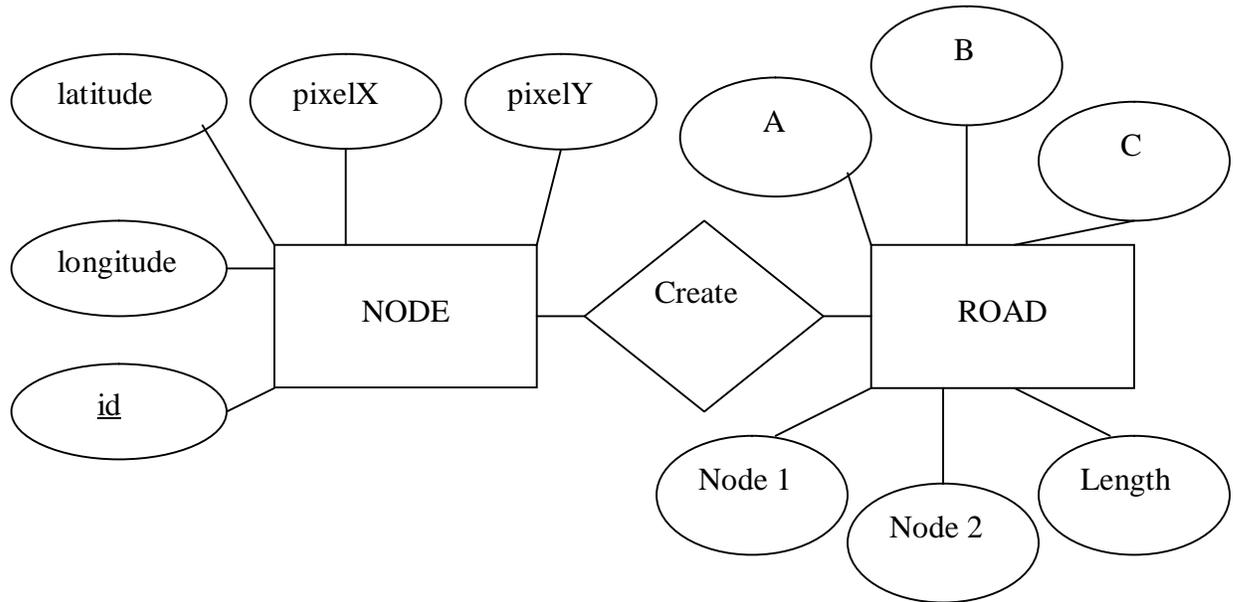So a road has 6 attributes which are: node1, node2, A, B, C and length.

Diagram: Entity Relationship Model

4.2 SYSTEM ENGINE DESIGN

System engine is made of Dijkstra algorithm. The system takes node and road information as input. Based on current position and destination it returns the shortest path. As the project is considered as graph problem so the system converts node data into a two dimensional matrix. In fact matrix original size is 155x155 as we have 154 nodes. One place in the matrix is wasted because there is no node identified by 0. This matrix contains distances between all nodes that are connected. When users inputs their current position first of all it need to be calculated that in which road they are on. To do that I have used the process below:

Find the intersection point from the source to the equation of every road. The formula is:

$$A2 = B1; \qquad B2 = -A1; \qquad C2 = -A2\text{*longitude-}B2\text{*latitude};$$

$$x = (B1\text{*}C2-B2\text{*}C1) / (A1\text{*}B2-A2\text{*}B1);$$

$$y = (C1\text{*}A2-C2\text{*}A1) / (A1\text{*}B2-A2\text{*}B1);$$

Where,

A1, B1, C1 represent the road's equation,

A2, B2, C2 represent

x, y represent intersection value

Then if the intersection point is on the road then the distance from the source and intersection point is calculated. The road that has the intersection point on it and has minimum normal distance from source is returned to Dijkstra algorithm for shortest path calculation.

