MOBILE ROUTING SERVICES FOR SMALL TOWNS USING CLOUDMADE API AND OPENSTREETMAP

Jianghua Zheng [a], Xiaoyu Chen [a], Błazej Ciepłuch [a], Adam C. Winstanley [a] , Peter Mooney [a, b] and Ricky Jacob [a]

[a] Department of Computer Science, National University of Ireland Maynooth, Co. Kildare. Ireland
jianghua.zheng@nuim.ie
[b] Environmental Research Centre, Environmental Protection Agency, Richview, Clonskeagh, Dublin 14. Ireland

Commission VI, WG VI/4

KEY WORDS: routing, navigation, CloudMade API, OpenStreetMap, Location Based Services, OSM

ABSTRACT:

This research presents a practical solution for mobile routing services for small towns using open sources. Free mapping application program interfaces (API) provided by web map services, including routing services, are available to create customised map based web services combining their cartographic base data with the users own data. However, most applications focus on big cities. Location based services in small towns are generally few as many people believe there is a little demand in such areas. However, the demand of LBS applications in some small towns can be as strong as big cities, for example university towns and tourist resorts. Better location based services, especially routing services, can help strangers get familiar with the environment in a short time and lead them to places of interest. However, there are two problems to overcome for such systems. One is cost both in terms of data costs and development time. Open source data and mash-up technology could provide an answer. The other problem is the availability of suitable data of the required accuracy and detail. This is more serious as most free map services, such as Google Maps and Microsoft Bing Maps (Virtual Earth), don't provide sufficient detailed and accurate data for routing services. One feasible and economical way is to create the map ourselves and have it updated by the public. OpenStreetMap (OSM) is a free, open and fast developing map of the world. Detailed data was collected using a GPS logging device and uploaded to OpenStreetMap. The CloudMade API was used to provide multi-mode routing services together with turn-by-turn descriptions for car users, bicycle riders, and pedestrians. This solution is relatively easy and fast to deploy. Maynooth, a small university town in County Kildare Ireland, was used as a test bed. A prototype navigation system was developed for mobile users using the Windows Mobile platform. The system demonstrates that a solution to detailed navigational services for pedestrians, cyclists and drivers can be economical and feasible for small towns.

1. INTRODUCTION

Routing is one of the most important services provided by Location Based Systems (LBS). Mobile routing services encompass way-finding applications for vehicle drivers, pedestrians and cyclists, delivered using mobile terminals. Much research has been carried out on mobile routing algorithms because of the complexity of application environments and variety of user requirements (Huang *et al*., 2007; Huang and Wu, 2008). Like POI (Point of Interests) query services, mobile routing services have become more and more popular in the real world, especially those applications for vehicle drivers, TomTom car navigation systems are typical examples. NAVITIME (Japan) (Arikawa *et al.*, 2007; Zheng *et al*., 2006) and Nokia Maps 2.0 (Dominique, 2008) are two typical examples for pedestrians (Zheng *et al*., 2009). Some free map platforms, such as Google Maps, Yahoo Maps and Microsoft Bing Maps, include direction modules which provide turn-by-turn routing services. Google also plans to allow Android 2.0 phones to give users real-time turn-by-turn walking directions (Adhikari, 2009). However, most current mobile routing services are provided mainly with detailed content only in relatively big cities or for major streets in rural areas. Location based services for small towns are generally ignored as many people and most companies believe there is little demand in such areas and there is no need to invest in detailed data collection for such places.

However, the demand of LBS applications in some small towns may be as strong as that in big cities. University towns and some tourist resorts are typical examples of small towns where there are many visitors potentially requiring access to LBS. They do not have much time to get familiar with the places. Better local location-based services, especially routing services, can help strangers get familiar with a strange environment in a short time. The aim of this work is to provide an effective, efficient and low cost solution to providing LBS, especially routing services, for small towns and tourist resorts. The research question contains three constraints: effectiveness (containing useful data and services); lowest cost; and efficiency (fast system development and easy maintenance). A solution using CloudMade API and OpenStreetMap (OSM) is described. The work takes Maynooth, the only University Town in Ireland, as an example and we put forward a mobile routing services prototype to demonstrate the solution.

