ARTICLE IN PRESS

Journal of Network and Computer Applications ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at SciVerse ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Real-time GPS track simplification algorithm for outdoor navigation of visually impaired

Rosen Ivanov *

Department of Computer Systems and Technologies, Technical University, Gabrovo, Bulgaria

ARTICLE INFO

Article history:
Received 2 December 2010
Received in revised form
23 December 2011
Accepted 8 February 2012

Keywords:
Real-time GPS track simplification
Navigation of visually impaired
Navigation along a track
Java applications for navigation

ABSTRACT

Outdoor navigation of visually impaired people most often is based on data from the GPS maps and Geographic Information Systems (GIS). Such systems cannot be used for navigation in the regions where there are not any GPS maps or the maps are not sufficiently precise and detailed. This article describes an algorithm for real-time GPS track simplification. The test results show that the proposed algorithm (RSTS) has efficiency similar to the Douglas–Peucker algorithm, which is regarded as the best for track simplification. The reduction of the number of points in different transportation modes, while keeping the shape of the route, is over 90%. The size of the file describing the track is reduced more than 30 times. The algorithm finds the critical points of the route. This allows for navigation along the track so conversion of track to route is not needed. The algorithm is part of a low cost and widely accessible Java 2 Mobile Edition (J2ME) application for navigation of visually impaired.

1. Introduction

Most of the existing GPS navigation systems for people with visual disabilities and blindness use digital commercial GPS maps from a third party or GIS. These include Wayfinder Access (Wayfinder Systems AB), Trekker and BrailleNote GPS (Human Ware), MobileGeo (Code Factory) and Drishti (University of Florida).

Such systems are unsuitable for regions where there are not any GPS maps or the maps are not sufficiently accurate. This is typical for most small towns and regions outside the cities. The main weaknesses of the existing GPS navigation systems for the blind are as follows: (1) The systems are priced too high and do not match the purchasing power of people with visual disabilities. Most of the navigation systems require specialized hardware and/or mobile phones such as smartphones or PDAs, screen readers such as Talks and Mobile Speak, access to GIS, and last but not least GPS maps that need to be updated periodically. (2) Navigation along a track, if maintained, in most cases is not adapted for use by people with visual disabilities.

When a GPS map is missing or it is not sufficiently accurate or detailed, the only alternative is to navigate along a track. Most of the GPS navigators and mobile GPS navigation systems allow for a trace of routes. The problem is that very few of them allow navigation along a track. The reason is the difficulty in determining the points at which there is a real change of direction (critical points), mainly due to errors in GPS data. Erroneous identification of critical points is fatal for navigation of visually impaired.

Another major problem in obtaining a trace of route is a large amount of memory needed to record the track, especially when walking. This imposes a restriction on the number of saved tracks and the maximum number of points by which they are described. For example, Garmin GPS navigators for pedestrians, which have not supported GPS maps, can save 10 or 20 tracks, and each track can contain up to 10,000 points. The solution for this problem is to use algorithms for real-time track simplification.

Since each track is described by a sequence of points, it can be assumed that track simplification algorithms are part of the line simplification algorithms. McMaster (1987) classifies them into five categories: (1) Independent point algorithms, (2) local processing algorithms, (3) Unconstrained extended local processing algorithms, (4) constrained extended local processing algorithms and (5) global processing algorithms. The first four types of algorithms belong to Local line simplification algorithms, and the last—to the Global line simplification algorithms (Shi and Cheung, 2006).

