TESTING goGPS LOW-COST RTK POSITIONING WITH A WEB-BASED TRACK LOG MANAGEMENT SYSTEM

E. Realini [a,] *, D. Yoshida [b], M. Reguzzoni [c], V. Raghavan [a]

[a] Graduate School for Creative Cities, Osaka City University, 3-3-138 Sugimoto, Sumiyoshi-ku, 558-8585 Osaka, Japan - realini@gscc.osaka-cu.ac.jp, raghavan@media.osaka-cu.ac.jp
[b] Faculty of Liberal Arts, Tezukayama Gakuin University, 2-1823 Imakuma, Osaka Sayama City, 589-8585 Osaka, Japan - yoshida@la.tezuka-gu.ac.jp
[c] Dept. of Geophysics of the Lithosphere - OGS, c/o Politecnico di Milano - Polo Regionale di Como, via Valleggio 11, 22100 Como, Italy - mirko@geomatica.como.polimi.it

Commission IV, WG IV/5

KEY WORDS: Web based, Cooperation, Software, GPS/INS, Accuracy, Real-time

ABSTRACT:

Location-based online collaborative platforms are proving to be an effective and widely adopted solution for geospatial data collection, update and sharing. Popular collaborative projects like OpenStreetMap, Wikimapia and other services that collect and publish user-generated geographic contents have been fostered by the increasing availability of location-aware palmtop devices. These instruments include GPS-enabled mobile phones and low-cost GPS receivers, which are employed for quick field surveys at both professional and non-professional levels. Nevertheless, data collected with such devices are often not accurate enough to avoid heavy user intervention before using or sharing them. Providing tools for collecting and sharing accuracy-enhanced positioning data to a wide and diverse user base requires to integrate modern web technologies and online services with advanced satellite positioning techniques. A web-based prototype system for enhancing GPS tracks quality and managing track logs and points of interest (POI), originally developed using standard GPS devices, was tested by using goGPS software to apply kinematic relative positioning (RTK) with low-cost single-frequency receivers. The workflow consists of acquiring raw GPS measurements from the user receiver and from a network of permanent GPS stations, processing them by RTK positioning within goGPS Kalman filter algorithm, sending the accurate positioning data to the web-based system, performing further quality enhancements if needed, logging the data and displaying them. The whole system can work either in real-time or post-processing, the latter providing a solution to collect and publish enhanced location data without necessarily requiring mobile Internet connection on the field. Tests were performed in open areas and variously dense urban environments, comparing different indices for quality-based filtering. Results are promising and suggest that the integration of web technologies with advanced geodetic techniques applied to low-cost instruments can be an effective solution to collect, update and share accurate location data on collaborative platforms.

1. INTRODUCTION

The latest efforts in technological advancement are gathering more and more computational capabilities, wireless connection tools and storage capacity on small devices such as smartphones, PDAs (Personal Digital Assistants), netbooks and UMPCs (Ultra-Mobile Personal Computers). Nowadays most of these devices incorporate also GPS chipsets, which allow users to interact spatially with the Web 2.0 by geolocating features and events; this merging between geographical and digital worlds is giving birth to what is commonly known as the GeoWeb (Leclerc, 2001). One of the main reasons that drive hardware producing companies to embed GPS chipsets on more and more mobile devices is to provide users with LBSs (Location Based Services) where wireless connectivity is available. These embedded GPS chipsets are usually defined “low-cost” in the geodetic community because they are compared to high level (and high cost) professional receivers, which generally provide accuracies of some centimeters in real-time navigation, while low-cost ones usually range from 10 m to 1 m. Anyway, apart from the cost issue, high level GPS receivers could not be embedded in mobile devices due to physical limitations that usually require miniaturization, while high precision antennas are often bigger than the mobile devices themselves. Therefore, in order to increase the localization accuracy of mobile devices, low-cost receivers have to be exploited at their maximum extent, also by supporting them with information coming from external sources.