4.3 GUI DESIGN

When the system engine is done the next job is to design GUI. The source, path and destination are to be shown graphically. Notification also needs to be graphical.
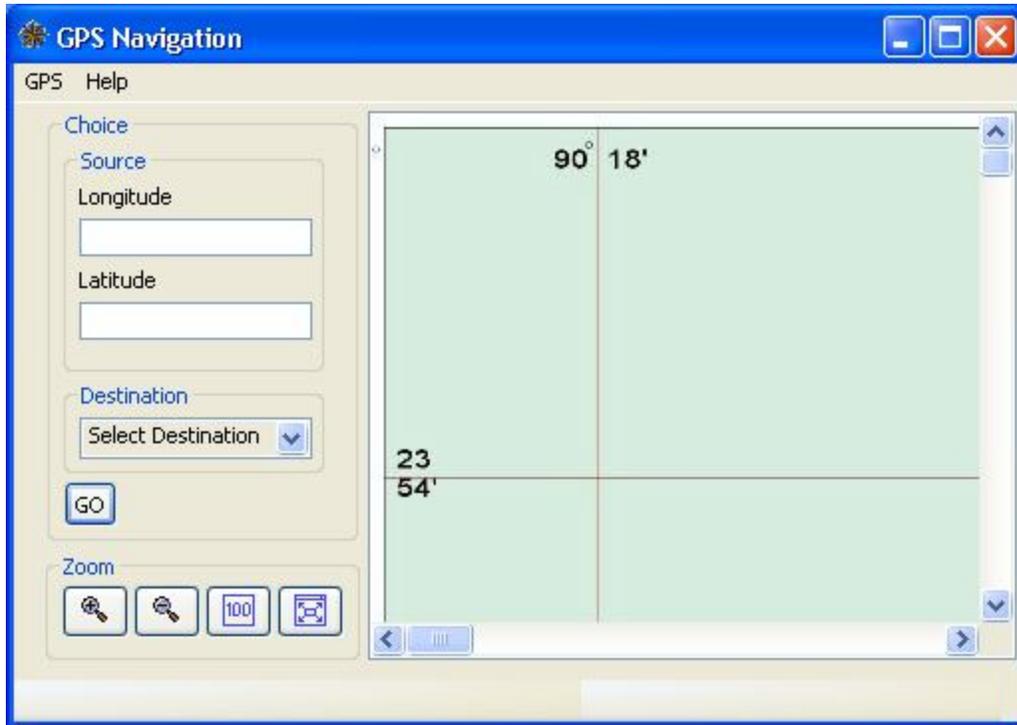

Figure: GUI

4.3.1 FEATURES

This two text box is used to take input the longitude and latitude of users current position.

Figure: Users Current Position

The combo box contains all the nodes. So user can select any one as his/her destination. The button below is used to draw everything. If source longitude and latitude field is blank then nothing is drawn. If destination field is blank then only source is drawn. Otherwise the shortest path is drawn on the map. The source is drawn is such a way that it always remains in the middle.

Inside Zoom group there are four buttons. First button enlarges the map. Second one reduces the map. Third one makes the map to its original size. And the fourth one makes the map fit to the canvas.

Resolution of the map is 2459x4000. When mouse is moved on the map corresponding pixel value is shown with the mouse pointer.

At the same time the coordinate of that particular point is shown below the map.

To calculate pixel value from coordinate the formula below is used:

Two points is taken whose coordinate value and pixel value is both known. These two points are

GIS coordinate	Pixel
90.3233805556/23.8988361111	113,182
90.4180222222/23.6700194444	1918, 3928

GIS difference along x-axis =0.0946416666, PIXEL difference along x-axis = 1805

GIS difference along y-axis =-0.2288166667, PIXEL difference along y-axis = 3746

Formula to get pixel from co-ordinate

==================================

PIXELx = (GISx-90.3233805556)*1805/0.0946416666+113;

PIXELy = - (GISy-23.8988361111)*3746/0.2288166667+182;

Where

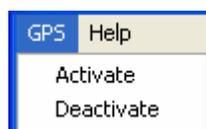GISx = GIS coordinate along x-axis

GISy = GIS coordinate along y-axis

PIXELx = Pixel along x-axis

PIXELy = Pixel along y-axis

At the menu GPS has two radio buttons. One is Activate and the other one is Deactivate. When Activate is marked then the simulation starts and when Deactivate is marked then it stops.