Local simplification algorithms: The relationship between every two or three consecutive points is analyzed. These types of algorithms are very simple and fast, but because of the local data processing it is very difficult to obtain optimal results. More commonly used local track simplification algorithms are as follows: *n*th point – only one point between *n* consecutive points is retained; distance threshold based algorithms – all points, for which the distance to the preceding track point is less than the

* Tel.: +359 887285445.
E-mail address: rs-soft@ieee.org

1084-8045/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2012.02.002

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002

predetermined threshold is deleted; changes in direction based algorithms – the point is retained if the change in direction is greater than the predetermined threshold and perpendicular distance based algorithms – the point is retained if distance from point to the segment formed by the previous and next point is greater than the predetermined threshold. To the local algorithms belong and algorithms that analyze the section of consecutive points, for example: the Reumann–Witkam algorithm, the Opheim algorithm and the Lang algorithm. The aim is to obtain better performance, based on an analysis of a larger number of track points.

Global simplification algorithms: In this category of algorithms, the decision about which point to delete and which to retain is based on the relationship of all points that belong to the track. Global algorithms extract critical points more accurately than local algorithms, but require more memory and are much slower. This makes them unusable when real-time track simplification algorithms are needed, which are implemented on platforms with limited resources. At this moment, mainly in cartography and GIS, two global algorithms and their modifications are used: the Douglas–Peucker (DP) algorithm (Douglas and Peucker, 1973; Hershberger and Snoeyink, 1992) and the Visvalingam–Whyatt algorithm (Visvalingam and Whyatt, 1993). The experiments show (Cheung and Shi, 2006) that the DP algorithm produces the most accurate track simplification with minimum shape distortions. A major drawback of the DP algorithm is the computation time. The average complexity of the algorithm is $O(n^2 \log m)$, and in the best case it is $O(n \log n \log m)$, where n is the total number of points and m is the number of points retained.

Most mobile applications that enable track simplification and recording (see Table 1) and algorithms for real-time track simplification (Barbeau et al., 2008; Chen et al., 2009; Zhung et al., 2008) use local algorithms, based mainly on the heading change and neighbor points distance. These algorithms can find practical use only if there is no noise in the GPS data and therefore they cannot be used for navigation of visually impaired people.

The track simplification algorithm, which works in real-time and its accuracy is comparable with the DP accuracy, is presented. The algorithm uses a combination of several local simplification techniques: changes in direction, perpendicular distance and distance threshold. To reduce the sensitivity of the algorithm to errors in GPS data, an algorithm for adaptive filtering of GPS positions is proposed. It combines: knowledge-based GPS data pre-processing, a fuzzy estimation of noise in GPS data and Adaptive Kalman Filtering. Since algoritamat is adaptive to the velocity, it can be used not only for developing applications for navigation of visually impaired but also in GPS navigators, location based services (LBS) applications, location based social networking (LBSN) applications and GIS-based services.

2. GPS track simplification algorithm architecture

Fig. 1 shows a sequence of implementation of the proposed algorithm in the form of a process flow diagram.

The following describes the stages of implementation of the algorithm.

2.1. GPS data logging

The stage is implemented by the NMEA 0183 parser, which analyzes data from the GPS receiver (NMEA sentences GPGGA, GPGSA, GPGSV and GPRMC) and extracts the necessary for the

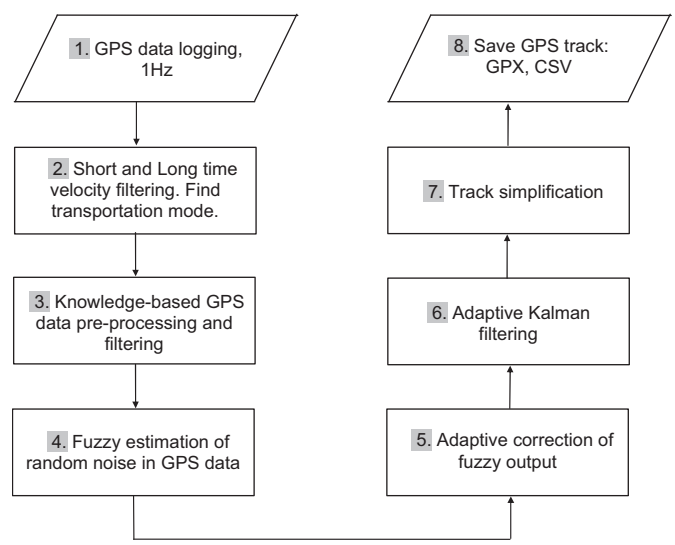


Fig. 1. Track simplification algorithm process flow diagram.