1.1 Geospatial data and the Web 2.0

Tagging web contents with geographic information (i.e. geo-tagging) has become a common feature on most Web 2.0 applications. For example, online photo management and sharing applications like Picasa (http://picasaweb.google.com) and Flickr (http://www.flickr.com) georeference photos either automatically, using coordinates stored in EXIF data (metadata associated to digital photos), or by letting users locating them on a map. Also places and historical events described in Wikipedia (http://www.wikipedia.org) can have associated coordinates, that can be passed through GeoHack (https://wiki.toolserver.org/view/GeoHack) to dozens of other location-aware web services, to get more location-based information. Location can then be associated to many different

* Corresponding author. This is useful to know for communication with the appropriate person in cases with more than one author.

kinds of information made available on the web (e.g. blog posts, news, events, shops with the products they sell, etc.), bridging in this way the digital and real worlds. This kind of geo-tagging usually does not require high-accuracy location data, thus it is often sufficient to use a standard low-cost GPS device or even just clicking on a map. Nevertheless, there are some web applications for collaborative surveys, like for example OpenStreetMap (http://www.openstreetmap.org) that would benefit from getting higher accuracy GPS data, possibly with metadata about their accuracy level, in order to automatize as much as possible the map updating process and to provide end-users with an estimation of the level of error for the data they are going to use.

1.2 State-of-the-art of low-cost GPS positioning

Low-cost GPS positioning massively spread in the last few years thanks to seven technology enablers: assisted GPS, massive parallel correlation, high sensitivity, coarse-time navigation, low time-of-week, host-based GPS and RF-CMOS (Van Diggelen, 2009). These technologies allowed for faster time-to-first-fix (i.e. faster positioning once the hardware is switched on) and lower hardware cost, causing GPS chipsets to be embedded in almost all smartphones and in a conspicuous number of mobile phones. The tendency now is to move as much as possible the hardware components to software, in order to lower even more the bill of materials and exploit at best the increasing computation and memory capabilities of handheld devices (Söderholm et al., 2008). Moreover, future GPS devices are going to be more and more integrated with other technologies to overcome the limitations imposed by the urban/indoor environments where mobile phones are usually employed: Wi-Fi, MEMS (MicroElectroMechanical Systems), power measurements from GSM and 3G phones, and obviously other GNSS (GLONASS, Compass, QZSS and Galileo) (Van Diggelen, 2010).

Though low-cost GPS devices are traditionally designed to perform positioning in stand-alone mode, research is being carried out to evaluate their performance by real-time differential positioning with respect to a master station. This technique, known as Real-Time Kinematic (RTK), is usually employed for higher cost professional receivers, but there is interest in applying it also to low-cost receivers in order to remove systematic errors (Alanen et al., 2006; Wirola, 2008).

2. METHODOLOGY

This paper investigates three aspects of the improvement of low-cost GPS positioning:

1 - applying single-frequency (L1) RTK positioning to low-cost devices by goGPS open source software
2 - comparing host-based and remote (server-side) processing
3 - finding a suitable index for filtering out poorly positioned points

2.1 goGPS L1 low-cost RTK positioning

Low-cost GPS devices typically implement small patch or helix antennas and single frequency (L1) receivers functioning in stand-alone mode; moreover, they are often highly sensitive to GPS signal (even if degraded) in order to assure positioning in dense urban environments. The error budget can thus be identified both as a shift due to atmospheric delays and clock errors, not erased or negligible in stand-alone mode, and as noise due to the low quality of involved hardware and to multipath resulting from high sensitivity. The first part of the error budget can be removed by applying relative positioning with respect to a master station, which erases clock errors and makes atmospheric errors negligible if the rover is within 10 km from the master station. The most efficient way of doing this is to exploit RTK (Real Time Kinematic) services provided by a network of permanent GNSS stations, in particular using VRS (Virtual Reference Stations). The second part of the error budget cannot be systematically removed, thus it is needed to minimize it: the use of Kalman filtering (Kalman, 1960; Grewal and Andrews, 2001) with a proper observation weighting and also the exploitation of external information sources not directly related to GPS (e.g. height values from a digital terrain model if the receiver is on the ground, line constraints if it is moving on a path known a priori, etc.) can help in this sense. In particular observation weighting is based on signal-to-noise ratio, which is for instance degraded by multipath phenomena.

