http://dx.doi.org/10.5392/IJoC.2012.8.1.055

Geospatial data Acquisition Using the Google Map API

Seung-Hee Han
School of Civil & Environmental Engineering, Urban Future Institute of Convergence Technology
Kongju National University, Cheonan, Rep. of Korea, shhan@kongju.ac.kr

Jin-Duk Lee
School of Civil & Environmental Engineering
Kumoh National Institute of Technology, Kumi, Rep. of Korea, jdlee@kumoh.ac.kr

Hyo-Beom Ahn
Division of Information & Communication Engineering
Kongju National University, Cheonan, Rep. of Korea, hbahn@kongju.ac.kr

ABSTRACT

Most Korean and overseas major portal sites include map servers to provide map services, and offer open APIs to allow their users to make use of maps or spatial information directly. At the early design stage, geographic spatial data do not tend to require high accuracy, and thus there would be no problem using data which have been obtained and then utilized through map servers provided by portal sites. This study has chosen the shortest route between starting point and destination, using GIS techniques. Moreover, for the chosen route, it also has calculated the elevation for the cross-section, using Google map and GPS measurements. This study aims to create APIs, which can extract vertical profile of routes from the Google Map server, by using JAVA, and to compare centerline profile results obtained by GPS(Global Positioning System) to explore their utilize abilities. The result demonstrated a height error of 0.5 to 1 m, compared to the GPS results, but they were mutually satisfactory. In short, the data extracted in this study are useful for centerline profile drawings in selecting routes, such as streets, Olle roads, and bike lanes.

***Keywords**: Google map API, Route profile, Geospatial data acquisition, GPS*

1. INTRODUCTION

In the past, GIS has been allowed to be used on the web, only when access rights and usage rights were granted to consumers. Lately, however, changes are being made towards web-based spatial information that can generate direct spatial objects or provide editing or direct saving functions to general users in a client environment. Web portals that provide such web-based spatial information do have map servers and also offer APIs that would allow users to make use of application programs. [1],[2]

Google Map has begun the service (http://maps.google.co.kr/) in Korea and raised its popularity of map data to general users. In the future, it should be focused on the service based on the search portal as well as the location. With the strategy for occupying the GIS markets, it has a plan to open the Map API in Naver, Daum, and Yahoo. Even the researches about what the users are highly able to apply the spatial information for various preliminary plans using variable Map API service are gradually increased, it is unusual for the researches of actual applications.[2],[3]

They are good enough for general purposes, but if users want to use the data for professional engineering and designing projects, they may not be satisfactory. For instance, data can be used for selecting Olle roads and bike lanes for leisure purposes. This study demonstrates the usability of the Google Map Server and application API by using them to create centerline profile of bike lanes and comparing them to GPS measurements.

2. STUDY DETAILS

The basic information is necessary for designs and plans in the case of city planning and construction development. The executional design requires highly accurate spatial information but the planning stage can be operated with low accuracy. If we can directly extracted from the information which is already serviced in the web site, the costs and expenses for collecting

This is an excellent paper selected from the papers presented at ICCC 2011.
* *Corresponding author. E-mail : hbahn@kongju.ac.kr*
Manuscript received Jan. 20, 2012 ; accepted Mar.20, 2012

the data can be innovatively decreased. For example, there are Aerial Ortho photo service and Road view from Google Earth and Daum. Of course, we have to consider the allowable precision according to the using purposes. This study is to produce the centerline profile for the selection of bicycle routes after extracting the 3D spatial information (X, Y, H) using from API service of Google Earth.

The purpose of this research is to present the ways to utilize the findings comparing the outcome of VRS continues mode to further evaluate the precision of centerline profile. Refer to the Fig.1 to refer to the overall research workflow.

Nodes were expressed by each intersection and the Euclidean shortest path was selected from sample networks of roads which are connected between nodes so that 3D coordinate was obtained by GPS measurements for its path. VRS continues mode was used for GPS measurements. The elevation was requested to Google Map Server through customized Google API after converting 2D (X,Y) coordinate to the longitude and latitude coordinate (λ , φ). The responded elevation data as well as longitude and latitude coordinate were converted to the coordinate of ITRF2000(X,Y,H) again.[4] As the results, 3D coordinates obtained from GPS measurements and extracted from Google Map information were acquired and the centerline profile was drawn up and compared using the accumulated distance and the altitude information.