The rest of the paper is organized in four sections. Section 2 discusses why OpenStreetMap is used as a data source. The CloudMade API is described in the following section. In section 4, we provide a detailed discussion of the development and implementation of the prototype for the LBS for small towns. Finally, the paper closes with a discussion of conclusions from the work and puts forwards some suggestions for future work. The work is part of the eCampus project, which is constructing a major testbed for StratAG, the Strategic Research Cluster in Advanced Geotechnologies (www.stratag.ie) centred at

National University of Ireland Maynooth. It focuses on constructing a campus information system including diverse location-based services.

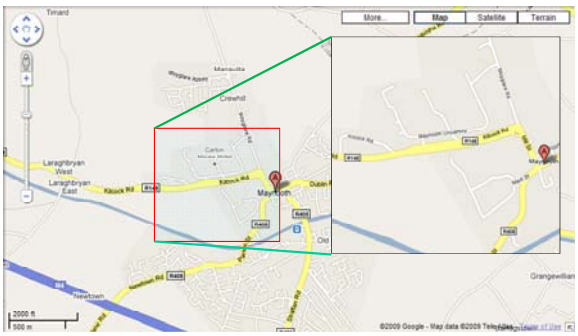
2. OPENSTREETMAP

2.1 Experimental Area and Data Collection Methods

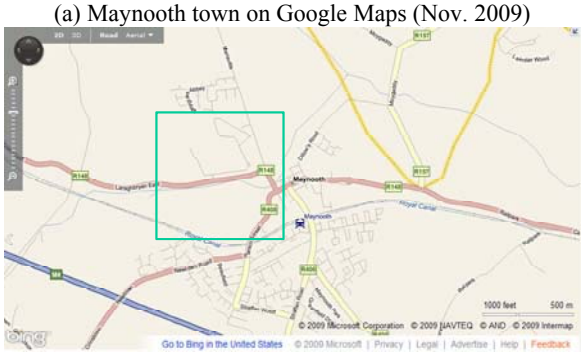
Maynooth is a university town located in north County Kildare, Ireland. It is about 25km west of Dublin city centre. Accurate and sufficient data of the area is the basis for local mobile routing services together with POI queries. There are four major ways to obtain local spatial data and associated attribute data:

- From current data available for LBS
- From professional survey companies or agencies
- From free map providers
- Collect the data by yourself or by crowd sourcing

In most cases, the former two are not suitable, especially the second one which might cost a lot of money or have low cost performance. The third one is the most convenient. However, if we check Maynooth town, like most other small towns and tourist resorts, there is poor representation on the most popular commercial free map platforms, such as Google Maps, Yahoo Maps and Microsoft Bing Maps. Navteq and TeleAtlas, the two biggest map data companies in the world, are the map providers of those platforms and their data just focuses on the requirements of vehicle navigation. Figure 1(a) shows the map of Maynooth on Google Maps and Figure 1(b) the content on Microsoft Bing Maps.



(a) Maynooth town on Google Maps (Nov. 2009)



(b) Maynooth town in Microsoft Bing Maps (Nov. 2009)

Figure 1 Maynooth town on web map services

The rectangles in Figure 1 show the approximate area of National University of Ireland Maynooth. Only the main streets and a few POIs of the town can be obtained from Google Maps and Microsoft Bing Maps. This poor spatial data coverage is not sufficient for establishing effective LBS routing services for this area. As purchasing commercial data is expensive, we have to collect the data ourselves. OpenStreetMap provides an outstanding example of a spatial data source for this area to which we can contribute. The next section describes the OpenStreetMap project.

2.2 OpenStreetMap

OpenStreetMap (OSM) is a free map of the entire world. It allows you to view, edit and use geographical data in a collaborative way from and for anywhere on Earth. It uses a crowd sourcing model to provide user-generated street maps. There have been various geo-wiki applications that utilise user-generated content for maps (Jacob *et al*., 2009). However, OSM is probably the most extensive and effective project currently under development (Haklay and Weber, 2008). OSM development is geographically unbalanced. In general, it is more complete in Europe. Unlike other web map services, OSM provides a set of tools to create a free editable map of the world. The maps are created using data from portable GPS devices, aerial photography, other free sources or simply from local knowledge (http://en.wikipedia.org/wiki/OpenStreetMap). It integrates some useful tools for importing, editing, exporting and generating geometry from GPS trails which encourages users to be not only users but also data generators. For this purpose, there are also some offline editing tools, such as JOSM and OSM2Go. All these tools are free and easy to use.

