

goGPS: ACCURATE ROAD MAPPING USING LOW-COST GPS RECEIVERS

Lisa Pertusini¹, Eugenio Realini², Mirko Reguzzoni¹

¹ DIAR, Politecnico di Milano, Como Campus, via Valleggio 11, 22100 Como, Italy
Email: lisa.pertusini@mail.polimi.it; mirko@geomatica.com.polimi.it

² Graduate School for Creative Cities, Osaka City University
3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-8585, Japan
Email: realini@gsc.osaka-cu.ac.jp

ABSTRACT

Road mapping is traditionally performed by using high-level and high-cost instrumentation such as photogrammetric digital cameras, double frequency GPS receivers, inertial measurement units, etc. However, latest developments in GPS data analysis allow obtaining good results also using low-cost devices supported by software like for example goGPS. This software allows enhancing GPS positioning with low-cost receivers mainly by exploiting the principle of relative positioning with respect to a master station located in the area where the survey is performed. Typically goGPS output is a sequence of estimated points at 1 second sampling rate leading to a huge amount of data and, consequently, arcs when these data are used for road mapping. In this paper we implement an algorithm to reduce the number of arcs by first selecting the nodes using an agglomerative clustering procedure and then fitting the resulting polyline on the original GPS dataset by least-squares adjustment. In this way we aim at automatizing the production of road networks, simplifying the procedure currently used for example in OpenStreetMap. This could be particularly useful for developing countries, where the availability of high cost professional instrumentation is often limited. The proposed method is applied in a test scenario evaluating its performances.

1. INTRODUCTION

A problem of great interest in the field of GPS navigation, especially in developing countries, is how to estimate a road network with low-cost instrumentation. Despite the increasing popularity of route guidance systems, current digital maps are still inadequate in developing countries for many advanced applications. Among others, the main drawbacks are the insufficient accuracy of road geometry and the lack of fine-grained information, like for example lane positions.

In this paper, an approach to derive high-precision road maps by using low-cost GPS receivers in differential mode is presented. The procedure is based on the goGPS software, which is introduced in Section 2, and consists of the successive steps described in Section 3. An example on how the procedure works with real GPS data is provided in Section 4, also discussing future developments.

2. goGPS OPEN SOURCE SOFTWARE FOR GPS NAVIGATION

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 with bad sky

visibility conditions, like in dense urban environments or under forest canopy. Nevertheless, the positioning accuracy with this kind of devices is generally low, not only because of the classical errors induced by atmospheric and clock delays, but also because of the low quality of involved hardware and the multipath resulting from high sensitivity.

goGPS (<http://www.gogps-project.org>) is an open source software package designed to perform relative positioning with code and phase (Real Time Kinematic, RTK) using low-cost GPS instrumentation, if this gives raw data (GPS observations) in output: in this way atmospheric and clock errors are removed or made negligible. Moreover, goGPS includes a Kalman filter applied to GPS observations which reduces the estimation error by modelling the receiver dynamics, by weighing observations on the basis of their quality (i.e. signal-to-noise ratio and satellite elevation), by integrating additional information like digital terrain models or road networks and by exploiting at best the information provided by the L1 phase, managing its ambiguities when changes of satellite configuration or cycle slips happen. goGPS can work either in RTK or stand-alone mode and it can be used either for real-time or post-processing tasks. Observations coming from a reference GPS station are needed for the RTK mode, thus a mobile Internet connection has to be available when using goGPS in real-time. On the other hand, for post-processing tasks, reference station data in RINEX format can be used.

3. ARC-SEGMENTATION ALGORITHM

The simplest and probably most common solution to the problem of road mapping by using low-cost GPS receivers is to report directly the computed trajectory on the map of interest. However it should be noticed that the path returned by GPS receivers (e.g. NMEA) is nothing but a sequence of points which can be extremely lengthy if the sampling interval of the GPS observations is equal to 1 second. For example, there is no simplification in the case of a straight line that could be equivalently described by a single arc instead of many points. Hence a simple network topology (polyline), based on a limited number of arcs and nodes, should be preferred for the trajectory modelling. To this aim, many solutions have been studied (see e.g. Douglas and Peucker, 1973); here we propose an algorithm based on the following steps:

1. automatic selection of significant nodes chosen from several GPS points, i.e. among the positions estimated by goGPS at every second;
2. assignment of each GPS point to a certain arc (which is a segment joining two nodes);
3. least-squares adjustment of the node positions, in such a way that the resulting polyline correctly interpolates the GPS points.

The nodes estimated at step 3 are provided as input to the step 2 and the procedure is iterated to obtain the final polyline. Let us now describe the individual steps in more detail.