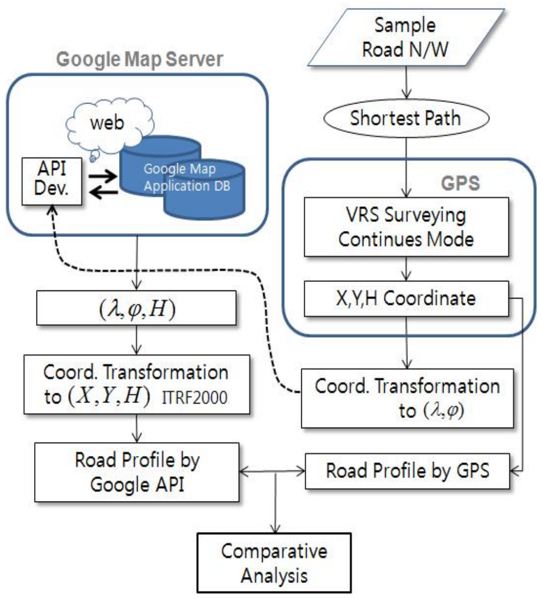


Fig.1. Step of profile drafting 1

3. ELEVATION API OF GOOGLE

The Google Elevation API provides a simple interface to query locations on the earth for elevation data. Moreover, you can request sampled elevation data along paths, and then calculate elevation changes along routes.

The elevation API provides altitude data about all ground locations as well as ocean depth locations which restore the negative numbers. Also it interpolates and restores the average used for the most four adjacent locations in the case that Google does not have the accurate altitude measurements for the accurate locations requested by users.

The Google Elevation API allows you to develop hiking and biking applications, mobile positioning applications or low resolution surveying applications. Although you can only query the elevation of up to 25,000 locations in each request per day with the Google Elevation API, since you can always extract elevations at regular intervals and create centerline profile, it is sufficient enough. The elevation API can only use for signing the results on the Google Map. Because it is impossible to use the altitude data without signing on the target map requested with the altitude data, it is also appropriate to apply the altitude data extracted from Google to high resolution aerial photo provided from Daum (www.daum.net).

The Google Elevation API returns elevation data for locations on the earth. Its location data allow us to request elevation data for a specific location or path. The Elevation API returns elevation data for a specific location or path per requested interval. In other words, we can pass any number of multiple coordinates within an encoded polyline. However, if we pass multiple coordinates, the accuracy of returned data might be of lower resolution than when we request data for a single coordinate. Returned data can be requested in the JSON format and XML format, based on our choice. The sample formats are provided below (for a location):[5],[6]

(JSON format request)
http://maps.google.com/maps/api/elevation/**json**?locations=36.7919444,127.1050|36.800555,127.1027778&sensor=false
(XML format request)
http://maps.google.com/maps/api/elevation/**xml**?locations=36.7919444,127.1050|36.800555,127.1027778&sensor=false

Fig.2. Example of Elevation request (JSON format and XML format)

3.1 The altitude request

The altitude request cannot be done by independent points and can be requested by constant intervals to the path comprised of starting and ending points. When the altitude is requested, the converting formats can be requested with JSON and XML formats. Due to the inconvenience of JSON format by taking the program of scripted, the altitude was obtained by parsing with XML format. Following responses are acquired individually.[5]-[8]

3.1.1 Location appointments

The location request is showed through locations parameters that indicate the altitude request for a transferred special location to the values of latitude/longitude. Unless it exceeds the service quota, various coordinates can be transferred within aligned or encoded polylines, as expected numbers, for producing the effective URL. The accuracy of data restored when various coordinates are transferred could have low resolutions compared to the case of single coordinate requested.

3.1.2 Path appointments

Like the location request using location parameters, path parameters designate the group of latitude and longitude values. Unlike the location request, however, it appoints a group of the angular points assigned by the order of path. The path request can not only restore the altitude data of the angular points but get sampling by the path distances based on the number of designated samples (including ending points).

path parameter
Path Parameters
path=40.714728,-73.998672|-34.397, 150.644
path=enc:gfo}EtohhU

```
< <?xml version="1.0" encoding="UTF-8" ?>
- <ElevationResponse>
<status>OK</status>
- <result>
- <location>
<lat>36.7919444</lat>
<lng>126.1050000</lng>
</location>
<elevation>-15.9907179</elevation>
<resolution>610.8129272</resolution>
</result>
- <result>
- <location>
<lat>36.8005550</lat>
<lng>127.1027778</lng>
</location>
<elevation>39.1829948</elevation>
<resolution>152.7032318</resolution>
</result>
</ElevationResponse>
```