Table 1
Comparison of real-time GPS track simplification algorithms.

Application (platform)	Type of algorithm	Adaptive to the velocity	"Wild"[a] points remove	GPS positions filtering	Adaptive to noise in GPS data
PDA.bgmaps (Microsoft Pocket PC)	Local: distance threshold	No	No	No	No
gpsVP (Windows Mobile)	Local: distance threshold	No	No	No	No
Odgps 1.2 (Pocket PC, J2ME)	Local: *n*th point, changes in direction	No	No	No	No
AFTrack 1.20 (S60)	Local: *n*th point, distance threshold, changes in direction	No	No	No	No
GPS Mid (J2ME)	Local: *n*th point, distance threshold	Yes	No	No	No
Nokia Sport Tracker (S60)	Local: *n*th point, distance threshold	No	No	Yes, filter out erroneous GPS locations	No
Mobile Trail Explorer (J2ME)	Local: *n*th point, distance threshold	No	No	No	No
TRACK-IT(J2ME), Barbeau et al. (2008)	Local: changes in direction	Yes	No	No	No
Chen et al. (2009)	Local: distance threshold, changes in direction	Yes, adaptive to transportation mode	No	No	No
Proposed algorithm **RSTS** (J2ME)	Local: distance threshold, changes in direction, perpendicular distance	Yes	Yes, knowledge-based	Yes, adaptive Kalman filter	Yes, fuzzy logic

[a] "Wild" points are typically caused by poor GPS reception.

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002

proposed algorithm GPS data: number of visible satellites; satellite position – azimuth and elevation; SNR for each satellite; Horizontal Delution of Precision (HDOP) – estimation of horizontal accuracy of GPS receiver; WSG 84 user position (longitude and latitude) and velocity.

2.2. Short and long time velocity filtering: Find transportation mode

Velocity filtering is used to determine transportation mode—"walking" or "not walking" and in other stages of the proposed algorithm. To determine the transportation mode, a first order Inverse Impulse Response (IIR) filter is used—Short Time Velocity Filtering (STVF):

$$\begin{aligned} &\mathrm{STVF}_i = \alpha \mathrm{STVF}_{i-1} + (1-\alpha) v_i \quad \text{if} \quad v_i > 0, \\ &\mathrm{STVF}_i = \mathrm{STVF}_{i-1} \quad \text{if} \quad v_i = 0, \end{aligned} \tag{1}$$

where α is a parameter that controls degree of filtering, and v_i is the velocity in discrete time i. It is assumed that transportation mode is "walking" if STVF < 7 km/h.

Long time velocity filtering (LTVF) is calculated as the average velocity for the interval of last M discrete (s):

$$\mathrm{LTVF}_i = \frac{1}{M}\left(\sum_{j=i-M+1}^{i} v_j\right). \tag{2}$$

2.3. Knowledge-based GPS data pre-processing and filtering

The task of this stage is to filter errors ("wild" points) in GPS data that cannot be removed at the stage "Adaptive Kalman Filtering." This is realized on the basis of knowledge and expertise for raw GPS data in the form of predefined rules.

Experiments show that the largest errors in GPS data were when the following events occurred: transition from invalid to valid GPS data, changes in the number of visible satellites, transitions "movement-stop-movement," walking with very low speed (less than 3 km/h), which is typical for the blind, and when static GPS navigation is active.

Rule 1. Number of visible satellites and accuracy of the GPS receiver.

