Appl Geomat (2010) 2:123–135
DOI 10.1007/s12518-010-0028-7

ORIGINAL PAPER

Development of track log and point of interest management system using Free and Open Source Software

Daisuke Yoshida · Xianfeng Song · Venkatesh Raghavan

Received: 15 March 2010 / Accepted: 30 June 2010 / Published online: 23 July 2010

Abstract Recent advanced performance of low-cost Global Positioning System (GPS) and GPS-enabled cell phones has contributed a great deal to the development of location-aware services and systems. High-speed broadband technology has promoted collaborative projects such as OpenStreetMap or other User Generated Contents services. In this research, a Web-based prototype system for GPS track log and point of interest (POI) management was developed to archive a collaborative framework in field surveys. The main function of the system can be separated into three parts: data collection, data management, and data quality enhancement. The system supports real-time data collection for the future ubiquitous environment and also can monitor real-time GPS positions. This research shows functionalities that can minimize GPS errors using Dilution of Precision filtering and data quality enhancing techniques using the Douglas-Peucker algorithm and PgRouting. The research introduces a system that provides an interoperable framework in which to work with other geospatial services through open geospatial standards.

D. Yoshida (✉)
Faculty of Liberal Arts, Tezukayama Gakuin University,
2-1823, Imakuma,
Osakasayama City, Osaka 5898585, Japan
e-mail: yoshida@la.tezuka-gu.ac.jp

X. Song
Graduate University of Chinese Academy of Sciences,
19A, Yuquan Road,
Beijing 100049, China

V. Raghavan
Graduate School for Creative Cities, Osaka City University,
3-3-138 Sugimoto Sumiyoshi-ku,
Osaka 5588585, Japan

Keywords Point of interest (POI) · Location-based service (LBS) · Global Positioning System (GPS) · Quality filtering · Simplification

Introduction

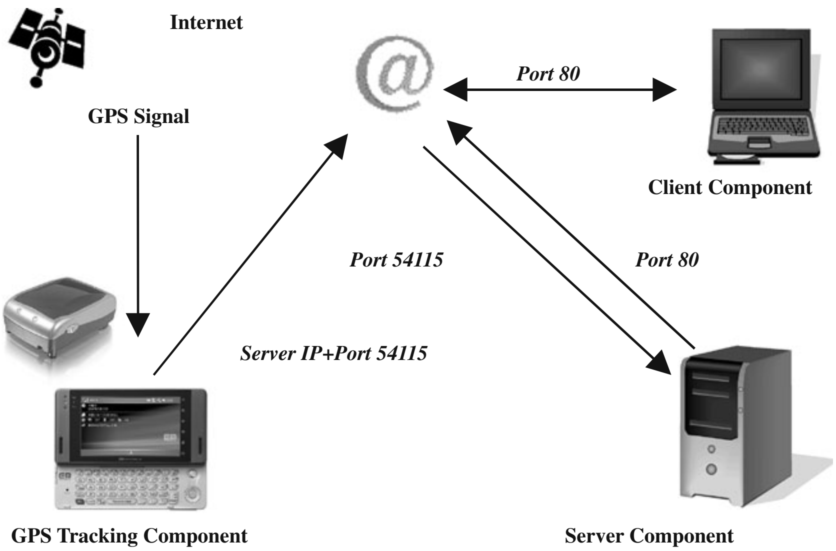
One of the recent developments in geospatial technology is the ubiquity of location information. The emergence of Global Positioning System (GPS)-enabled cell phones and mobile devices has facilitated mass marketing of location-based services (LBS; e.g., navigation services, geo-tagged contents, and tracking of assets and personnel). Integration of GPS with Geographic Information Systems (GIS) is also effective for fleet management, logistics, emergency medical services, rescue, and relief work (Derekenaris et al. 2001; Panahi et al. 2008). There are many GPS tracking devices and systems available on the market. However, accessing LBS requires the use of customized devices or service-supported cell phones with their own closed proprietary protocols or closed source code. The devices cannot be modified and can be only used for their particular purposes. Therefore, it is difficult to develop or tailor LBS that could meet a wide variety of application scenarios and users' needs. There is a need to implement a framework, wherein LBS and other geospatial services can interoperate more easily and effectively.

FOSS4G for LBS

A number of "open" initiatives have been developed to support LBS. An open source approach to implement Web-based GPS tracking management system, OpenGTS,[1]

[1] http://www.opengts.org

Fig. 1 System overview



provides Web-based authentication, mapping, report services, and other useful functions. The "Open Device Monitoring and Tracking Protocol," otherwise known as OpenDMTP,[2] is a protocol and framework that allows bidirectional data communications between servers and devices such as GPS, as well as temperature and other data collected in remote-monitoring devices. In order to have GPS data online, Gpsd[3] works as a service daemon that monitors GPS devices and provides the data to other GPS applications through the Internet. GPSBabel[4] is one of the most popular Free and Open Source Software (FOSS) packages for GPS data conversion and transfer. Many other FOSS tools are available through the Internet which can be applied to LBS. These tools basically comply with open geospatial standards and have the potential to overcome the proprietary limitations in LBS and provide an interoperable framework to work with other geospatial services.

Prototype system for track log management

High-accuracy GPS data is an essential concern for LBS. The performance of low-cost GPS receivers has been greatly improved due to better GPS hardware and enhanced receivers' functionality. However, the GPS signal is degraded in "urban canyons," under areas of forest canopy, and also due to other interference factors. There are several desktop tools to minimize GPS errors (e.g., GPSBabel, OpenJUMP with RoadMatcher Application,[5] and GRASS GIS[6]). GPSBabel has filtering functions to remove unreliable points with Dilution of Precision (DOP), the minimum required number of satellites and GPS fix status. However, these desktop tools need to be installed in PCs, and the operations are cumbersome for users. It is necessary to have user-friendly interfaces and also interoperable functions in order to provide collaborative frameworks for multiple users. The prototype system cannot only manage GPS track logs such as OpenGTS, but should also minimize GPS errors through Web-based interfaces.

One of the important geospatial services is point of interest (POI) management. POIs provide vital information as contextual data to enable spatial searching since street maps alone are not enough to provide LBS. Recently, many approaches to manage POIs have been implemented. Nguyen et al. (2006) developed a prototype system to manage historical POIs in Hanoi, Vietnam, with a spatially enabled Contents Management System (CMS). In this research, the prototype system also provides a POI management function that can store POIs in a Data Base Management System (DBMS) with multimedia data and can synchronize POIs with GPS track points automatically.

Yoshida and Raghavan (2008) developed a prototype system for managing GPS track logs. The research was focused on real-time tracking and showed an interactive interface for displaying track logs on a Web browser. The article introduced a function for enhancing GPS data quality using a map-matching method. The approach was effective in simple street areas, but in intricate areas such as downtown in urban cities, it was not effective.