The basic way of using a Kalman filter for stand-alone GPS-only navigation is to apply it directly on the estimated positions. The positioning is typically computed by least-squares adjustment from code and phase measurements and the Kalman filter acts *a posteriori* with a smoothing effect on the estimated trajectory. This approach with low-cost devices results in a positioning as accurate as that provided by performing absolute positioning using phase-smoothed code measurements (5 - 10 m of error). This means that atmospheric delays and clock errors cannot be efficiently corrected, because this kind of receivers is usually not designed for relative positioning, and computed trajectories are often affected by large biases.

The Kalman filter implemented in goGPS is not applied on estimated positions, but on double difference observations with respect to a reference station (i.e. relative positioning). The possibility of directly processing GPS observations allows us to remove most of the bias associated to absolute positioning, obtaining accuracies of less than 1 m. Its use in real-time obviously requires a wireless connection to the Internet in order to receive the master station observations and a GPS chipset that provides raw measurements as output. The goGPS Kalman filter is “modular”, in the sense that it is conceived in such a way that it can be easily integrated with additional components. This means that new sources of measurements (i.e. observations) can be added or disabled without major changes in the algorithm. In its current state goGPS includes a first observation module applied to code double differences, a second one applied to phase double differences and a third one that exploits the information coming from a DTM (Digital Terrain Model).

goGPS main targets are single-frequency low-cost devices, but its code and phase modules are designed to work either in single-frequency or double-frequency mode. Therefore, it can be used also with double-frequency receivers, if it is needed. Finally, goGPS includes also an alternative version of its main Kalman filter algorithm, designed to obtain line-constrained positioning (e.g. to navigate on a network of roads). In this case the DTM information is not used anymore, because the constraint is already three-dimensional. For a detailed description of goGPS Kalman filter, see Realini (2009).

2.2 SoC, host-based, remote processing

Despite the latest technology trends are shifting from System-on-Chip (SoC) architectures to host-based ones, the majority of people (and some manufacturers) when dealing with GPS positioning engines still think in a SoC direction.

Though goGPS was not explicitly designed since the beginning with this distinction in mind, it has naturally become a host-based system for at least three reasons:

1 - it has been first developed as a tool for studying and teaching GPS positioning in a MATLAB environment
2 - applying RTK requires access to raw GPS observations, which means at least at GPS time, ephemerides, code pseudorange and carrier phase (and if possible additional information like signal-to-noise ratio, etc.)
3 - it has been developed with the concept of Free and Open Source Software (FOSS) and open data policy as its foundations. In order to let developers have the greatest freedom over how to implement their own positioning engines, GPS raw observations must be available and typically they will be processed on the host system.

The current limitation in using goGPS in real life applications lies in its programming language, usually limited to research and teaching environments. There are plans to port goGPS to a widespread language like C/C++ or Java, but until then the host system must be capable of running MATLAB.
A third option, explored in this paper, for processing GPS observations is to use GPS rovers as simple sensors that send raw data to a central server. The server would then do the processing and either send coordinates back to the rover or store/display them.

2.3 Web-based track log management system

goGPS can significantly mitigate GPS errors and enhance positioning accuracy. However, the software needs to be installed in PCs and the operations are cumbersome for users.
A web-based prototype system for track log management system was developed to support goGPS data management and also to provide collaborative frameworks for multiple users. The system allows users to query and display track logs interactively by selecting date, time, data source (by IP address of GPS receiver clients), GPS positioning status (e.g. 2D, 3D fix and no fix mode – see §2.3.1 for an explanation of these terms), number of satellites and Horizontal Dilution of Precision (HDOP) on OpenLayers (http://www.openlayers.org) web interface. OpenLayers is one of the most active open sources projects for web mapping using AJAX (Asynchronous JavaScript + XML). In this interface, Google Maps (vector and satellite maps) and OpenStreetMap road network were used as background layers.