CHAPTER V: IMPLEMENTATION

The system is built as windows application. To implement it programming language Java has been used. The compiler that is used to build the system is Eclipse SDK version: 3.2. A special library which is called SWT (Standard Widget Toolkit) of Eclipse is used to design GUI. The purpose is to make the application faster. SWT uses visual objects of which platform it is working on. That is why it is faster than swing or awt.

CHAPTER VI: TESTING

When the implementation phase is done testing is the last phase of constructing the system. Here several tests are done step by step and those are discussed in following subsections.

6.1 SYSTEM ENGINE TESTING

Firstly Dijkstra algorithm is needed to b tested. The algorithm takes a node as source and another node as destination. And return the shortest path between them. But in our system user might not always been on any node. He/She may be in the middle of the road. The algorithm first tests in which road the user is on and run the algorithm for both of the nodes of that road. Then it get two paths, take the better one and shows it.

Test Case:

User Input:

Enter the Longitude: `90.3835212711`

Enter the Latitude: `23.7617049267`

Enter Destination (Node Number): 110

Output:

The road is:

`9      11`

The shortest path is:

`9---->1---->2---->3---->4---->6---->64---->65---->7---->67---->68-->69`

`---->110`

6.2 GUI TESTING

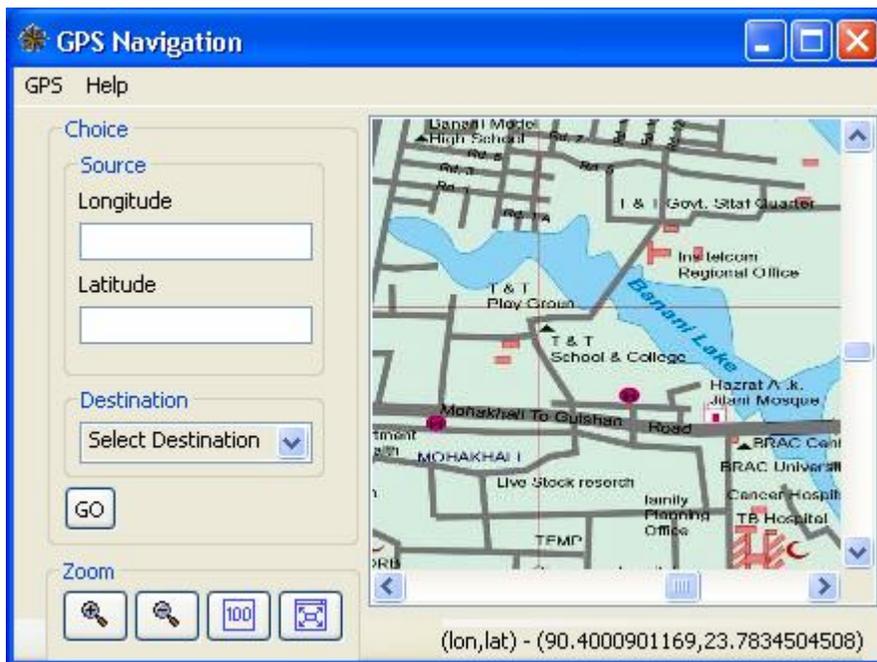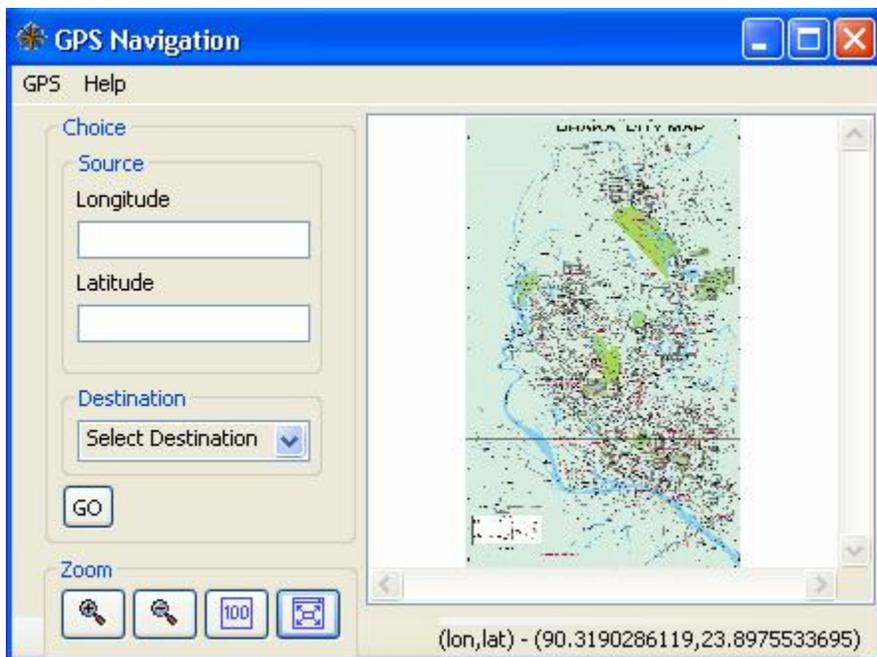GUI testing is very important because all the results needed to be shown graphically.