In this research, the functionality of the data enhancement was improved by implementing a data simplification algorithm and introducing a shortest-path algorithm using PgRouting.[7] This article also focused on implementing the

[2] http://www.opendmtp.org

[3] http://gpsd.berlios.de

[4] http://www.gpsbabel.org

[5] http://www.jump-project.org/project.php?PID=RM&SID=DOWN

[6] http://grass.itc.it

[7] http://pgrouting.postlbs.org

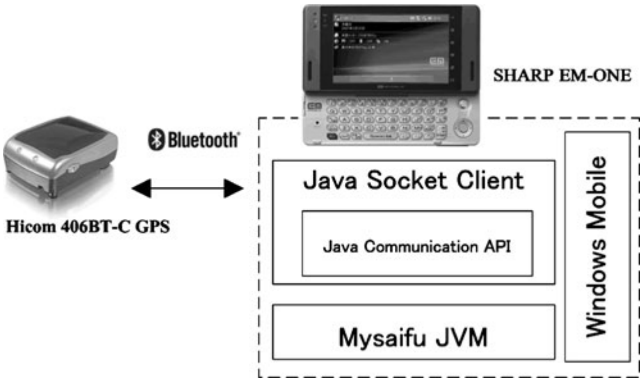


Fig. 2 System architecture of GPS tracking component

POI management functionality to provide a collaborative framework for field survey.

Architecture and feature of the prototype system

The prototype system supports real-time GPS tracking and track log management. The workflow for the system can be demarcated into three components (Fig. 1). The first being real-time and non-real-time GPS tracking components. Next is the server component that provides data processing, archiving, and geospatial services. The final component is the client component for viewing GPS locations and track logs. (Each of these components is described in following sections.)

GPS tracking component

With the widening of Internet coverage in public spaces, high-speed wireless access has become increasingly popular for the general public, especially in Japan. One service example is Worldwide Interoperability for Microwave Access (WiMAX) service started in Japan in 2009 and covering a wide area. The mobile service allows users to access fast Internet connectivity with a maximum speed of up to 40 Mbps. Also coming around 2010 is the next generation of Personal Handy-phone System (PHS) and CDMA services, known as Long Term Evolution (LTE) or Super 3G. Due to this growth, tracking services are also developing, for example, navigation services using cell phones.

In this research, a Personal Digital Assistant (PDA), Windows Mobile 5 in SHARP EM-ONE with four low-cost Bluetooth GPS receivers or loggers (e.g., Hicom 406BT-C, Wintec G-Rays2, i-Blue GPS loggers, and GlobalSat TR-102 Personal Tracker) were used for the experiments in Japan and India. Japanese cell phones use the Code Division Multiple Access (CDMA) system (W-CDMA and CDMA2000); however, most countries use Global System for Mobile communications (GSM) and General Packet Radio Service (GPRS) for data communications, with the maximum speed being 115 kbps.

The experiment in Japan was carried out using the High Speed Downlink Packet Access (HSDPA) connection of W-CDMA. HSDPA provides wireless connection at the maximum downloading speed of 21 Mbps, while the maximum uploading speed is up to 5.8 Mbps in Japan.

GPS data retrieval and socket program for real-time tracking

Figure 2 shows the system architecture of the GPS tracking component. A Java program was developed in order to retrieve GPS data in the National Marine Electronics Association (NMEA) format. The format consists of

Fig. 3 System architecture of server component

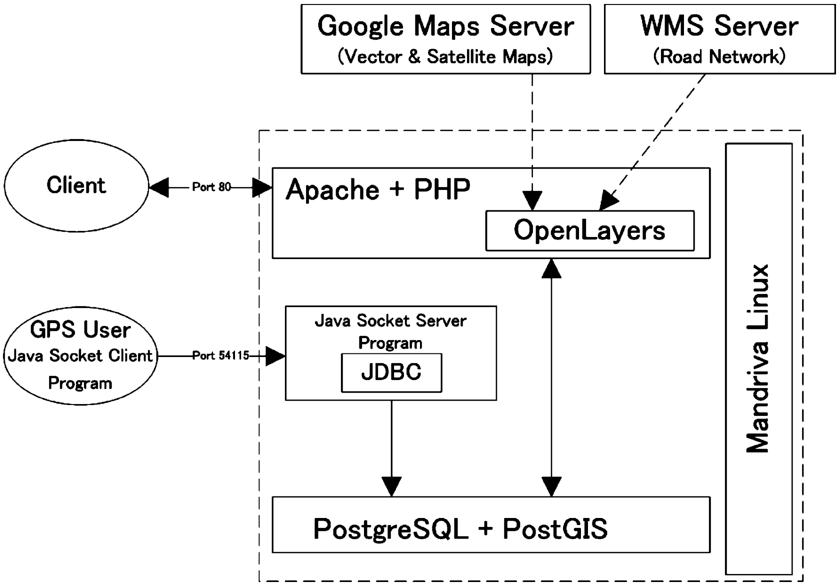


Table 1 Explanation of archived GPS data

Name	Description	NMEA
Date	Date in coordinated universal time	RMC
Time	Time in coordinated universal time	RMC
Direction	Direction of travel	RMC
Speed	Speed in km/h	RMC
DD Longitude	Longitude in decimal degree	GGA
DD Latitude	Latitude in decimal degree	GGA
Altitude	Estimated altitude above the sea	GGA
Number of Satellite	Number of active satellite	GGA
PDOP	Positional dilution of precision	GSA
VDOP	Vertical dilution of precision	GSA
HDOP	Horizontal dilution of precision	GSA
Checksum	Checksum of each NMEA sentence	GSA
GPS Mode	Positioning status: 2D, 3D or no fix	GSA

sentences such as Recommended Minimum Specific GPS Data (RMC), Global Positioning System Fix Data (GGA), and GPS DOP and Active Satellites (GSA) from GPS receivers. The Java Communication API[8] was implemented in the program to communicate with GPS receivers. In addition, Mysaifu JVM[9] was used for executing the Java program under Windows Mobile, since Windows Mobile does not support Java Virtual Machine (JVM). To send GPS data to the server, a socket (or network) client program was implemented in the Java program. The program retrieves NMEA data and sends it to the server regularly every 5 or 10 s, depending on users' needs. The program runs under any OS and devices that support the JVM (e.g., Linux, Apple Macintosh and also embedded devices).

Client component

The server component enables searches within track logs and tracking services to the client components through Hyper Text Transfer Protocol (HTTP). As such Web browsers, like Firefox, Opera, and Google Chrome are examples of the client components that can interact with the server. Digital globe viewers, such as Google Earth, can also be the client component for real-time tracking. Desktop GIS or other GIS applications can be used for displaying search results, which are exported from the prototype system. Detailed information will be explained in "POI management functionality."

Server component

The server component was implemented on a Linux platform using Apache, PHP, Java, PostgreSQL, and PostGIS[10] in order to store GPS data and display the real-time position of GPS targets on the Web map (Fig. 3). OpenLayers[11] was adopted for the Web mapping viewer application as the client interface.

GPS data parsing and archiving functionality for real-time tracking

The Java program opens a server socket to receive GPS data that comes from the GPS tracking component. NMEA data from the tracking component is parsed separately and archived in PostgreSQL. Java Data Base Connectivity (JDBC) was used for an interface to PostgreSQL. Table 1 shows parsed information, the description, and the NMEA data sources.