Fig.3. Example of elevation response(XML format)

3.2 The altitude response

Figure 3 shows the example of restore results for responding XML format. The results of restored values indicate the values of longitude, latitude, and altitude for appointed locations together. This study is to designate two points (latitude and longitude) and calculated the number of sampling points within the targeted paths. The altitude value is restored in the linear path by the appointed numbers as long as the sample numbers are designated. The following is divided the latitude of the starting point into the latitude/longitude of the ending point to estimate the altitude of the path and the numbers of the altitude for the path were input. .[5]-[8]

http://maps.google.com/maps/api/elevation/xml?path=Lat.&Longitude Coordinate of Starting point | Lat. & Longitude Coordinate of End point &samples=the number of Sample&sensor=false

This study estimated the altitude after treating XML results requested from XML format.

3.3 The altitude extraction of the path

The method to use the altitude API of Google can be obtained by the parsing process of the results requested by URL with HTTP protocol. Following three steps were treated to extract the altitude of the path.

- Step 1. Send a request to the altitude API
- Step 2. Parsing XML documents restored by the requested results
- Step 3. Extract the values of latitude, longitude, and altitude from the parsed results.

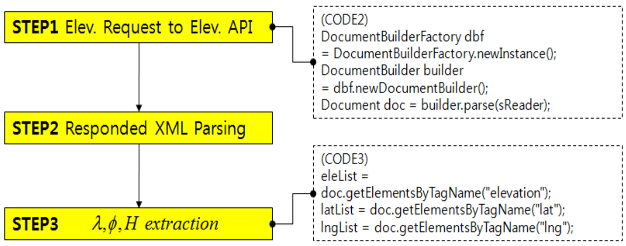


Fig. 4. Elevation extraction step for path

JAVA language was used for extracting the altitude of the path. URL class was used for the altitude API and "java.net.URL was also used for this class. The following code is to treat the results after requesting URL and by InputStream.

```
URL java = new URL(url);
InputStream sReader = java.openStream();
```

To request the altitude API, the results from the altitude API were saved into InputStream after saving Instance of URL class produced in Java variables and after saving the written letter codes in front of URL variables.

```
DocumentBuilderFactory dbf
= DocumentBuilderFactory.newInstance();
DocumentBuilder builder
= dbf.newDocumentBuilder();
Document doc = builder.parse(sReader);
```

To treat the saved results, the Stream should have parsing encoded from DocumentBuilderfactory for parsing XML documents. Parse() function produce XML documents parsed from the stream and save this into Document class. Finally doc save XML into the parsed results and it can bring the values of latitude, longitude, and altitude with saving NodeList class as following method.

```
eleList =
doc.getElementsByTagName("elevation");
latList = doc.getElementsByTagName("lat");
lngList = doc.getElementsByTagName("lng");
```

The reference variable, eleList, is NodeList class that provide the list structure and it can save only the value of <elevation> tag using getElementByTagName() method. Figure 5 shows the structure of XML documents of the results restored.

Through NodeList class, the altitude saves into eleList, the latitude into latList, and the longitude into lngList. The result can be obtained by printing out this.

The revised altitude chart can be obtained from the CAD with comparing these data to the altitude from GPS again, and the altitude of bicycle road can be gained by this chart.

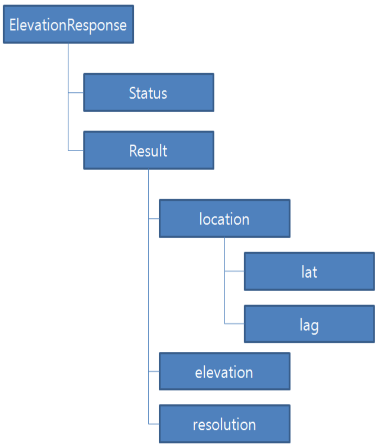


Fig. 5. XML structure from the altitude results.

Since the objective of this study is to extract elevations of routes, in using the Google Elevation API, we should obtain data by using the HTTP protocol and parse the URL requested results. In order to extract elevations of routes, the following steps should be taken.