Here are some tests.

Zoom In Test:

Zoom out test:

Show Full Size Image Test:

Image fit to Screen Test:

6.3 TOTAL SYSTEM TESTING

When the GUI is integrated with Dijkstra algorithm total system requires a test. A simple

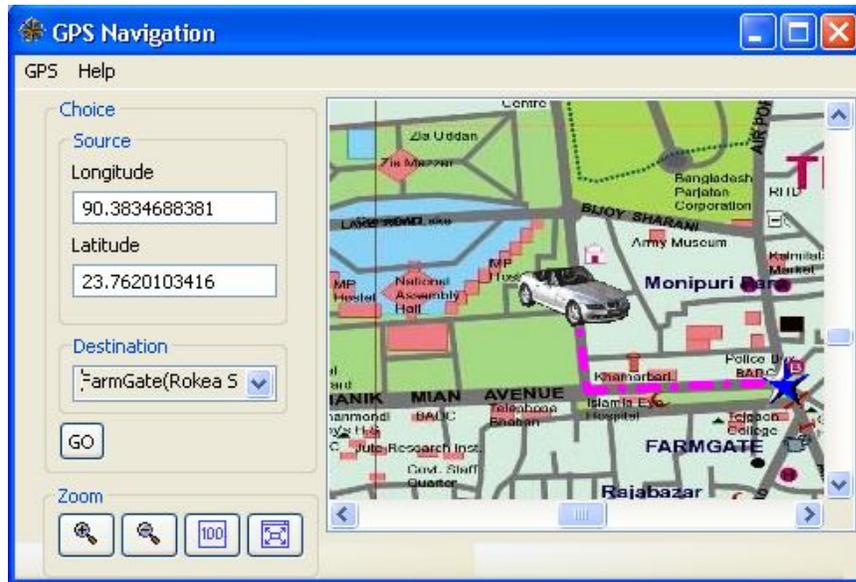test shows the system can depict correct scenario (Here shortest path).

Figure: System test

The image of the car represents current position. The star means destination which is

Farmgate (Rokea Shoroni). And the colored line stands for shortest path. So the output

is absolutely perfect as expected.

6.4 SIMULATION TESTING

Simulation is done reading from a text file. The text file consists of some coordinates.