The information is retrieved from NMEA sentences. Not all of the information is used for the prototype system such as the information of speed, direction, Vertical Dilution of Precision (VDOP), and Positional Dilution of Precision (PDOP), which are archived in the database for possible future use. For example, speed and direction information can be utilized for developing navigation applications. The system archives three types of location information: data of longitude and latitude in decimal degree, data of longitude and latitude in Spherical Mercator, and map-matched longitude and latitude in Spherical Mercator. The longitude and latitude data in Spherical Mercator are converted by the Java program. Google Maps, Yahoo Maps, and Microsoft Virtual Earth providers use Spherical Mercator as the projection that treats the Earth as a sphere and is measured in meters. Spherical Mercator is widely adopted in the open source Web mapping community. The International Association of Oil & Gas Producers (OGP)[12] maintains the European Petroleum

[8] http://java.sun.com/products/javacomm/index.jsp
[9] http://www2s.biglobe.ne.jp/~dat/java/project/jvm/index_en.html
[10] http://postgis.refractions.net
[11] http://openlayers.org/
[12] http://www.ogp.org.uk

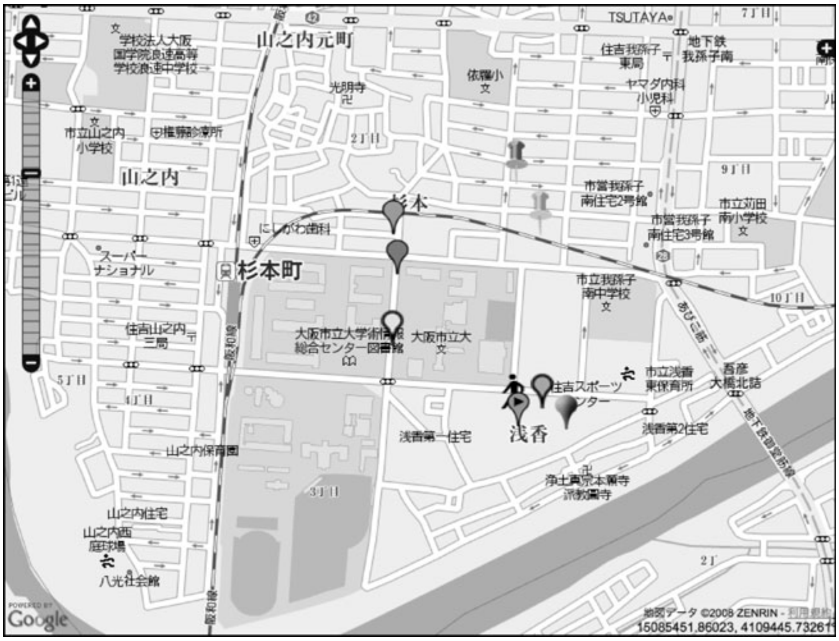


Fig. 4 Real-time tracking in the OpenLayers interface

Survey Group (EPSG) code.[13] This research adopted EPSG:900913, which indicates Spherical Mercator projection and it was defined by Google. Therefore, an open source cartographic projections library, Proj.4,[14] was implemented in the Java program in order to convert the projection to match the projection used by Google Maps and other Web map services. The following sentence is a parametric definition for EPSG:900913 in Proj.4.

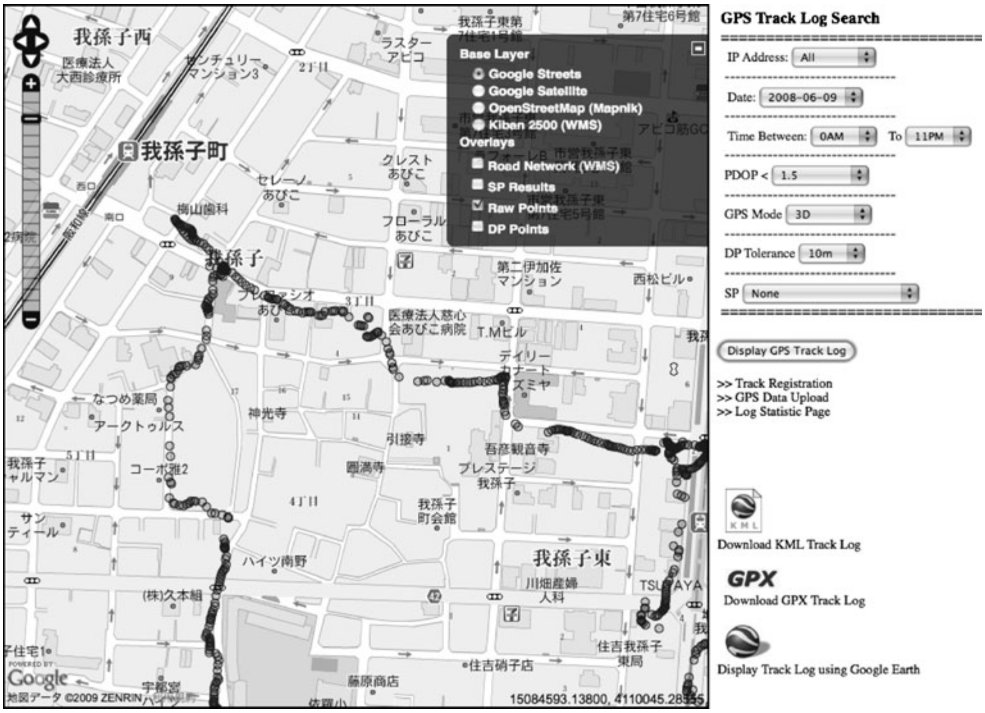
$$< 900913 > +proj = merc + a = 6378137 + b = 6378137 + lat\_ts = 0.0 + lon\_0 = 0.0 + x\_0 = 0.0 + y\_0 = 0 + k \\ = 1.0 + units = m + nadgrids = @null + wktext + no\_defs <>$$

Latitude and longitude in decimal degrees were used for displaying the result in Google Earth. In addition to the above projection processes, the program converts the parsed information to GeoRSS (Geographical Rich Site Summary)[15] and KML for displaying real-time locations in OpenLayers and Google Earth. GeoRSS provides geographically encoded objects for a RSS feed. GeoRSS allows encoding location information in RSS feeds (OGC 2006). OpenLayers does not support an autorefresh function for GeoRSS version 2.7. Therefore, a patch[16] was implemented in OpenLayers in order to refresh GeoRSS layer automatically. Figure 4 shows real-time tracking targets displayed in the OpenLayers interface. In the prototype system, 10 tracking targets were tested in this research.

Track log management in the OpenLayers interface

OpenLayers is one of the most active open sources projects for Web mapping using AJAX (Asynchronous JavaScript + XML). It makes it possible for users to operate a map interactively and for efficient data transmission between the server and clients. The overlay function of OpenLayers supports various data types, such as Google Maps, Yahoo Maps, Web Map Service (WMS), Web Feature Service (WFS), and GeoRSS layers. WMS (OGC 2003) and WFS (OGC 2004) are standard protocols in GIS data communication, which were defined by the Open Geospatial Consortium (OGC).[17] In addition, the International Organization for Standardization (ISO) accepted WMS as ISO 19128 (ISO 2005). In the prototype system, Google Maps (vector and satellite maps) and OpenStreetMap[18] were used as the background layer, and 1/25,000 scale road network layer using the Orkney GIS Data Pack 2007[19] can be

[13] http://www.epsg.org
[14] http://trac.osgeo.org/proj
[15] http://georss.org
[16] http://trac.openlayers.org/ticket/1064
[17] http://www.opengeospatial.org
[18] http://www.openstreetmap.org
[19] http://www.orkney.co.jp/english/data/index.html



Base Layer
○ Google Streets
● Google Satellite
● OpenStreetMap (Mapnik)
● Kiban 2500 (WMS)
Overlays
☐ Road Network (WMS)
☐ SP Results
☑ Raw Points
☐ DP Points

GPS Track Log Search

IP Address: All

Date: 2008-06-09

Time Between: 0AM To 11PM

PDOP < 1.5

GPS Mode 3D

DP Tolerance 10m

SP None

Display GPS Track Log

>> Track Registration
>> GPS Data Upload
>> Log Statistic Page

Download KML Track Log

GPX
Download GPX Track Log

Display Track Log using Google Earth

Fig. 5 OpenLayers interface

overlaid through WMS connection, which is provided by a remote server.