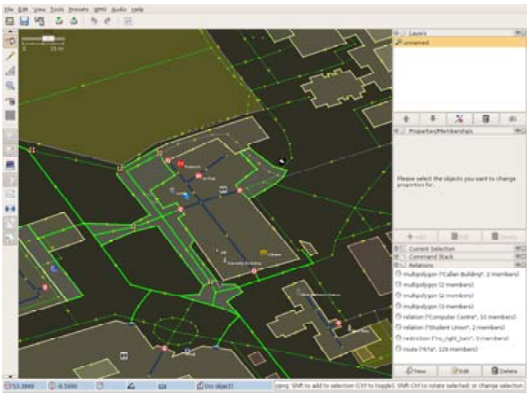


Figure 2 Interface of JOSM editor

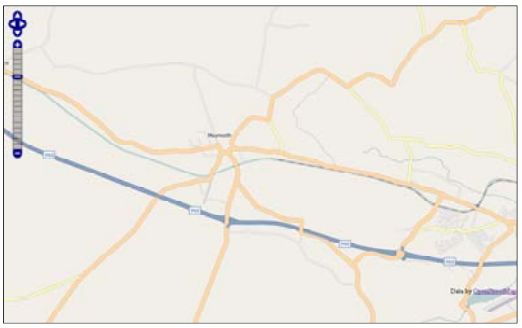
More and more organizations and individuals provide APIs using OSM data, such as for map rendering and routing such as the CloudMade Routing API. The reasons for selecting OpenStreetMap as a platform for data collection and representation can be summarised as:

- ***Totally free***. All data are generated by the public with little usage restrictions and no cost. The OSM tools and APIs are powerful but also are also free.
- ***Multiple outputs***. It is possible to render various popular formats of data of the same area from the OSM dataset giving flexibility of use.
- ***More vivid map data with various attributes***. OSM provide the public a set of powerful tools to render your own style OSM data for your personal map based applications. This attributes to its two major components, Mapnik and Osmarender.
- ***More current data***. OSM data is being updated constantly. OSM also has a mechanism for users to update local data. For example, the OSM data of Ireland is updated weekly.

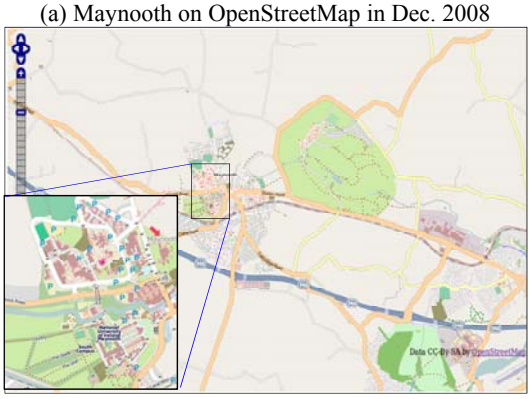
OSM data can be stored and managed easily and efficiently using PostgreSQL/PostGIS, a powerful open-source spatial database management system, making it easy to create new applications.

2.3 Data Creation

The OSM data collection of Maynooth town was done by members of our research group and student and volunteer helpers, mainly using GPS logger devices, such as GlobalSat® DG-100 GPS Data Logger. Much of the roads and paths were collected by bicycle.



(a) Maynooth on OpenStreetMap in Dec. 2008



(b) Maynooth on OpenStreetMap in Oct. 2009

Figure 3. Data Creation of Maynooth town

Figure 3(a) shows there was little information of Maynooth on OSM platform in December 2008. However, now following the upload and editing of the logged data, there is abundant information of Maynooth town, including various POIs, buildings, streets, bicycle lanes, and even data inside offices of some buildings.

3. THE CLOUDMADE ROUTING API

3.1 Developing Modes for Routing

There are two typical software development strategies to construct routing procedures:

- Third party APIs
- Implementing original routing algorithms.

The first is convenient and fast for the developer to deploy applications. However, if there are special requirements, for example if people have preferences such as passing through buildings as much as possible because it is raining heavily, it becomes necessary to build more parameterised routing modules.