3.1 Node selection by AGNES

The identification of significant nodes and consequently the determination of their number is performed using the agglomerative classification algorithm AGNES (Kaufman and Rousseeuw, 1990).

At first all the GPS points are considered as nodes. Then the angle between each pair of consecutive arcs is computed; to be more precise, calling A , O and B three generic

consecutive nodes of coordinates $\underline{x}_A = (x_A, y_A)$, $\underline{x}_0 = (x_0, y_0)$ e $\underline{x}_B = (x_B, y_B)$, the angle is calculated as:

$$A\hat{O}B = \arccos \left[\frac{(\underline{x}_A - \underline{x}_0) \cdot (\underline{x}_B - \underline{x}_0)}{|\underline{x}_A - \underline{x}_0| |\underline{x}_B - \underline{x}_0|} \right] \quad 0 \leq A\hat{O}B \leq 180^\circ \quad (1)$$

Among all the computed angles, the largest is detected, i.e. the closest to 180 degrees. In other words, the two consecutive arcs with the smallest change in direction are identified. If this maximum angle is larger than a certain threshold (e.g. 170 degrees), the node O , which is common to both arcs \overline{AO} and \overline{OB} , is eliminated, resulting in a single arc \overline{AB} . After recomputing the angles with respect to this new arc, the algorithm is repeated iteratively until the maximum angle is below the threshold (see Figure 1).

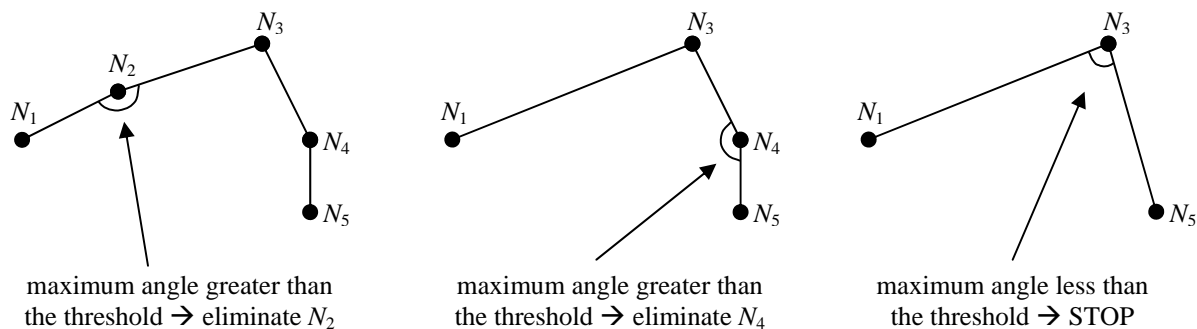


Figure 1. Scheme of how the agglomerative algorithm works for the selection of nodes.

The procedure is applied to the goGPS solution based on code and phase double differences, typically overweighing the dynamic model of the Kalman filter. In this way the sequence of GPS points draws a smooth path, avoiding the agglomerative algorithm to keep too many nodes.

Despite this expedient, when the receiver is kept stationary during the measurement, the Kalman filter returns a cloud of points around the true position with sudden changes of direction between two consecutive arcs (Broveli et al. 2008). This causes the selection of a large number of nodes in a very restricted area. To avoid this, a further selection is implemented downstream of the agglomerative method, verifying that the distance between two consecutive nodes is greater than a certain threshold. If this condition is not satisfied, only the first of the two nodes is selected. The test is repeated iteratively until the condition is true for all pairs of adjacent nodes (see Figure 2).

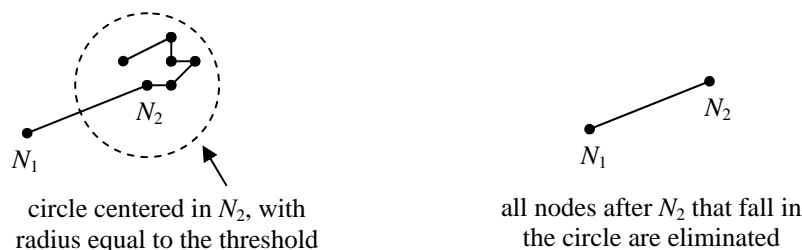


Figure 2. Scheme of the elimination of consecutive nodes too close each other.

3.2 Data classification using bounding boxes

Once the nodes have been selected, the arcs are implicitly defined by connecting consecutive nodes. However, the fact that the selected nodes are a subset of the original GPS

points implies that the corresponding arcs do not necessarily optimally interpolate the other GPS points (see Figure 3, on the left). Moreover, as mentioned above, the selection procedure of the nodes is done on a smoothed trajectory, while the final polyline should be calculated on a trajectory as faithful as possible to reality.