The prototype system allows users to query and display track logs on-the-fly by selecting date, time, IP address of GPS tracking components, GPS positioning status (e.g., 2D, 3D fix. and no fix mode) and Horizontal Dilution of Precision (HDOP) on the right-side of the OpenLayers interface (Fig. 5). Figure 5 shows the search result, displayed with a Google vector map in the interface. The data are shown in different colors on the interface. In addition, the system allows the selection of parameters for a simplification function and a routing line. These functions are explained in "GPS error mitigation functionalities."

The prototype system supports non-real-time GPS logs; for instance, the users can use GPS data loggers and then upload their GPS logs to the server when they get online.

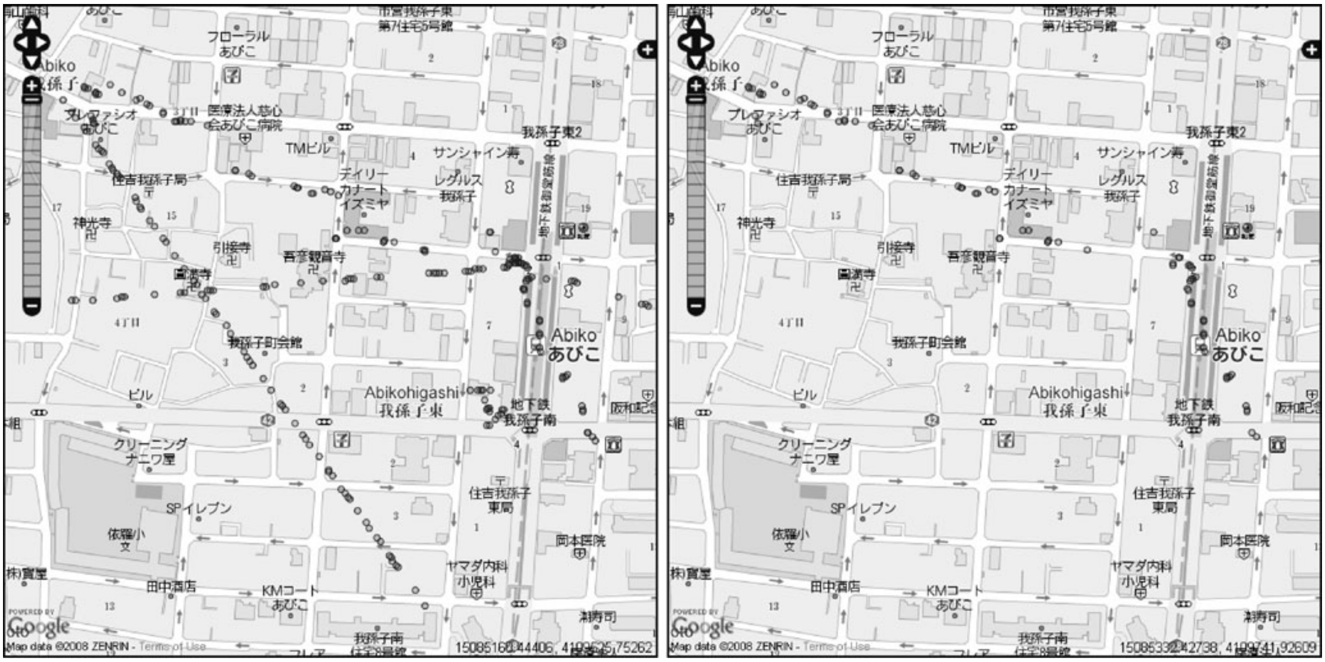


Fig. 6 Quality filtering result in the OpenLayers interface

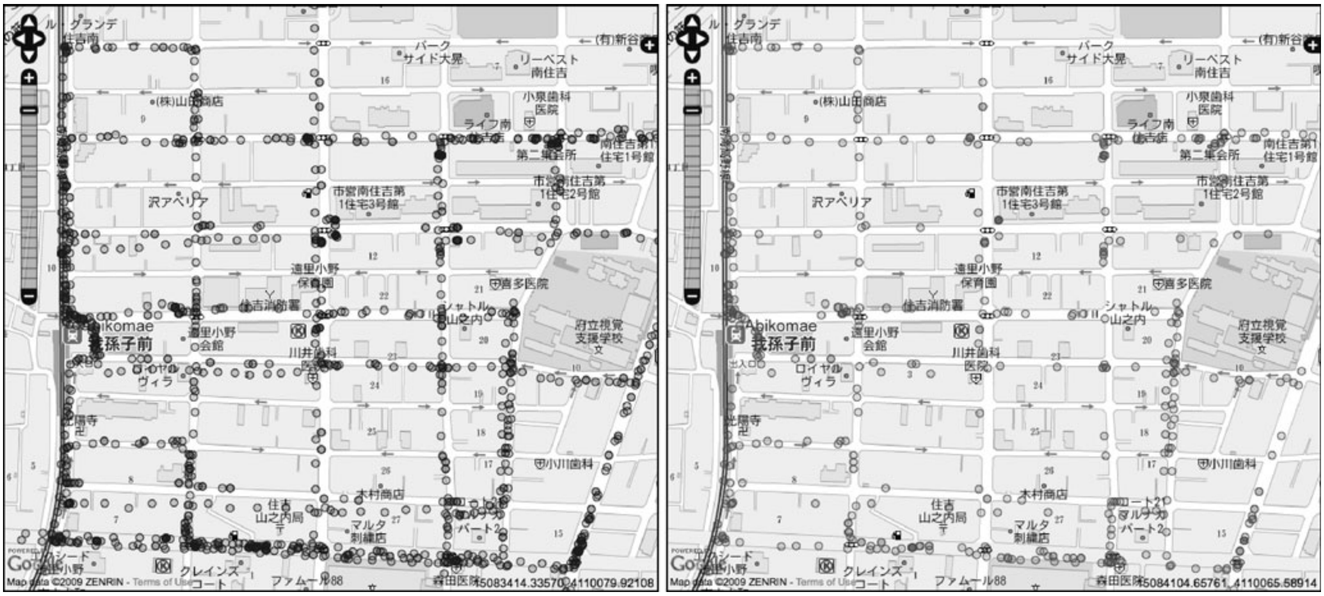


Fig. 7 Simplification result

The log upload interface is linked from the OpenLayers interface. The GPS logs are received as batch processes that are parsed and archived in the server.

The users can register the search result of track logs in the database from the link on the right side of the interface. The track registration function allows the users to register the metadata of the field survey along with track logs such as the survey name, place, and description. Further, the users can search the survey data and display the route via the Web interfaces.

A track log statistics function was also implemented to show the number of log points per day and hour as real-time graphs using PHP/SWF Charts.[20] It is useful to check visually GPS activity on a particular day or hour through this function. For example, when users are not sure what day or time that GPS tracks were recorded.