The file contains data in following format:

```
90.3835212711 23.7617049267
90.3834688381 23.7620103416
90.3834688381 23.7624990048
90.3833115389 23.7637817864
```

One row represents one point. The first double number means longitude and second

double number means latitude of the point. There are many more points like this. The

points are taken from a path. The path is from Farmgate to Mohakhali. The points are given as system input as they are set to two textboxes in sources group. Then go button invoked. In this way simulation is done.

CHAPTER VII: LIMITATION OF THE SYSTEM

Although the system can navigate users but it has some limitations. They are described below.

7.1 DATA COLLECTION

All data needed for this project was collected from Google earth. Internet data transfer rate in our country is very slow. So collecting data from Google earth consumes a lot of time. As there were no alternatives but Google earth so it reduced data collecting speed. Moreover there are thousands of roads and nodes. Collecting data about all of it was not possible in a limited period of time. That is why data collection is incomplete and small area has been covered for data collection.

7.2 SIMULATION

Not only GPS device is unavailable in our country but also no company provides the service for it. Therefore live simulation is not possible. Instead simulation is done by taking data from a text file.

7.3 NOTIFICATION

Notification about taking a turn, toll payment is not available. Road quality, drivers safety, POI (Point of Interest) alert, voice notification are not done due to lack of data. Due to lack of GPS data speeding alert is not possible to be shown.

7.4 ADDING MAPS

The system does not support adding maps. It also can not add new node or roads.

7.5 SEARCHING CATEGORY

The system can only search shortest path by distance. It can not search by traffic condition, road quality, POI or less number of red lights. It does not show the alternate path to go. It means it does show the 2nd best path to the user.

CHAPTER VIII: FUTURE IMPLEMENTATION

Based on the limitation I have some future visions for the system. On future versions of the software it will be more user-friendly. Mainly I will try to overcome the limitations mentioned earlier. At the first stage of future implementation more notification will be added to the system. Then it will be made real time by adding current traffic condition. User interface will be improved like it will show 3-D map. There are some more visions about the system.

8.1 DATA COLLECTION COMPLETION

Future version of the system will be complete in the sense of data collection. Information about all the nodes and roads will be present as well as all the houses of the city. Information about important places like hospital, educational institution, bank, police station, embassy, fire service, bus terminal, airport, railway station, petrol pump, hotel, post office will be there.

8.2 COVER THE WHOLE COUNTRY

Up next the plan is to collect data about the whole country and navigate from anywhere to anywhere in Bangladesh.

8.3 USE REAL GPS CONNECTION

Future versions will use GPS connection to get current position of the traveler.

8.4 APPLICATION FOR PORTABLE DEVICE AND WEB APPLICATION

One of the visions is to convert this application for Connected Limited Device Configuration (CLDC) such as mobile, PDA etc. Nowadays web applications are very

popular. So anyone can get help if there is a web application without installing the software on his computer. That is why converting the windows application to web application is a vision.

8.5 DATABASE DESIGN

After completion of data collection the database will be huge. So database design will be precise. The database will be maintained by SQL server and connection with front end (which may be PC, mobile, PDA) to the database will be done.

CHAPTER IX: CONCLUSION

In our country there is no navigation system. This thesis work entitled as "GIS Based Navigation System of Dhaka City" is not a complete system as it does not have complete database and it covers very small area. But this kind of work has not been done in our country; if is then in a very small scale. The system shows users current position, chosen destination and shortest path from source to destination. It dynamically updates the path with the change of users' position. It notifies users if they are out of road. It also shows the GIS position where the mouse cursor on the map is. It shows the pixel value along with it. The values shown are changed with the change of cursor's positions. Although the system has limitations but it will be overcome in its next implementations. In superior countries this kind of service was started a long time ago. This thesis work explores one thing that if we had not limited resources and facilities we would have done this with them.

REFERENCES

1. Google Earth

2. http://www.google.com/maps

3. http://www.mathforum.com/library/drmath/view/51711.html#search

4. SWT: A Developer's Notebook By Tim Hatton

5. http://jan.ucc.nau.edu/~cvm/latlongdist.html

6. http://sourceforge.net/project/showfiles.php?group_id=64816