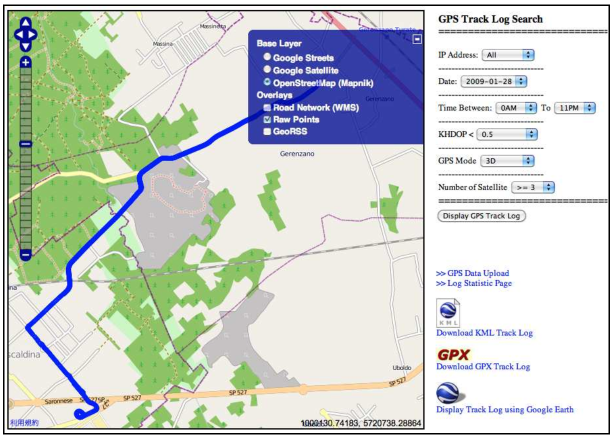

Figure 1. OpenLayers track log search interface

Additional data, as for example a 1/25000 scale Japanese road network layer (Orkney GIS Data Pack 2007), can be overlaid through WMS connection provided by a remote server. Figure 1 shows the web interface and displays an example search result of track log on the OpenStreetMap layer in Italy. Track logs displayed in the web interface can be exported in several formats such as KML and GPS eXchange Format (GPX). The exported data can be used in other GIS applications such as desktop GIS and virtual globe viewers.

2.3.1 Quality-based filtering: As explained in previous section, the web-based track log management system has on-the-fly functions to filter GPS data by quality related parameters such as number of satellites, Dilution of Precision (DOP) (Spilker, 1996) and fix mode. "Fix" is a term commonly used for consumer GPS devices that indicates the action of computing a position (e.g. "time-to-first-fix" or "TTFF" indicates the time the receiver needs to compute a position after being switched on). This is not to be confused with the fixing of ambiguities, which are typically referred to as "fixed" as opposite to "float". GPS measurement generally needs four satellites to calculate a "3D fix" position, which includes altitude. The calculation using three satellites performs "2D fix" positioning. Some receivers even record GPS data under "no fix" status (e.g. the quality of the signal was lower than a chosen threshold, so the positioning is labeled as "no fix") thus many noisy positions can be displayed on the web map. The web system provides an interactive web interface to check the parameters and results at the same time in order to remove such noisy data from the plot.

2.3.2 Kalman filter DOP: Using the HDOP as an index for filtering out poorly positioned points has the major drawback of taking into account only the satellite geometry. This information can be useful if the sky visibility condition is always the same, but in urban environments the actual positioning accuracy is highly dependent on the signal quality, especially when using high-sensitivity GPS devices. This means that the receiver can track a high number of satellites, maybe also with a good geometry, even in situations where the signal is highly degraded: in this case the HDOP would not be a good indicator of the actual positioning quality. To overcome this limitation, in goGPS we do not compute only usual DOP values from the coordinate co-factor matrix *Q*. This is obtained from the least-squares design matrix *A* as (Hofmann-Wellenhof et al., 2007)

$$Q = (A^T A)^{-1} \tag{1}$$

thus for example the traditional Position Dilution of Precision (PDOP) is

$$PDOP = \sqrt{q_X + q_Y + q_Z} \tag{2}$$

where q_X, q_Y and q_Z are respectively the X, Y and Z weights along the main diagonal of *Q*. In order to have indices that represent how the filter is performing, we exploit the coordinate error covariance matrix *C* estimated by the Kalman filter, i.e.