Track logs displayed in the OpenLayers interface can be exported in several formats such as KML and GPS eXchange Format (GPX).[21] The exported data can be used in other GIS applications such as desktop GIS and virtual globe viewers. Furthermore, Google Earth 3D viewer function was also implemented in the OpenLayers interface using Google Earth API.[22] The data export function will be explained in "POI management functionality."

GPS error mitigation functionalities

Generally, GPS receivers record data with variable accuracy and noise, for example, low accuracy or noise data. Dilution of Precision (DOP) provides a simple characterization of the geometric strength of satellite configuration on GPS accuracy. The more favorable geometry, the lower the DOP value returns. Wider angular separation between the satellites visible by the GPS receiver gives better quality position estimates (Misra and Enge 2004). Lima and Ferreira (2008) also used a similar approach to this research in order to enhance the precision of GPS data using DOP filtering. The prototype system can also have GPS quality filtering with GPS positioning status such as 2D, 3D fix, or no fix mode on the Web map. GPS measurement generally needs four satellites to calculate a 3D fix position, which includes altitude. The calculation using three satellites performs 2D fix positioning. Figure 6 shows the effectiveness of using the quality filtering. A lot of no fix data appeared as circles in the result (left figure), and the no fix data were removed after quality filtering (right figure).

In this research, a simplification function was implemented using the Douglas-Peucker algorithm (Douglas and Peucker 1973). The algorithm can reduce the number of points in lines to make lines smoother. It requires a threshold parameter for selecting the appropriate points for simplification. In particular, the algorithm works recursively, by at first finding the most significant point of a line and then subdividing the line at that point into two lines, which are then processed accordingly. The point is selected for being kept in the final result if the distance between the line connecting the two endpoints of the line exceeds the specific threshold (Foerster 2010).

Lima and Ferreira (2008) also adopted this algorithm to produce simplified lines using accumulated GPS track logs. The function simplifies points based on a given tolerance parameter, and it is also effective for reducing duplicated

[20] http://www.maani.us/charts/index.php

[21] http://www.topografix.com/gpx.asp

[22] http://code.google.com/intl/en/apis/earth

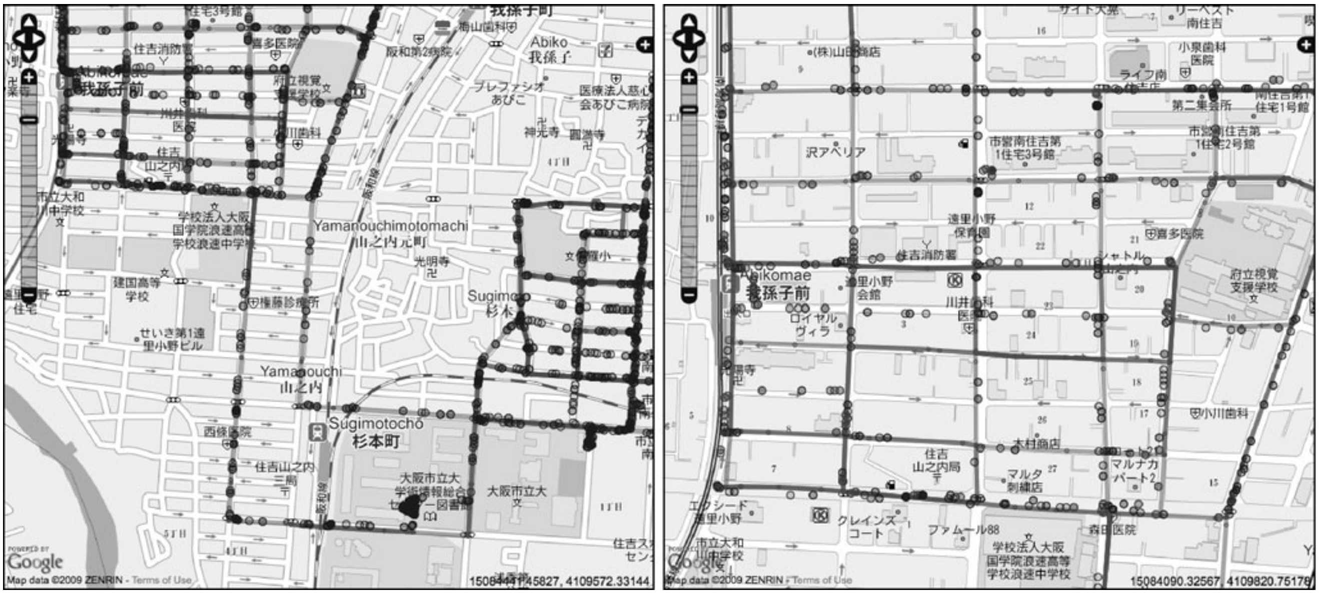


Fig. 8 Result paths using PgRouting

points. Figure 7 shows the simplification result. The left figure is before the simplification process, and the right figure is after the simplification, which highlights that a number of points were successfully reduced. This function not only allows to archive data efficiency but also reduces data noise.

GPS measurement includes errors such as satellite and receiver clock error, satellite ephemeris error, ionospheric propagation delay, tropospheric propagation delay, receiver noise, and multipath. Most of the errors can be mitigated by Differential GPS (DGPS) measurement except receiver noise and multipath. Multipath is caused near high buildings or under forest canopy. The signal may reach an antenna via two or more paths that are reflected from structure in the vicinity and from the ground (Misra and Enge 2004).

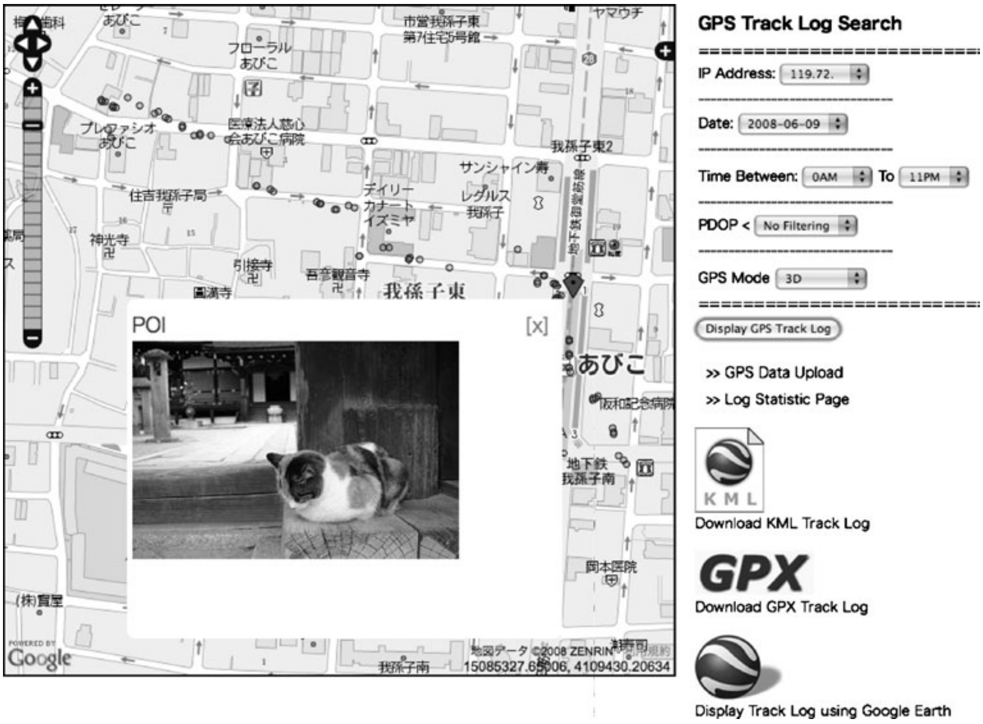
In this research, two approaches were implemented in the prototype system. One is a map-matching function for real-time tracking. The other is a shortest path function using PgRouting in order to display an accurate path in the OpenLayers interface. Under weak signal conditions, a position tends to be indicated far away from the actual position. The map-matching function eliminates erroneous GPS positions by snapping the GPS track to the nearest road line. There are several techniques to apply map-matching (e.g., geometric, probabilistic, and fuzzy logic). There are also some algorithms in the geometric approach [e.g., point-to-point, point-to-curve (link), and curve-to-curve (Schiller and Voisard 2004)]. In this research, the point-to-curve geometric technique was implemented using PostGIS since it already has some linear referencing functions. PostGIS functions, which were used for the map-matching function, are listed below:

- find_nearest_link_within_distance[23]—this function is part of PgRouting, which was developed by Orkney Inc., Japan, retrieves a nearest line node from a given point within certain designated area
- line_locate_point—a default PostGIS function, which returns a float between 0 and 1 representing the location of the closest point on a line string which is retrieved by *find_nearest_link_within_distance* to the given point
- line_interpolate_point—a default PostGIS function, which interpolates a point along a line based on the result value of *line_locate_point*

These PostGIS commands were used in the Java program. Yoshida and Raghavan (2008) described the technique in more detail.

Yoshida and Raghavan (2008) adopted the above map-matching approach for displaying the track-log function in the OpenLayers interface as well as real-time tracking. The map-matching approach works well in uncomplicated areas, but in complicated areas such as urban intersections, problems sometimes occur. In this research, the PgRouting shortest-path function and *find_node_by_nearest_link*. were implemented in the prototype system. The PgRouting shortest-path uses the Dijkstra algorithm, which is a well-known algorithm implemented in car navigation systems and other navigation services. The *find_node_by_nearest_link* PostGIS function retrieves a line node by the nearest link from a given point. The prototype system uses simplified points for finding starting nodes and ending nodes using the PostGIS function and processing the shortest paths. Figure 8 shows the result of these paths.

[23] http://pgrouting.postlbs.org/browser/trunk/core/sql/matching.sql



GPS Track Log Search

IP Address: 119.72.

Date: 2008-06-09

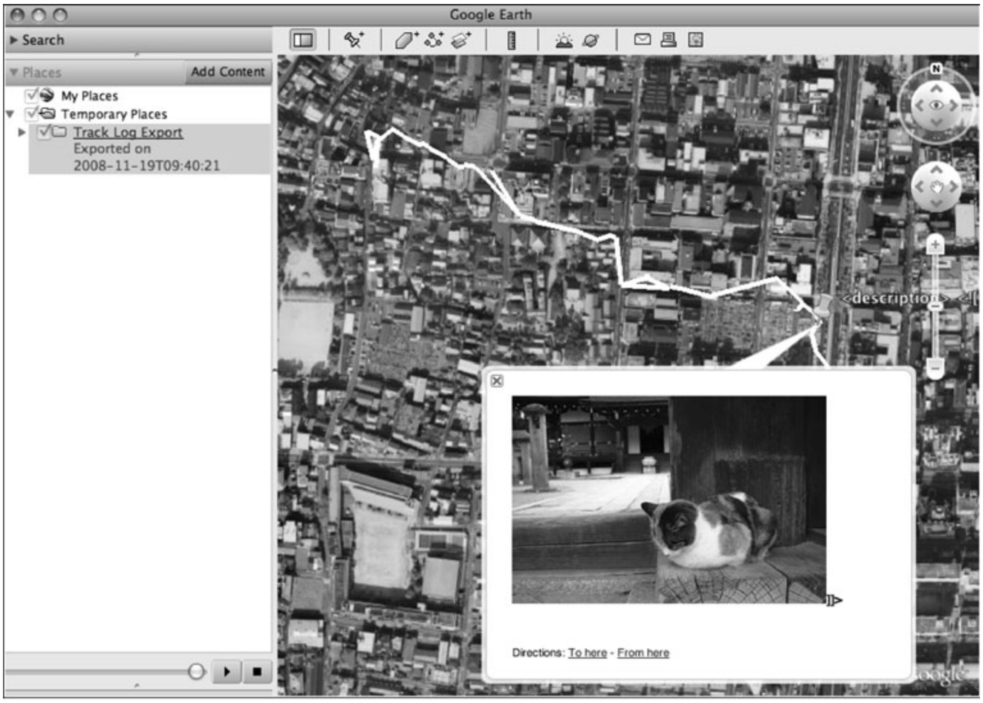
Time Between: 0AM To 11PM

PDOP < No Filtering

GPS Mode 3D

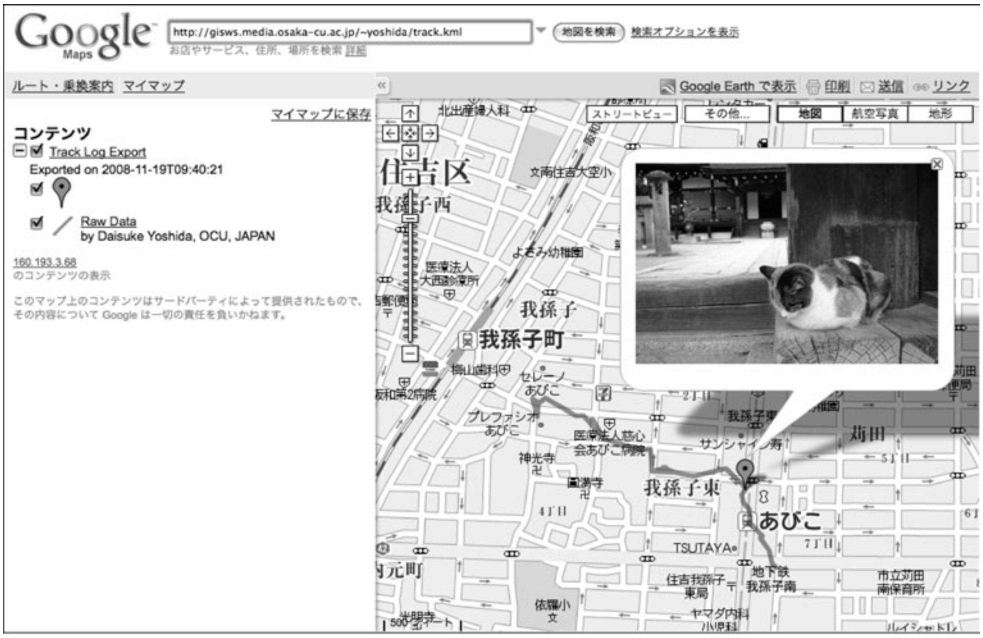
Fig. 9 POI display in the OpenLayers interface

The right figure is a zoomed image of the top-left area, showing that the paths at intersections were successfully drawn. However, more advanced techniques or additional support data such as differential GPS are necessary to handle more complicated case, for example, under high buildings with narrow streets and many intersections. Song et al. (2010) developed an advance-filtering method that implements the Hausdorff distance (Gao and Leung 2002) and Frechet distance (Alt and Godau 1995) algorithms. The article describes that the method can extract accurate GPS traces in complicated area. The implementation of the method as a Web service would be useful for general users to utilize GPS traces for various purposes.