4. DECISION OF TARGET PATHS AND GPS MEASUREMENTS

The path was selected on OD where is the traffic area of public transportation in CheonAn city and the altitude request was performed by choosing three Euclidean shortest path in road network. The altitude request interval of the path was conducted on each 5 ~ 10m for the comparison of GPS results. Route A covers about 2.8km from the Terminal to Dujeong station, Route B about 5.9km from the Terminal to the City Hall, and Route C about 10.4km from Kongju University Campus to KTX station at CheonAn-Asan. GPS measurements were estimated for the decided path. VRS continues mode was performed by driving 60km/hour.[4],[9]

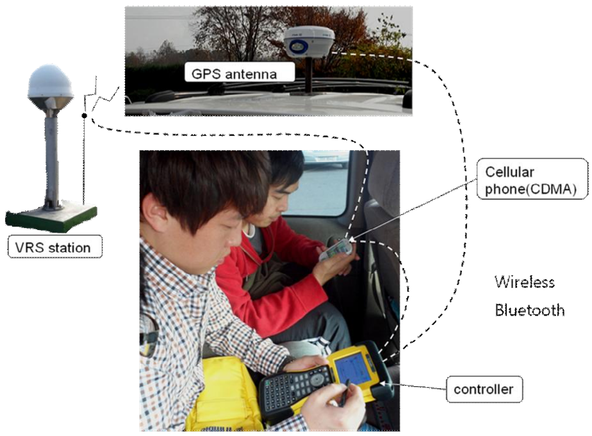


Fig.6. GPS Surveying(VRS)

The level measurements between antenna installed onto vehicles and the altitude of the ground for obtaining the altitude of the surface was 1.862m and it was input into GPS receiver with the same height of antenna. The coordinate system was used as GRS80 coordinate system and Geoid2000 was used for the altitude datum. The horizontal distance was calculated by using X and Y coordinate and the centerline profile was finished with the accumulated distance and the altitude data.[3]

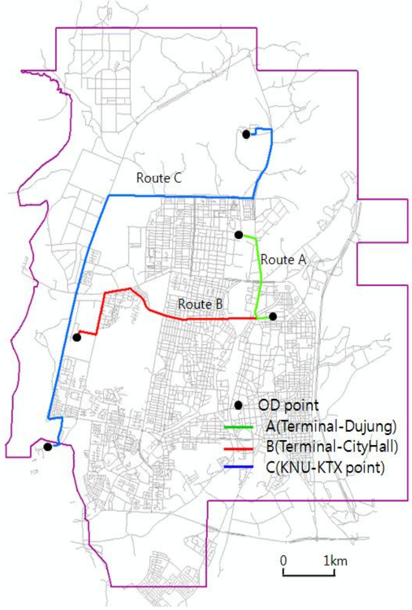


Fig. 7. Defined shortest route.

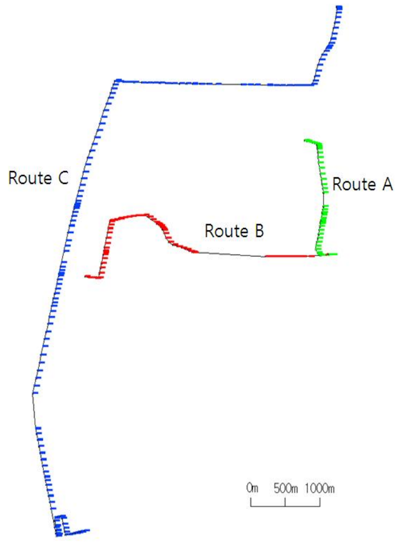
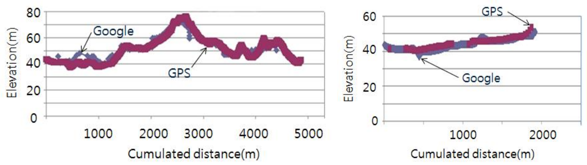


Fig. 8. GPS measurement points

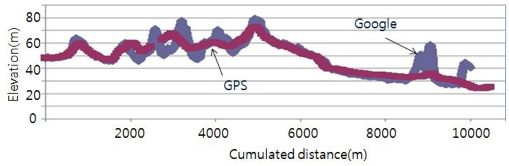
The differences between the shortest route selected from road networks and the route of GPS measurements are due by the revision of the construction of new roads and the inconvenience of bicycle passing. In addition, Route B missed the data caused GPS cycle slip phenomenon by passing through crowded building areas.