$$C = (I - GA)K \tag{3}$$

with *I* the identity matrix, *G* the Kalman filter gain matrix and *K* the coordinate error covariance matrix based on dynamics only. The PDOP value obtained from the Kalman filter coordinate error covariance matrix (from now on KPDOP) is computed as

$$KPDOP = \sqrt{c_X + c_Y + c_Z} \quad (4)$$

where c_X, c_Y and c_Z are respectively the X, Y and Z variances along the main diagonal of C.
As for the Kalman-based horizontal and vertical DOP values, respectively KHDOP and KVDOP, they are computed by following the same logic of traditional HDOP and VDOP, i.e. propagating the covariance from a global (X, Y, Z) to a local (East, North, Up) reference frame by the equation

$$C_{ENU} = RCR^T \quad (5)$$

where R is the rotation matrix from global to local frames, thus obtaining

$$KHDOP = \sqrt{c_E + c_N}, \quad KVDOP = \sqrt{c_U} \quad (6, 7)$$

where c_E, c_N and c_U are respectively the East, North and Up variances along the main diagonal of C_{ENU}.
In this way we obtain alternative DOP indexes that can better describe the positioning quality obtained by the Kalman filter. K*DOP values do not depend exclusively on satellite geometry, but also on the evolution of the filter itself, which includes also for example the variance increment for slipped satellites.

3. SYSTEM IMPLEMENTATION

goGPS and the web-based track log management system have been made interoperable by adding NMEA output capability to goGPS, by tuning the parsing algorithm on the web-based system (e.g. a goGPS customized NMEA sentence was added to the parser) and by developing client and server socket programs, in order to let goGPS receive raw data streams from a remote GPS receiver and process them at server-side. Both host-based and remote processing solutions are explored in this paper.

3.1 goGPS system architecture

goGPS (http://www.gogps-project.org) is a MATLAB open source software package that gets raw GPS data (code pseudorange, carrier phase, signal-to-noise ratio, ephemerides and timing) in input, processes them within a Kalman filter specifically developed to apply RTK and address low-cost GPS navigation issues, optionally includes external sources of data such as DTMs or line networks and gives enhanced positioning in output. goGPS receives data from a low-cost receiver through a COM port and from a master station, belonging or not to a network of permanent GNSS stations, through the Internet via NTRIP protocol (Figure 2). At the moment goGPS is designed to obtain GPS raw data from u-blox LEA modules ("T" version) by decoding their proprietary binary stream (UBX format) and from the master GNSS station in RTCM 3.1 format. goGPS can run either in real-time or post-processing mode, synchronizing the rover and master data streams and managing temporary outages or permanent data losses. Various events such as satellite additions/losses, change of pivot satellite (highest satellite used for double differences) and cycle-slips are managed. Real-time data can be displayed either on a MATLAB figure or on Google Earth and at the end of each session goGPS produces files in NMEA 0183 and KML formats (besides its own binary and text data files).

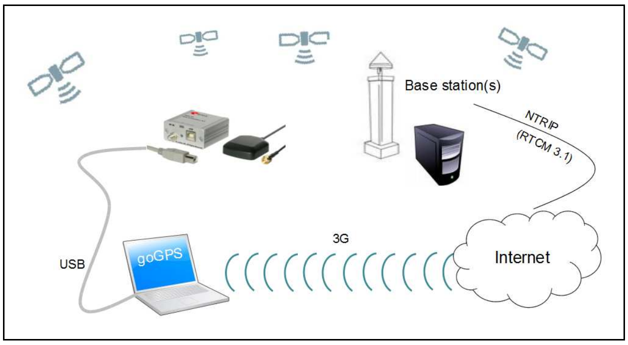

Figure 2. goGPS real-time system

3.1.1 Customized NMEA sentence for Kalman filter DOP: The NMEA 0183 specifications include the possibility to define customized (usually vendor-specific) sentences to provide additional information not encompassed by standard sentences. A customized sentence was defined to output KPDOP, KHDOP and KVDOP values in NMEA format.