3.2 Why use CloudMade Routing API?

Routing algorithms are easy to realize on a well-formed link-node network (Zhan, 1997). However, the network data obtained from OSM is not suitable for direct optimal path computing. This is because the network data is generated from public submissions that may not result in a well prepared link-node network. A road will not usually be captured in segments though it might have several intersections with other roads. Rather than having to manually edit the road network, the PostgreSQL/PostGIS spatial database provides powerful and helpful standard functions to generate intersections and build the link-node network. Third-party routing APIs allow developers the possibility of quick development provided they fulfil the specific routing requirements of the application.

There are several third-party routing APIs, which could be used with OSM data, such as the CloudMade Routing API (http://cloudmade.com/, 2009) and pgRouting. Both of them are open source. The main objective of pgRouting is to provide routing functionality for PostgreSQL/PostGIS (http://pgrouting.postlbs.org/, 2009). It includes several routing algorithms such as traditional Dijkstra, A*, Shooting Star and Travelling Sales Person (TSP). Since OSM data is stored and managed in PostgreSQL/PostGIS, we can add and use pgRouting as standard functions of PostGIS. The CloudMade Routing API is currently the most used with OSM data. It provides car, foot and bicycle modes for users. It also provides turn-by-turn direction descriptions with multi-lingual templates, including Chinese. Another advantage of CloudMade is it generally serves requests more quickly than a local server does. As a whole, using CloudMade Routing API makes routing services easily and economically realized.

4. PROTOTYPE OVERVIEW

We could build the prototype with B/S or C/S architecture. The trend is B/S architecture. However, we have selected C/S architecture in our prototype for fast design and easy testing. The application is designed to run on a smart phone (HTC 3470), running Windows Mobile OS. C# is the development language for the mobile terminal. PHP is used to develop the server-side program.

4.1 System Architecture

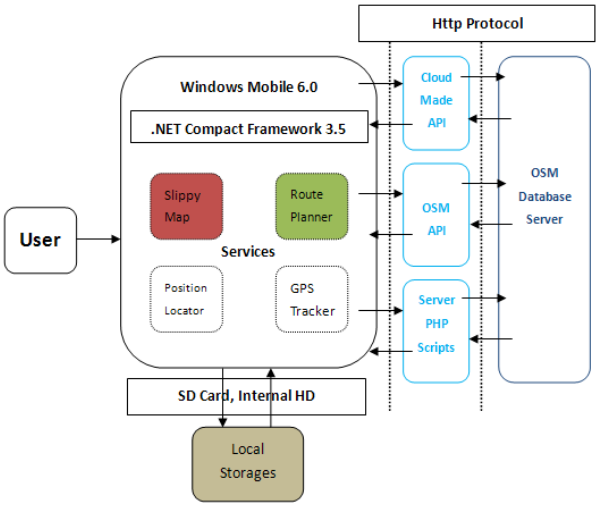


Figure 4 System Architecture of the prototype

Figure 4 shows the architecture of the prototype. The basic spatial data is on an OpenStreetMap database server and it provides map services through the OSM API. The CloudMade API provides powerful routing algorithm for both web and

mobile applications based on OSM data. We provide three travel modes: car, walking and bicycle.

4.2 Slippy Map for Mobile Terminals

Fast and continuous map presentation on the mobile terminal is a basic requirement of this application. Slippy map mechanisms are widely used in map-based applications. For example, Google Maps and Microsoft Bing Maps have their own slippy map system based on commercial map databases (Haklay, 2008). There are also third party web services which provide the slippy map interface for developers, which can be used to render various map resources. Mapnik and OpenLayers are mainly used for such proposes. Unfortunately, how to implement the slippy map on a mobile device is hardly seen for developers to reference. We have implemented our own module for this.

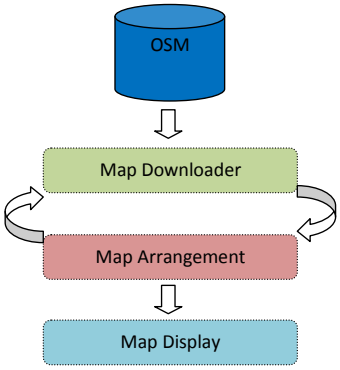


Figure 5. Main software architecture for slippy map

Figure 5 shows the software architecture for slippy map implementation. The core of the map downloader class is OSM API calls wrapped within a C# http request. OSM provides a structured URL to obtain the map tiles. The URL is a combination of zoom level, X and Y coordinates, as in the example in figure 6.