Exported on
2008-11-19T09:40:21

Fig. 10 Displaying exported KML using Google Earth



http://gisws.media.osaka-cu.ac.jp/~yoshida/track.kml

コンテンツ

☑ Track Log Export
Exported on 2008-11-19T09:40:21

☑

☑ Raw Data
by Daisuke Yoshida, OCU, JAPAN

160.193.3.68
のコンテンツの表示

このマップ上のコンテンツはサードパーティによって提供されたもので、その内容について Google は一切の責任を負いかねます。

Fig. 11 Displaying exported KML using Google Maps

POI management functionality

It is easy for users to make POI databases for their various purposes since GPS-enabled cell phones and digital cameras are widely used by the general public. In addition, many Web applications for building POIs have been increasingly popular and also implementing Web mapping functions, such as Flickr[24] and Google Picasa Web Album.[25] These Web services provide various Web Application Programming Interface (API).

Taking pictures using GPS-enabled cell phones or cameras can save location information in an Exchangeable Image File Format (EXIF) image file, a process called geo-tagging. EXIF is standard for storing interchange information in image files such as date, time, shutter speed, and focal ratio. Horning et al. (2008) introduced a type of geo-tagging application for monitoring the environment. Katsura et al. (2008) developed a mobile application for field surveying using Google Maps API and GPS-enabled cell phones. The system was designed for an application within cell phones and allowed users to upload geo-tagged images and metadata through their GPS-enabled cell phones. This application also shows the registered POIs in Google Maps. The system also supports embedding location information into the EXIF portion of non-geo-tagged images.

The prototype system of this research can synchronize non-geo-tagged POIs with GPS track logs based on the common date and time attributes. Figure 9 shows a POI marker and the image in a popup box. Hence, the users are not required to carry special devices for field surveys, for example, professional GPS, GPS-enabled cell phones, or cameras. The system also supports geo-tagged images to make use of the embedded location information.

The system provides a POI management function that not only can register POIs but also edits and deletes unnecessary information of POIs. For examples, users can check POIs with track logs on Web map after field suveying and edit the meta-information such as date, time, and description of photos if necessary. The system provides Web interfaces for all of the functions. Therefore, it enables multiple users to work at same time and also facilitate collaboration in collecting POIs and track logs.

The outcome of the search in the OpenLayers interface results in raw GPS points and POIs (as KML only) that can be exported as KML and GPX formats for further use on other GIS applications, as explained in the earlier section. Recently, many applications have supported KML, such as ArcGIS, Flickr, Yahoo! Pipes,[26] WorldWind,[27] and Microsoft Live Search Maps[28] since OGC has adopted OGC KML Encoding Standard (OGC 2007). Therefore, KML makes it possible to interoperate between different applications. Figures 10 and 11 show the examples of displaying exported track logs and POIs in Google Earth and Maps through KML.

Discussion

Along with recent LBS-related developments, systems for Volunteered Geographic Information (VGI) (Goodchild 2007)

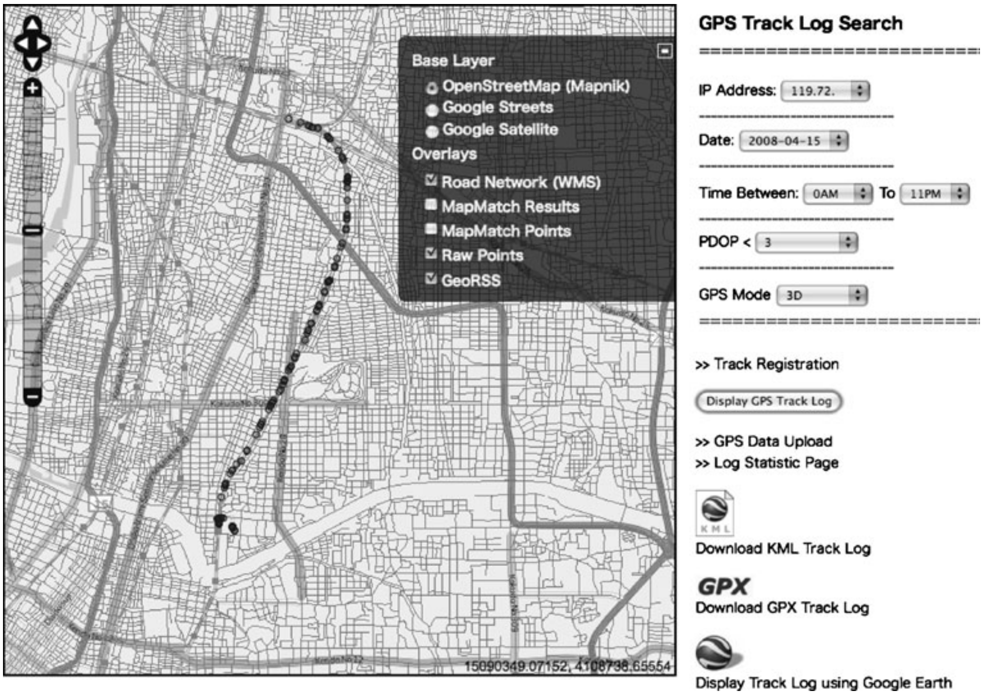
[24] http://www.flickr.com

[25] http://picasa.google.com/features.html

[26] http://pipes.yahoo.com/pipes

[27] http://worldwind.arc.nasa.gov

[28] http://maps.live.com


Base Layer
○ OpenStreetMap (Mapnik)
● Google Streets
● Google Satellite
Overlays
☑ Road Network (WMS)
□ MapMatch Results
□ MapMatch Points
☑ Raw Points
☑ GeoRSS

GPS Track Log Search

IP Address: 119.72.
Date: 2008-04-15
Time Between: 0AM To 11PM
PDOP < 3
GPS Mode 3D

>> Track Registration
Display GPS Track Log
>> GPS Data Upload
>> Log Statistic Page
Download KML Track Log
GPX
Download GPX Track Log
Display Track Log using Google Earth

Fig. 12 OpenStreetMap and WMS layer in the OpenLayers interface

or User Generated Contents (UGC) services, for example, OpenStreetMap and WikiMapia,[29] have been developed, and volunteers all over the world are increasingly contributing geographical content. VGI systems can greatly help in developing spatial databases, otherwise known as Spatial Data Infrastructure (SDI) (Global Spatial Data Infrastructure 2004; Yoshida et al. 2007), and are especially useful in developing nations with few up-to-date geographic data sets or under emergency situations that need the latest information after natural catastrophes. The prototype system manages GPS track logs and builds POI data sets with multiple users, and the data are interoperable with different applications through the use of open standard formats such as KML and GPX. Therefore, the system can be used as an interface for OpenStreetMap. There are tools for OpenStreetMap to edit GPX data before uploading, for example, desktop map editing tool, JOSM,[30] and online map editing tool, Potlatch.[31] However, these tools only support GPX format for the road and POI data. Most GPX data contain limited information in comparison with NMEA. For example, HDOP and GPS status data are eliminated when GPS-attached software converts the data format to GPX or during importing data from GPS receivers. The prototype system can contribute the data enhancement to OpenStreetMap utilizing quality information in NMEA.