3.1.2 Real-time data stream through socket communication: Socket programs were developed in order to carry out initial tests of remote processing between a GPS rover and server running goGPS. A socket is a kind of network address, a combination of IP address and a port number in a network computer. Socket communications work between a socket server and a socket client programs. The socket server opens a certain port and receives the data with TCP or UDP connection. The client designates the port and IP address of the server and sends data to the socket server.
Socket server and client programs were developed using Java. One of the reasons for adopting Java is that the programs can run under any OS and devices that support the Java Virtual Machine (JVM) (e.g. Linux, Apple Macintosh and also embedded devices).
The Java client program enables a GPS rover to send raw data (in this example: u-blox binary stream) to the server running goGPS. The Java server program receives the stream and redirects it to a local UDP port, which is then opened by MATLAB standard UDP functions (Figure 3).

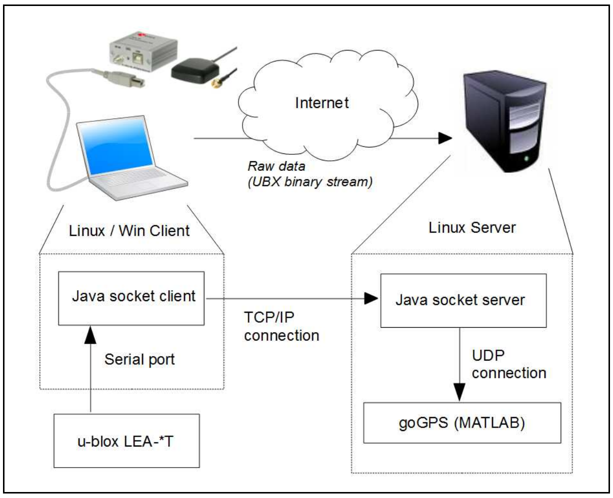

Figure 3. Remote processing by socket communication

3.2 Web-based track log management system architecture

The architecture of the system can be divided into three components (Figure 4). The first being goGPS component which processes GPS raw data and enhances the accuracy as explained in previous sections. The second is the server component that provides data archiving and geospatial services, this will be explained in following section. The third is the client component for viewing GPS locations and track logs. In particular the server component enables searches within track logs and tracking services to the client components through Hyper Text Transfer Protocol (HTTP). 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. Desktop GIS or other GIS applications can be used for displaying search results which are exported from the OpenLayers interface.

3.2.1 Server component: After goGPS host-based processing (client-side), the resulting position data can be uploaded to the server database (PostgreSQL) through a web interface, in NMEA format. Once stored in its database, the data can be queried by some parameters through the OpenLayers web interface. Then PostgreSQL and PostGIS query and retrieve the corresponding data.
The server component was implemented on a Linux platform using Apache, PHP, PostgreSQL and PostGIS in order to store and query GPS data and OpenLayers was adopted for the web mapping viewer application for the client interface.

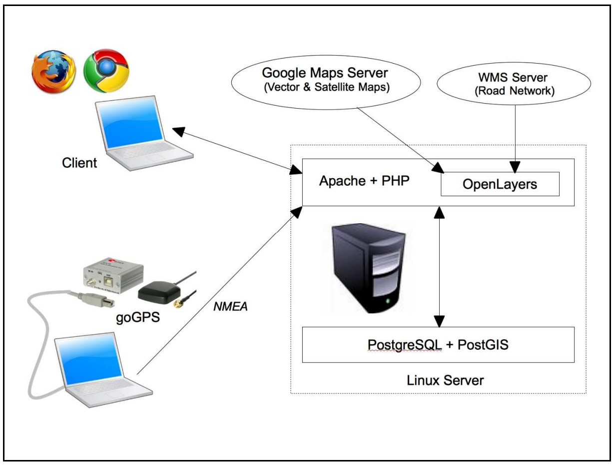

Figure 4. System overview of the track-log management system

3.2.2 Line generalization function: Some queries extract a large number of data and take some time to display the result due to data transmission between the server and client. In order to reduce the amount of data transmission, a line simplification function was implemented using the Douglas-Peucker algorithm. Song et al., 2010 also adopted this algorithm to produce simplified lines using accumulated GPS track logs. The application of Douglas–Peucker algorithm for line generalization is more effective and efficient in eliminating those error points. In addition, the algorithm also eliminates extraneous position points and can speed up the performance at the next processing stage.
The function simplifies points based on a given tolerance parameter, and it is also effective for the points in which GPS receivers record the same positions. This function not only archives efficient data management, but also reduces data noise. Figure 5 shows the simplification result. The left figure is before the simplification process and the right figure is after the simplification, emphasizing how the number of points is successfully reduced.