For these reasons, the GPS points are re-estimated using goGPS, but this time by properly weighing the dynamic model of the Kalman filter with respect to the measurement errors of code and phase observations. The obtained GPS points have to be assigned to the different arcs, in order to subsequently optimize the interpolating polyline. In other words each arc is named with a certain label (a number from 1 to n , where n is the number of segments) and one and only one label is assigned to each GPS point (including the label 0 if the GPS point is not assigned to any arc).

In order to perform this classification, a rectangular buffer (i.e. a bounding box) of size Δ is built around the line connecting two consecutive nodes (see Figure 3, on the right). The following classification rules are implemented:

- if a GPS point does not fall into any buffer, it is labelled with 0 and it is excluded from the subsequent process of least-squares interpolation;
- if a GPS point falls into one and only one buffer, it is marked by the corresponding arc label;
- if a GPS point falls into more than one buffer (an event that frequently occurs near the nodes if the value of Δ is too high), the sum of the distances between the GPS point and the two nodes is computed for each arc involved; the label of the arc for which this sum is minimum is attributed to the GPS point.

Finally it may occur that no GPS points or only a single GPS point fall in the buffer of a certain arc. This does not allow to re-estimate the arc by the subsequent least-squares interpolation and then the arc itself is eliminated by deleting one of its two nodes.

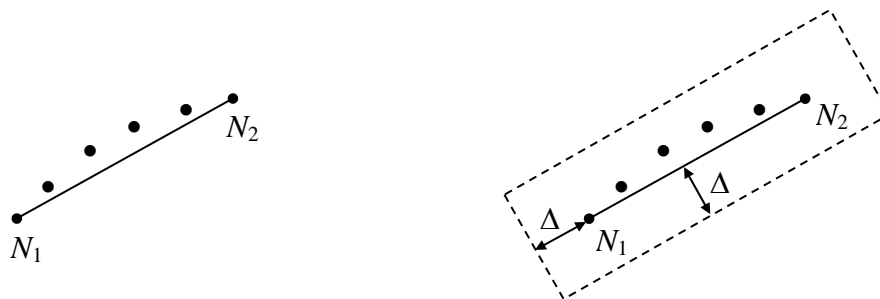


Figure 3. Example of arc non-optimally interpolating the GPS points (on the left); definition of the corresponding bounding box of size Δ (on the right).

3.3 Least-squares adjustment of the nodes

The idea is to interpolate the GPS points falling into the same bounding box with a line, namely with the following model:

$$y - y_0 = m(x - x_0) \quad (2)$$

where x_0 is an arbitrary value set a priori (for example, the midpoint of the x coordinates of the GPS points), while the corresponding value y_0 and the angular coefficient m of the line are the two unknowns of the problem. Note that the GPS points are “observed” with an error in both coordinates, so we can write:

$$x_{0_i} = x_i + v_{x_i}, \quad y_{0_i} = y_i + v_{y_i}, \quad \text{with } i = 1, 2, \dots, N \quad (3)$$

where the variances of the errors $\sigma_{x_i}^2$ and $\sigma_{y_i}^2$ are provided by the Kalman filter implemented in goGPS, while the covariance is neglected for simplicity. Assuming that the angular coefficient m can be written as $m = \tilde{m} + \delta m$ and that the approximation $m v_{x_i} \cong \tilde{m} v_{x_i}$ holds, the observation equation of the least-squares problem results:

$$y_{0_i} - v_{y_i} - y_0 - m(x_{0_i} - x_0) + \tilde{m} v_{x_i} = 0 \quad (4)$$

where the approximate value of the angular coefficients is generally estimated as:

$$\tilde{m} = \frac{1}{N} \sum_{i=1}^N \frac{y_{0_i} - \bar{y}}{x_{0_i} - \bar{x}}, \quad \text{where } \bar{x} = \frac{1}{N} \sum_{i=1}^N x_{0_i}, \quad \bar{y} = \frac{1}{N} \sum_{i=1}^N y_{0_i}. \quad (5)$$

The target function of the least-squares problem can be written as:

$$\phi = \frac{1}{2} \sum_{i=1}^N \left(\frac{v_{y_i}^2}{\sigma_{y_i}^2} + \frac{v_{x_i}^2}{\sigma_{x_i}^2} \right) \quad (6)$$

which has to be minimized with respect to the condition (4) using Lagrange multipliers. The resulting system by which y_0 and m are estimated can be written in matrix notation as:

$$\begin{bmatrix} \sum_i q_i & \sum_i q_i (x_{0_i} - x_0) \\ \sum_i q_i (x_{0_i} - x_0) & \sum_i q_i (x_{0_i} - x_0)^2 \end{bmatrix} \begin{bmatrix} y_0 \\ m \end{bmatrix} = \begin{bmatrix} \sum_i q_i y_{0_i} \\ \sum_i q_i y_{0_i} (x_{0_i} - x_0) \end{bmatrix} \quad (7)$$

where $q_i = (\sigma_{y_i}^2 + \tilde{m}^2 \sigma_{x_i}^2)^{-1}$.