For estimation of the accuracy of GPS positions, HDOP parameter is used. As its value is less, the accuracy is better. This value depends on the relation between positions of visible satellites and GPS receiver. The HDOP parameter does not take into account signal quality of the individual satellites. In practice, it appears that even at high HDOP values, the position may be sufficiently accurate if the number of visible satellites (N) is large enough. Rule 1 defines which GPS data is to be used for further processing:

$$N > 5 \quad \text{OR} \quad (N > 3 \quad \text{AND} \quad \mathrm{HDOP} < 3.25)$$

Rule 2. Changes in the validity of GPS data.

Analysis of raw GPS data shows that the error in position is the greatest in the transition from invalid (no GPS fix) to valid (3D GPS fix) data for $N > 3$. Deviation from the actual position can be over 50 m. After a certain time this error is minimized. Rule 2 minimizes this type of error as follows:

Skip the next 5 GPS positions AND filter maximally the next 5 GPS positions.

Rule 3. Changes in the number of visible satellites.

A change in the number of visible satellites leads to calculating the position based on data involving different combinations of satellites. Therefore, the accuracy is improved or gets worse. As a result, deviation from the previous position is within 10–30 m. This can lead to insertion of redundant points. Rule 3 must remove most of these points:

Skip GPS positions if $|N_i - N_{i-1}| > 2$ AND filter maximally the next 3 GPS positions.

Rule 4. Status of static navigation.

Most of the GPS receivers are not designed especially for the pedestrians and static navigation is enabled by default. In this mode the GPS data is filtered by the GPS receiver firmware and when the velocity is lower than a predefined threshold the GPS receiver starts to generate the last calculated position. When the user is walking at a low speed and frequently stops, for some sections of the road (10–40 m) information about the position is missed. The proposed algorithm for GPS positions filtering relies on the fact that the GPS receiver generates data in the same interval of time (1 s). This can lead to large errors in regions, containing a smooth curve. Rule 4 is responsible for eliminating these errors:

Disable static navigation OR filter minimally the next 3 GPS positions.

Rule 5. "Movement-stop-movement" transition regions.

For these regions, errors in GPS position are big, regardless of the transportation mode. The greatest error is 10–40 m, if the transportation mode is "not-walking" and static navigation is active. Fig. 2 shows the error in position in two regions of the track, containing transitions of type "movement-stop-movement."

Rule 5 aims to remove GPS positions in transitional regions:

Skip all points between the start and stop point AND filter maximally the next 3 GPS positions.

For this purpose, start and stop points of these regions is looking for.

Get a start point: For the interval of M discrete, the number of points (*counter*) is calculated for which the following condition is fulfilled:

$$\mathrm{LTVF}_{i-j} < \mathrm{LTVF}_{i-j-1}, \quad j = [0,\ M-1]. \tag{3}$$

If *counter*$=M$, it is assumed that point i is a start point of the region.

Get a stop point: For the stop point to be considered it must meet the following condition:

$$\mathrm{STVF}_i > Th_1 \quad \text{and} \quad (\mathrm{LTVF}_i < Th_2/3 \ \text{ or } \ v_i = 0), \tag{4}$$

where the thresholds are as follows:

$$\mathrm{Th}_1 = 15\,\mathrm{km/h},\ \mathrm{Th}_2 = \min(\mathrm{Th}_1, 0.7\mathrm{Th}_3),\ \mathrm{Th}_3 = \frac{1}{M/2}\sum_{j=0}^{(M/2)-1} \mathrm{LTVF}_{i-j}. \tag{5}$$

2.4. Fuzzy estimation of random noise in GPS data

For proper operation of stage "Adaptive Kalman Filtering," an accurate estimation of measurement errors in GPS data is needed. As such errors are larger, data filtering must be stronger and vice versa—if the error is small, filtering should be minimal. Errors in GPS data are of two types (Ogle et al., 2002): (1) systematic errors and (2) random errors. Systematic errors are mainly due to a low