Figure 5. Line simplification result

4. TESTS

Some tests were performed in order to check the performance of the KHDOP index compared to the standard HDOP and to verify the feasibility and performance of raw data remote processing by socket network communication.

4.1 Quality-based filtering

Various tests were performed in low-density and high-density urban environments, in order to check if data could be more effectively filtered out by standard HDOP or by using KHDOP. The example presented here, surveyed by car in a low-density urban environment in Italy, was chosen because part of it was recorded on the first floor of a two-story parking structure, providing a sudden change from good sky visibility to no sky visibility at all. Since the GPS device used was a u-blox AEK-4T, its high sensitivity allowed for signal reception and positioning even with the highly degraded signal inside the parking structure.

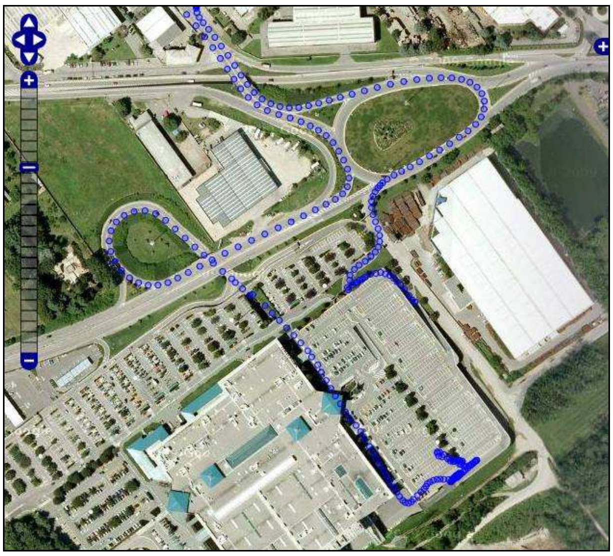

Figure 6. Complete track (on Google satellite image)

Figure 6 shows the complete track without filtering (the parking structure is in the bottom right-hand part of the image), while Figure 7 shows the comparison between HDOP-filtered and KHDOP-filtered tracks. The positioning inside the parking structure was very bad due to the lack of sky visibility, but the HDOP-filtered track keeps some of its points even when the threshold is very low. By using such a strict threshold on HDOP, based on satellite geometry only, part of the outdoor track is deleted although it is much more accurate than the indoor track.

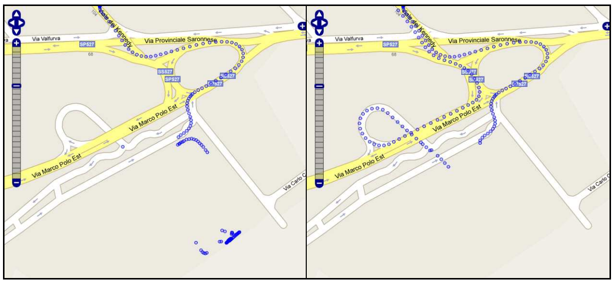

Figure 7. HDOP-filtered track (left) and KHDOP-filtered track (right) (on Google Maps)

On the other hand, the track filtered by KHDOP effectively removes only the points surveyed within the parking structure, leaving unchanged the rest of the track. Other tests confirm the better performance of the KHDOP over the HDOP as an indicator of the positioning quality.

4.2 Remote processing

goGPS remote processing was tested at a basic level for the purpose of this work: a u-blox AEK-4T was connected to a client laptop, providing data at 1 Hz; a Java socket client program was installed on the laptop; goGPS was installed on a Linux server, together with a Java socket server program. Once the connection was established between the socket client and server, u-blox binary data started flowing through the network and they reached goGPS. The integrity and speed of the received data were verified by making goGPS decode and display them in real time. This procedure was repeated both over a LAN network and over a mobile Internet connection. No data loss was experienced, but the data transmission was delayed slightly more when using the mobile connection, as it was expected.