Since the values of y_0 and m are independently estimated arc by arc, it is necessary to recompute the intersection of two consecutive lines, thus determining the new location of the intermediate node, which varies not only with respect to the y coordinate but also with respect to the x coordinate (Figure 4, on the left). However if the new intersection is too far from the previous node, i.e. if it is beyond a certain distance threshold defined by the user, the x coordinate of the node is not changed, while the y coordinate is computed as the average of the two values of the two lines at the x coordinate (Figure 4, on the right). The resulting sequence of nodes and arcs defines the network topology (polyline).

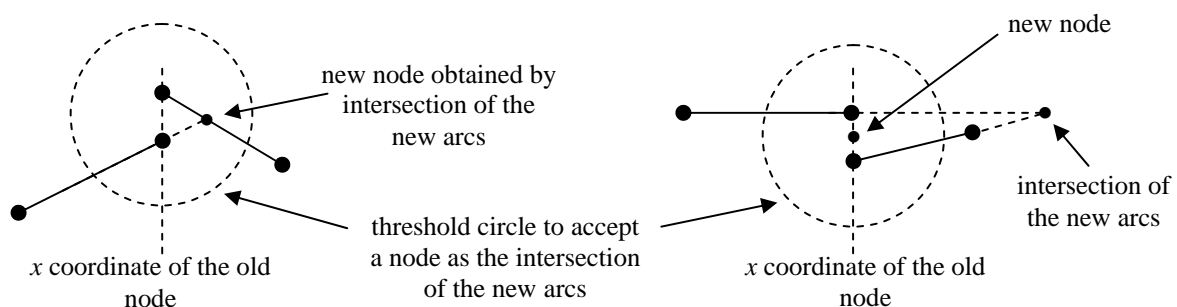


Figure 4. Determination of new nodes after least-squares interpolation of the arcs.

4. TESTS AND FUTURE DEVELOPMENTS

Various tests were performed to check the performance of the proposed segmentation algorithm. In this paper we are going to illustrate only one of them, representative of the current state-of-the-art of the algorithm. The test dataset was surveyed by mounting a low-cost GPS device (u-blox AEK-4T) on the rooftop of a car, driven for about 50 minutes

through the streets of Izumi and Sakai cities, part of the Osaka metropolitan area. The path covered approximately 18 km, with a total number of 3047 surveyed points (at 1 Hz measurement rate). The average speed on straight roads was about 40 km/h, while it was about 20 km/h on curves. The parameters of the segmentation algorithm were manually adjusted to obtain a good balance between data reduction and detail level, especially for representing curves. The node selection was performed by setting AGNES angular threshold to 178 degrees and the minimum distance between consecutive nodes to 12 m. The bounding box size was set to 20 m for the first iteration and 15 m for the second iteration (smaller sizes led to the deletion of too many arcs). The distance threshold to accept or reject new nodes was 2 m for both iterations. These values allowed a very good simplification of the trajectory on straight paths, nevertheless preserving a good level of smoothness for curves (Figure 5).



Figure 5. Arc-segmentation example on a path with smooth curves (path overlaid on Google Earth).

The parameterization obviously depends on the type of survey, therefore adjusting parameters manually could be very cumbersome for users. To address this issue, future developments include the automatic tuning of parameters according to the path features.

Another limitation of the current implementation of the arc-segmentation algorithm is that it can work only on non-intersecting paths. This issue is going to be addressed by identifying sub-paths composed by non-intersecting segments, which could then be processed independently. The computation of a full network topology is planned for the future. Further developments include also the comparison of the algorithm proposed in this paper with the widely used Douglas-Peucker algorithm for line simplification (Douglas and Peucker, 1973) and the testing of the resulting polyline with OpenStreetMap map drawing tools.

5. ACKNOWLEDGEMENTS

We acknowledge for his collaboration the student Befkadu Nigussie Alemu, who based his master thesis on this research.

6. REFERENCES

- Brovelli, M.A., Realini, E., Reguzzoni, M., Visconti, M.G., 2008. Comparison of the performance of medium and low level GNSS apparatus, with and without reference networks. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XXXVI, part 5/C55, 54-61.
- Douglas, D.H., Peucker, T.K., 1973. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2), 112-122.
- Kaufman, L., Rousseeuw, P.J., 1990. *Finding groups in data. An introduction to cluster analysis*. John Wiley & Sons, New York.