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002

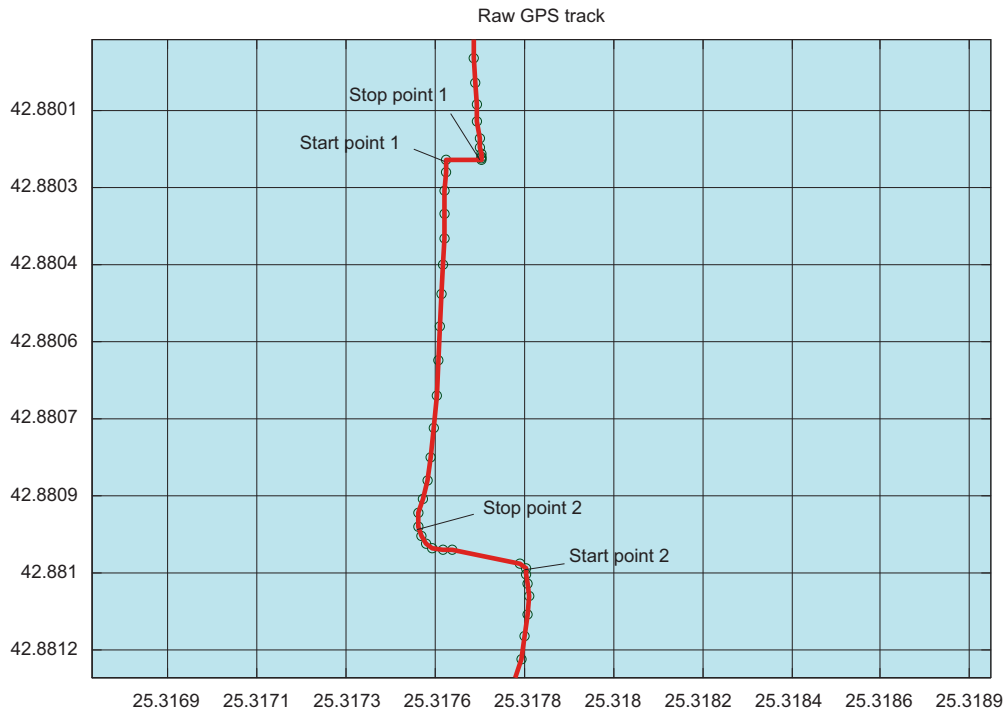


Fig. 2. GPS positioning errors in "movement-stop-movement" regions: 6.6 m deviation for region 1 and 23.1 m deviation for region 2.

Table 2
Random GPS data errors.

Error type	Average error (m)	Time dependent
Ionospheric effects	± 5.0	More than 1 h
Shift in satellite orbit	± 2.5	–
Satellite clock errors	± (1.5–2.0)	–
Tropospheric effects	± 0.5	More than 1 h
Multi-path effects	± 0.5	0.5–10 min

number of visible satellites, a high value of the Position Delution of Precision (PDOP), and a GPS receiver antenna placement. Systematic errors are identified and removed easily. The random errors are the result of shift in satellite orbits, satellite clocks errors, effects in the troposphere and ionosphere and multi-path signal reflection. Due to their nature, random errors are difficult to be identified and filtered. Table 2 shows the average deviation in the position for different types of random errors (Parkinson and Spilker, 1996).

To minimize the effect of random errors in practice, statistical smoothing techniques are most commonly used (Hastie et al., 2001). At this time, the best statistical technique to minimize the effect of random errors in GPS data is the Kalman filter. The discrete Kalman filter takes as input a stream of noisy measurements of the GPS receiver. It uses this measurement to estimate the state of the receiver. If input information is a stream of GPS positions, the Kalman filter could predict more accurate position information. The main problem is the precise estimation of measurement error, since it is random in nature.