5. THE RESULTS OF CENTERLINE PROFILE

The path data from GPS had the altitude request to Google Map API after converting into the longitude coordinate. The centerline profile was finished by repeating GPS measurements after converting latitude, longitude, and altitude data gained from the altitude request into X,Y,H. It was unified into ITRF2000 coordinate via coordinate converting from Google API and GPS.



a) Route A(terminal-dujeong) b)Route B(terminal-city)



c) Route C(Campus to Cheonan-Asan KTX station)

Fig. 9. The centerline profile obtained from Google Map API and GPS measurements

In Figure 8a, the average of altitude deviations between two methods are 1.2~1.8m which is a little large values and it seems to be based on the altitude change by partial developments. Fig 8b is the shortest section and shows the largest altitude change at 50m point. That is the reason why pass the overhead bridge and shows an acceptable consistency with 0.5~1.1m in other sections. The longest path C indicated an suitable consistency at 1.9km and 4.6~8.3km but the path of 2-4.6km and 8.3-10.0km showed big differences due to construct the roads after setting existing mountain area into the transportation routes

As a result, all three paths has only 1m deviation difference within no geographical changes and showed an acceptable consistency, but there was a huge difference in the developed section by data renewal period of Google. It is expected that the results of this study have enough applicable into a stabilized city and no geographical rural area.

6. RESULTS

This study has drawn the following conclusion after extracting the road centerline profiles and comparing them against GPS measurements using Google API.

First, we were able to extract elevation information by inputting the paths from the sample routes as the requested data, and made centerline profile using the Google Map API.
Second, when we compared the values with the GPS measurements of the same routes, they were in the range of 0.5 and 0.7 m on average. Therefore, we concluded that they can be used sufficiently as low resolution elevation data.
Third, for a more practical study in the future, we can use them to plan routes (such as bike lanes) that would require low resolution elevation data, and carry out a comparative analysis of the results obtained using GPS and DEM data.
If APIs that allow people to utilize map servers provided by various portal sites are developed, and apps that allow people to obtain profiles on their smart phones are created, its applicability will be likely to increase.

REFERENCES

[1] Yong-Jae Park, Ki-Won Lee, "Mashup implementation for spatial feature creation using Google Map API", *Proc. The Korea Society for Geospatial Information System*, 2008, p. 73-75.

[2] H.K Kwon, J.C Kim, J.W Kim, Y.S Oh, T.H Kim, S.K Nam, "A study on facility management system based on USN using Open space", *Proc. . The Korea Society for Geospatial Information System*, 2009, p.306-306

[3] Jung-sik Kim, Ho-joon Kim, "Real-time location tracking system of moving vehicle using open map interface", *Proc. Korean Institute of Information Technology*, 2010, p. 578-582.

[4] Yong-Chang Lee " A study on the comparison of real-time GNSS Satellite surveying methods", Journal of Korea Society of Civil Engineers D, vol.28, no.4D, July, 2008, pp.579-586.

[5] Google. "Using JSON with Google Data APIs". http://code.google.com/apis/gdata/json.html. Retrieved July 3, 2009.

[6] Dae-Ki Kang, Piljoong Kim, "Social networking service on smartphone augmented with location information", *Proc. The Institute of Electronics Engineers of Korea*, 2010, pp. 1463-1465.

[7] http://code.google.com/intl/ko-KR/apis/maps/documentation /elevation/

[8] Google Maps JavaScript API V3 Reference Documents http://code.google.com/intl/ko-KR/apis/maps/documentation/javascript/reference.html#DirectionsDistance

[9] Sang-Kyu Jang, Jin-Soo Kim, Gong-Uhn Jeong, " Accuracy Evaluation of Boundary corner point using VRS GPS", Journal of Korea Society for Geospatial Information System, vol.17, no.1, Mar., 2009, pp. 37-42.

Seung-Hee Han
He received the B.S., M.S., and Ph.D. in Civil Engineering from Chungnam National university, Korea in 1984, 1987 and 1993 respectively. Since 1993, he has been with the Dept. of Civil & Environmental Engineering, Kongju National University, Rep. of KOREA. His main research interests include GIS and Digital Photogrammetry

Jin-Duk Lee
He received the B.Eng. in agricultural engineering from Chungnam National University, Korea in 1983 and also received M.Eng.., Ph.D. in civil engineering from Chungnam National University, Korea in 1986 and 1992 respectively. Since 1990, he has been with Department of Civil Engineering, Kumoh National Institute of Technology. His main research interests include geographic information system, digital photogrammetry, remote sensing and digital mapping.

Hyo-Beom Ahn
He received the B.S. in Computer Science and M.S., and Ph.D. in Computer Science and Statistics from Dankook University, Korea in 1992, 1994 and 2002 respectively. Since then, he has been with the Dept. of Information and Telecommunication, Kongju National University, Rep. of KOREA. His main research interests include Computer Networks, Network Security, Smart Grid Security and GIS information System.