Google has released My Maps service in Google Maps.[32] The users can make maps with lines and POIs in Google Maps and share with friends. Multiple users can collaborate to create maps together. Maps can also be imported and exported through KML. The service could be useful for field surveying. However, the service has some limitations which are listed below:

- My Map data cannot be totally restricted from the public. Many users have registered their own private data in My Maps service without fully understanding the Google policy and the data can been seen by other users.
- The service cannot be used for noncoverage areas of Google Maps, since the service uses only Google Maps as the base layer. There are still many places Google Maps does not cover, especially in remote sites and developing countries. This may be a problem, because field surveying is often in remote sites that Google Maps have not yet covered.
- GPX format is not supported by Google Maps service. After recording their field survey tracks using GPS, the users need to convert their GPX data to KML manually when they would like to use the GPS logs for My Maps service.

The prototype system in this research can overcome the above Google My Map problems by making use of OpenStreetMap. The system supports OpenStreetMap, other WMS

[29] http://www.wikimapia.org

[30] http://wiki.openstreetmap.org/index.php/JOSM

[31] http://wiki.openstreetmap.org/index.php/Potlatch

[32] http://local.google.com/support/bin/answer.py?hl=en&answer=68480

layers, and also Google Maps by displaying track logs and POIs. Figure 12 shows the search result displayed with OpenStreetMap as the background layer and overlaid with WMS road layer in the OpenLayers interface. Users recorded the track logs and POIs with a GPS logger during the surveys and uploaded the data to the prototype system. Then the users can register the metadata of the surveys as already explained in "Track log management in the OpenLayers interface" section (e.g., the survey name, place, and description, along with tracks and POI data in the database). Thus, the track registration function allows users to manage their own maps for field surveys in the system or they can upload their maps to OpenStreetMap.

GPS quality filtering and map-matching functions were implemented for minimizing GPS errors in the prototype system. However, the errors cannot be mitigated in some cases, for example, around high objects, complicated streets, and intersections. Higher accuracy positioning data and utilizing other aid information such as speed and heading direction are needed to be incorporated in the system to cope with the errors. However, most of the implementations are done as stand-alone applications, so it is difficult for users to receive these contributions easily. Therefore, a further step in this research will focus on augmenting the error mitigation function and implementing it in the Web-based system.

Conclusion

In this research, the Web -based prototype system for GPS track log and POI management was developed in order to achieve an effective framework for field surveying. The track log management function provided an interactive interface for log management and data enhancement. The POI management function facilitates a collaborative framework for building POIs with multiple users together through the Internet. This article also showed that functions of the system successfully provided an interoperable environment between other geospatial applications through open standards.

The article explained the possible usage of the system as an effective tool for field surveys using OpenStreetMap and also discussed the advantages and limitations of currently available proprietary open services, such as Google geospatial services.

The research also focused on expanding a real-time data collection function for future ubiquitous environments. The system supports real-time and non-real-time tracking targets using low-cost GPS receivers or data loggers, which were tested using both GSM and CDMA wireless networks. The research outcome contributes to various location-based data collection projects such as OpenStreetMap or other UGC services.

Acknowledgement The authors wish to thank Mr. Ned Horning, American Museum of Natural History, and Professor John Wilson, University of Southern California, for providing critical advice and suggestions. The authors would like to express their gratitude to Ms. Naomi Mills and Ms. Michelle Graves, visiting professors to Tezukayama Gakuin University, for carefully reviewing this paper.

References

Alt H, Godau M (1995) Computing the Frechet distance between two polygonal curves. Int J Comput Geom Appl 5:75–91, http://www.cs.uu.nl/people/marc/asci/ag-cfdbt-95.pdf. Accessed 2 Jun 2010

Derekenaris G, Garofalakis J, Makris C, Prentzas J, Sioutas S, Tsakalidis A (2001) Integrating GIS, GPS and GSM technologies for the effective management of ambulances. Comput Environ Urban Syst 25:267–278

Douglas DH, Peucker TK (1973) Algorithms for the reduction of the number of points required to represent a line or its caricature. Can Cartographer 10(2):112–123

Foerster T (2010) Web-based architecture for on-demand maps—integrating meaningful generalization processing. Dissertation, University of Twente. http://doc.utwente.nl/70642/. Accessed 2 Jun 2010

Gao YS, Leung MKH (2002) Line segment Hausdorff distance on face matching. Pattern Recognit 35:361–371

Global Spatial Data Infrastructure (2004) The SDI Cookbook. http://www.gsdi.org/gsdicookbookindex.php. Accessed 28 Jun 2009

Goodchild M (2007) Citizens as sensors: Web 2.0 and the volunteering of geographic information. Geofocus 7:8–10

Horning N, Ersts P, Thinh NV (2008) Using open source software to support photo monitoring of environmental change. In Proceedings of International Symposium GIS-IDEAS2008:173–178

ISO/TC211 (2005) ISO19128 Geographic information—Web map server interface. http://www.iso.org/iso/en/CatalogueDetailPage.Catalo gueDetail?CSNUMBER=32546&ICS1=35&ICS2=240&ICS3=70. Accessed 28 Jun 2009

Katsura S, Ninsawat S, Raghavan V (2008) Implementing mobile geotagging application using free and open source software. Geomatics 19:144–145

Lima F, Ferreira M (2008) Mining spatial data from GPS traces for automated map generation. In Proceedings of International Conference FOSS4G2008:116. http://conference.osgeo.org/index.php/foss4g/2008/paper/view/328. Accessed 28 Jun 2009

Misra P, Enge P (2004) Global positioning system. Ganga Jamuna, Massachusetts

Nguyen HB, Raghavan V, Ninsawat S, Shibayama M (2006) Implementing spatially enabled portal and content management systems. In Proceedings of International Symposium GIS-IDEAS2006. http://wgrass.media.osaka-cu.ac.jp/gisideas06/viewabstract.php?id=165. Accessed 28 Jun 2009

Open Geospatial Consortium (2003) OpenGIS Web Map Service (WMS) Implementation specification. http://www.opengeospatial.org/standards/wms. Accessed 28 Jun 2009

Open Geospatial Consortium (2004) OpenGIS Web Feature Service (WFS) implementation specification. http://www.opengeospatial.org/standards/wfs. Accessed 28 Jun 2009

Open Geospatial Consortium (2006) OGC white paper an introduction to GeoRSS: a standards based approach for geo-enabling RSS feeds. http://portal.opengeospatial.org/files/?artifact_id=15755. Accessed 28 Jun 2009

Open Geospatial Consortium (2007) OGC KML. http://www.opengeospatial.org/standards/kml/. Accessed 28 Jun 2009

Panahi S, Delavar MR, Shiran GhR, Mostafavi MA (2008) Dynamic shortest path in emergency vehicle routing using GIS. In Proceedings of International Conference Health GIS 2008:67–70

Schiller J, Voisard A (2004) Location-based services. Elsevier, San Francisco

Song X, Raghavan V, Yoshida D (2010) Matching of vehicle GPS traces with road networks for precise vehicle tracking and navigation in urban environment. Current Science 98(12):1592–1598

Yoshida D, Ninsawat S, Raghavan V (2007) Service oriented geospatial data management using free and open source software—a prototype for Northern Region of the Philippines. Int J Geoinformatics 3:9–18

Yoshida D, Raghavan V (2008) Development of real-time tracking and log management prototype system using free and open source software. In Proceedings of International Conference FOSS4G2008:30. http://conference.osgeo.org/index.php/foss4g/2008/paper/view/138. Accessed 28 Jun 2009