5. CONCLUSIONS

goGPS software and a web-based track log management system were made interoperable by homogenizing output and input data using the standard *de facto* NMEA 0183. This procedure lead to various improvements: goGPS output was standardized to NMEA and the ability to process raw data coming from remote GPS devices was added; Kalman filter based DOP values (KPDOP, KHDOP and KVDOP) were developed and used as effective alternatives to traditional DOP values for discriminating between poorly positioned points and higher accuracy points; in particular the KHDOP index was implemented into the web-based system.

Future developments include the measurement of delays induced by the remote network transmission, thorough testing of remote processing and porting goGPS from MATLAB to Java.

6. ACKNOWLEDGEMENTS

The authors acknowledge the positioning services IREALP GPSLombardia (Italy) and JENOBA (Japan) for their support. The first author acknowledges the Japan Society for the Promotion of Science (JSPS) for the post-doctoral fellowship under which this work was carried out. This research is supported by JSPS Grant-in-Aid for Scientific Research (Issue No. 2109737) entitled "Development of Ubiquitous LBS Web-Service using Free and Open Source Software".

7. REFERENCES

Alanen, K., Wirola, L., Käppi, J., Syrjärinne, J., 2006. Inertial Sensor Enhanced Mobile RTK Solution Using Low-Cost Assisted GPS Receivers and Internet-Enabled Cellular Phones. *Inside GNSS*, 1(4), May/June 2006, pp. 32-39.

Grewal, M.S., Andrews, A.P., 2001. *Kalman Filtering: Theory and Practice Using Matlab*. John Wiley & sons, Ltd.

Hofmann-Wellenhof, B., Lichtenegger, H., Wasle, E., 2007. *GNSS Global Navigation Satellite Systems: GPS, GLONASS, Galileo, and more*. SpringerWienNewYork, Wien, pp. 262-266.

Kalman, R. E., 1960. A New Approach to Linear Filtering and Prediction Problems. *Transaction of the ASME - Journal of Basic Engineering*, pp. 35-45.

Leclerc, Y. G., Reddy, M., Iverson, L., Eriksen, M., 2001. The GeoWeb - A New Paradigm for Finding Data on the Web. In: *Proceedings of the International Cartographic Conference*.

Söderholm, S., Jokitalo, T., Kaisti, K., Kuusniemi, H., Naukkarinen H., 2008. Smart Positioning with Fastrax's Software GPS Receiver Solution. Presented at ION GNSS 2008, September 16-19, Savannah, Georgia, USA.
http://www.fastraxgps.com/products/softwaregps/ (acc. 24 May 2010)

Song, X., Raghavan, V., Yoshida, D., 2010. Matching of vehicle GPS traces with urban road networks. *Current Science*, 98(12).

Spilker, J.J., 1996. Satellite Constellation and Geometric Dilution of Precision. *Global Positioning System: Theory and Applications*, 1, American Institute of Aeronautics and Astronautics (AIAA), Progress in Astronautics and Aeronautics, 163, pp. 177-208.

Van Diggelen, F., 2009. The Smartphone Revolution. *GPS World*, December 2009.
http://www.gpsworld.com/wireless/smartphone-revolution-9183 (acc. 24 May 2010)

Van Diggelen, F., 2010. Expert Advice: Are We There Yet? The State of the Consumer Industry. *GPS World*, March 2010.
http://www.gpsworld.com/consumer-oem/expert-advice-are-we-there-yet-the-state-consumer-industry-9577 (acc. 24 May 2010)

Wirola, L., 2008. High-accuracy Positioning for the Mass Market. Presented at FIG Working Week 2008, June 14-19, Stockholm, Sweden.
http://www.fig.net/pub/fig2008/ppt/ts08f/ts08f_03_wirola_ppt_3167.pdf (acc. 24 May 2010)