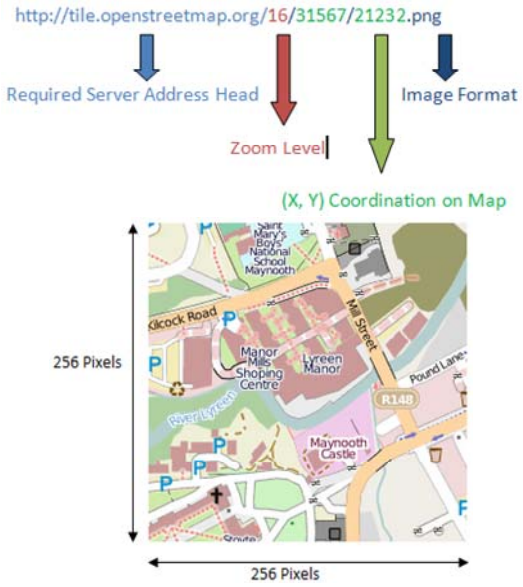


Figure 6. Map as rendered on mobile terminal

Figure 6 displays the map representation on a mobile terminal after the slippy map processing. C# code segment for downloading a OSM map tile is as figure 7.

```
public static void GenerateImages(String[] urls,String[] names)
{
    int i = 0;
    foreach (string uri in urls)
    {
       Image result = null;
       Stream stream=null;
       HttpWebResponse response = null;
       {
       HttpWebRequest request =
(HttpWebRequest)WebRequest.Create(new Uri(uri));
       request.Method = "GET";
       response = (HttpWebResponse)request.GetResponse();
       stream = response.GetResponseStream();
       result = new Bitmap(stream);
       result.Save(Path.Combine(filepath, names[i]),
ImageFormat.Png);
       stream.Dispose();
       result.Dispose();
       response.Close();
       }
       i++;
     }
 }
```

Figure 7. C# code segment for downloading OSM map tile

Map re-arrangement is triggered when certain events are detected, for instance a map container going out of valid region, or relocation of the map centre. Four map tiles are used to fully cover the screen of the device. If an empty space occurs during map operation, the tile positions will be rearranged accordingly. Figure 8 shows examples of scrolling resulting in the retrieval and addition of new map tiles.

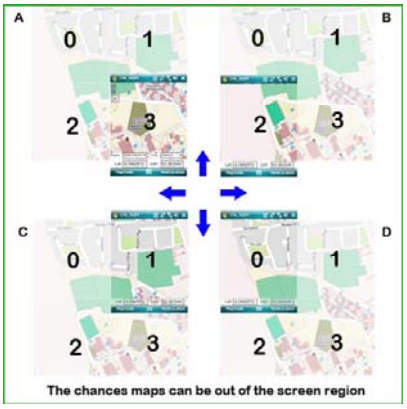


Figure 8. Four situations trigger map arrangement event

Map Tile 0 X-1, Y-1 .../31565/21230.png	Map Tile 1 X, Y-1 .../31566/21230.png
Map Tile 2 X-1, Y .../31565/21231.png	Map Tile 3 X, Y .../31566/21231.png

Figure 9. Positional relationship of tiles in OSM

Figure 9 illustrates the determination of which tiles to download during scrolling. Map tile 3 is assigned to be the reference tile on right-lower corner of the map container and from this the URL references of surrounding tiles can be calculated. OSM provides a formula to derivate X-Y coordinates of maps from latitude and longitude values. The formula can also work in reverse.

OSM Formula to derivate X Y Coordination

```
n = 2 ^ zoom
xtile = ((lon_deg + 180) / 360) * n
ytile = (1 - (log(tan(lat_rad) + sec(lat_rad)) / π)) / 2 * n
```

C# Implementation of OSM Tile Name Derivation Formula

```
public PointF WorldToTilePos(double lon, double lat, int zoom)
{
    PointF p = new Point();
    p.X = (float)((lon + 180.0) / 360.0 * (1 << zoom));
    p.Y = (float)((1.0 - Math.Log(Math.Tan(lat * Math.PI / 180.0) +
                1.0 / Math.Cos(lat * Math.PI / 180.0)) / Math.PI) /
2.0 * (1 << zoom))
    return p;
}
public PointF TileToWorldPos(double tile_x, double tile_y, int zoom)
{
    PointF p = new Point();
    double n = Math.PI - ((2.0 * Math.PI * tile_y) / Math.Pow(2.0,
zoom));
    p.X = (float)((tile_x / Math.Pow(2.0, zoom) * 360.0) - 180.0);
p.Y = (float)(180.0 / Math.PI * Math.Atan(0.5 * (Math.Exp(n) -
      Math.Exp(-n))));
    return p;
}
```

Figure 10. OSM derivation formula

The two C# functions in figure 10 can be used to construct the URL for a map tile. For example, test data (latitude= 53.381646, longitude= -6.582667, zoom= 16) here is chosen to input into C# function WorldToTilePos(). Point (31569, 21232) is returned after execution. From this the URL for this map tile is http://tile.openstreetmap.org/16/31569/21232.png . If this tile is the reference tile in slippy map system, URLs for rest of tiles are as in Figure 11.

Map Tile 0 http://tile.openstreetmap.org/16/31568/21231.png	Map Tile 1 http://tile.openstreetmap.org/16/31569/21231.png
Map Tile 2 http://tile.openstreetmap.org/16/31568/21232.png	Map Tile 3 http://tile.openstreetmap.org/16/31569/21232.png

Figure 11. URLs of four tiles in Slippy Map System

Displaying the map is the final and simplest of the three components of slippy map. The image in the control area is changed according to the required arrangement of the map tiles. After the required tiles are downloaded, an image merging function is used to join them together in memory in order to fit into map container and then render the data to the screen. The advantage of doing this is to create a clean and easy drawing environment for other map functions, like route planning and image icon display.

4.3 Routing using CloudMade API

The CloudMade Routing uses an http protocol. The structure of the URL is similar to the OSM tile querying URL, but instead of returning a PNG image, a GPX or JS format file is returned.

```
http://routes.cloudmade.com/YOUR-API-KEY-GOES-HERE/api/0.3/start_point,[transit_point1,...,transit_pointN],end_point/route_type[/route_type_modifier].output_format [?lang=(en|de)][&units=(km|miles)]
```

Figure 12. CloudMade Routing planning URL structure

Figure 12 shows the CloudMade route planning URL structure. The general parameters in the structure are “start_point”, “end_point”, “route_type” and “output_format”. Other parameters are optional.

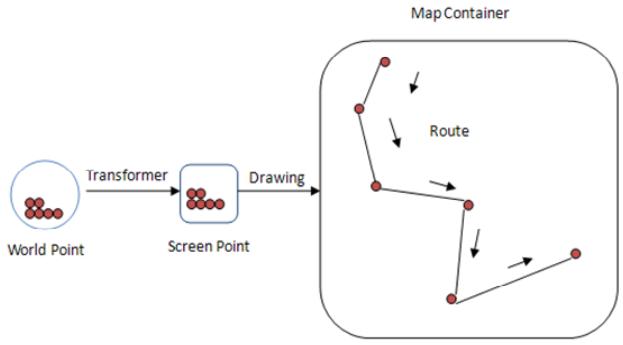
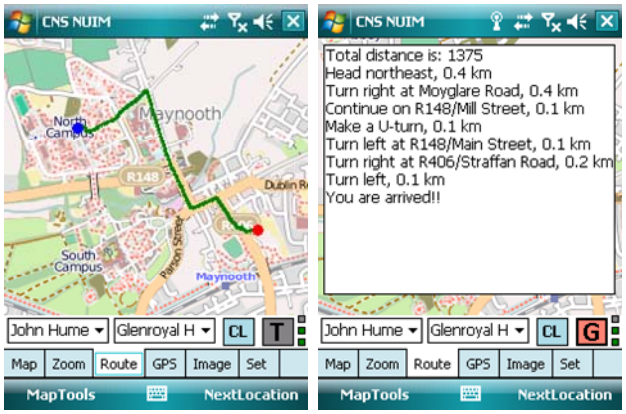


Figure 13. Drawing route on map of mobile phone

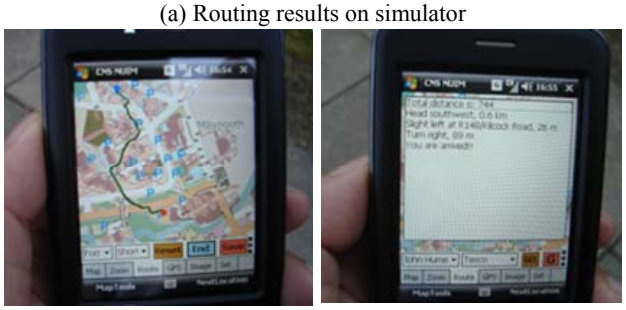
Figure 13 displays the main steps to draw a route on the map displayed on the mobile terminal.

4.4 User interfaces of the prototype

Figure 14 shows examples of the map and textual routing interfaces on both the API simulator and as actually rendered on the mobile terminal.



(a) Routing results on simulator



(b) Routing on HTC 3470 model

Figure 14. Routing interfaces

5. CONCLUSION AND FUTURE WORK

The demand of LBS applications in some small towns might be as strong as big cities; university towns and some tourist resorts as examples of such small towns where there are many tourists and people potentially requiring access to LBS. This paper describes an effective, low cost and efficient solution of mobile routing services for such small towns. The core parts of the solution are a spatial database platform (OpenStreetMap and PostgreSQL/PostGIS) and a routing module (CloudMade Routing API). These are all open source products. The work also discussed a successful method to display slippy maps on mobile terminals. By demonstrating a prototype of a mobile

routing service for Maynooth town, we have showed how this could benefit more small towns.

The interfaces and response speed should be optimized in the future. During the routing process, landmarks are helpful for pedestrian users to make certain that he is walking in the right direction (Millonig and Schechtner, 2007; Hile *et al*., 2009). We plan to carry out some landmark based applications, such as using geotagged photography, as the extension of the solution described in this paper.

ACKNOWLEDGEMENTS

Research presented in this paper was funded by a Strategic Research Cluster grant (07/SRC/I1168) by Science Foundation Ireland under the National Development Plan. The authors gratefully acknowledge this support.

REFERENCES

Adhikari R., 2009. Android 2.0 Phones Get New Google Nav App, http://www.technewsworld.com/story/Android-20-Phones-Get-New-Google-Nav-App-68496.html (accessed 30th Oct. 2009)

Arikawa M., *et al*., 2007. NAVITIME: Supporting Pedestrian Navigation in the Real World. *Pervasive Computing*, 6(3), pp. 21-29

Bonte D., 2008. The Mobile World Congress 2008: Pedestrian Navigation at Last. 11 Feb. 2008, Barcelona, Spain. http://www.abiresearch.com/Blog/Telematics_Blog/474 (accessed 28 Oct. 2009)

Jacob R., 2009. Campus Guidance System for International Conferences Based on OpenStreetMap. In: *LNCS: Web and Wireless Geographical Information Systems*, Springer, Berlin / Heidelberg, German, Volume 5886/2009, pp. 187-198

Haklay M. and Weber P., 2008. OpenStreetMap: User-generated street maps. *Pervasive Computing*, 7(4), pp.12-18

Hile, H., *et al*., 2009. Landmark-Based Pedestrian Navigation with Enhanced Spatial Reasoning. In: *LNSC: Pervasive Computing*, Springer, Berlin/Heidelberg, Volume 5538/2009, pp. 59-76

Huang, B., Wu, Q., and Zhan, F. B., 2007. A shortest path algorithm with novel heuristics for dynamic transportation networks. *International Journal of Geographic Information Science*, 21(6), pp. 625-644.

Huang, B. and Wu, Q., 2008. Dynamic Accessibility Analysis for Location Based Service Using an Increment Parallel Algorithm. *Environment and Planning B*, 35(5), pp. 831–846.

Millonig A. and Schechtner K., 2007. Developing Landmark-Based Pedestrian-Navigation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 8(1), pp. 43-49

Zhan F B., 1997. Three Fastest Shortest Path Algorithms on Real Road Networks. *Journal of Geographic Information and Decision Analysis*, 1 (1), pp. 69−82

Zheng J.H., *et al*., 2006. Study on Data Organization of Personal Navigation Services, *Computer Engineering*, 32(24), pp. 41-47

Zheng J.H., *et al*., 2009. Spatial characteristics of walking areas for pedestrian navigation. In: *Proceedings of the 2009 Third International Conference on Multimedia and Ubiquitous Engineering*, IEEE, Piscataway, NJ, USA, pp.452-458