It is proposed to use fuzzy logic to estimate the noise in GPS data. In the proposed algorithm two input fuzzy variables, Mean SNR (MSNR) and Number of Satellites, Elevation and Azimuth (NSEA) are used. The MSNR input variable is calculated as an average value of SNR for all visible satellites. The values of the MSNR are divided into 4 segments: S (Small), MS (Medium Small), MH (Medium High) and H (High). It is assumed that if the value of the MSNR is greater, the more likely it is that noise in the GPS data is low. The input linguistic variable NSEA account for three parameters: number of visible satellites, mean azimuth, and mean elevation. Thus, the number of input linguistic variables and time for realization of the algorithm are reduced. The accuracy of GPS receivers is proportional to the satellite geometry—relative positions of satellites and GPS receiver. The estimation of these positions is given by the Delution of Precision (DOP) parameter. The DOP value is inversely proportional to the volume of the tetrahedron that satellites and GPS receiver form. As the DOP value is lower, the more accurate the GPS receiver is. GPS data error is less so as the area of the base of the tetrahedron is higher and elevation is lower (see Fig. 3).

To obtain a larger volume of the tetrahedron, low satellite elevation is needed. But the errors due to processes in the troposphere and the ionosphere are big if elevation is low. For example, if satellite elevation is up to 5°, the error affecting measurement is about 3–10 times larger than the error if the satellite is at the zenith (Verhoel, 2009).

Since the DOP parameter does not take into account the influence of SNR and elevation of the satellites, it will not be used directly. The NSEA value is obtained by giving points for all parameters that are involved in its calculation. The maximum number of points of each parameter is 12. The values of the NSEA variable are as follows: S (Small), M (Medium) and H (High). As the NSEA value is greater, the lower the noise in GPS data:

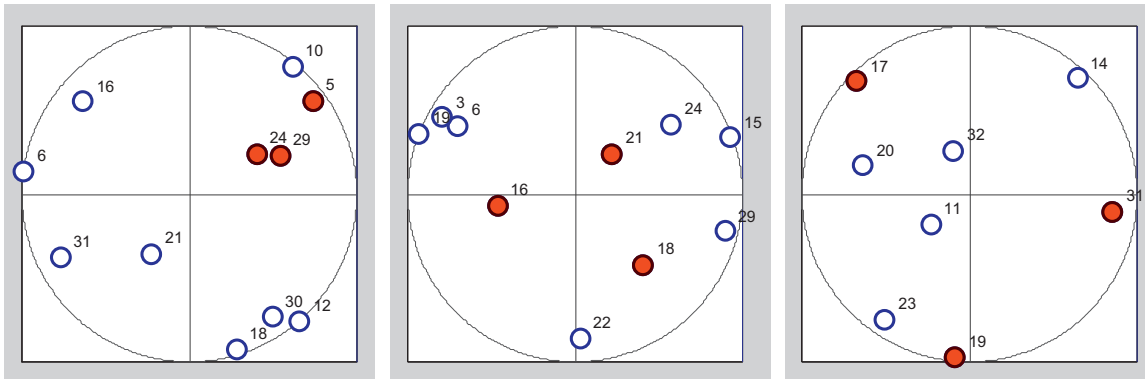
$$\mathrm{NSEA} = N_1 + N_2 + N_3, \tag{6}$$

where N_j, $j=1$–3 is the number of points for parameter j.

Parameter 1: Number of visible satellites. Each satellite is given 1 point; therefore, $N_1 = N$, where N is the number of visible satellites.

Parameter 2: Position of satellites above the horizon. We give greater importance to satellites that have low elevation:

$$N_2 = \frac{N_{max}}{N} \sum_{j=1}^{N} \left(1 - \frac{\delta_j}{90}\right), \tag{7}$$



HDOP=12.5, Mean elevation=46.3°, Mean SNR=29.0dB

HDOP=2.8, Mean elevation=62.3°, Mean SNR=26.0dB

HDOP=1.4, Mean elevation=20.7°, Mean SNR=34.0dB

Fig. 3. Sky view: relationship between HDOP, mean SNR and mean elevation at different positions of 3 visible satellites.

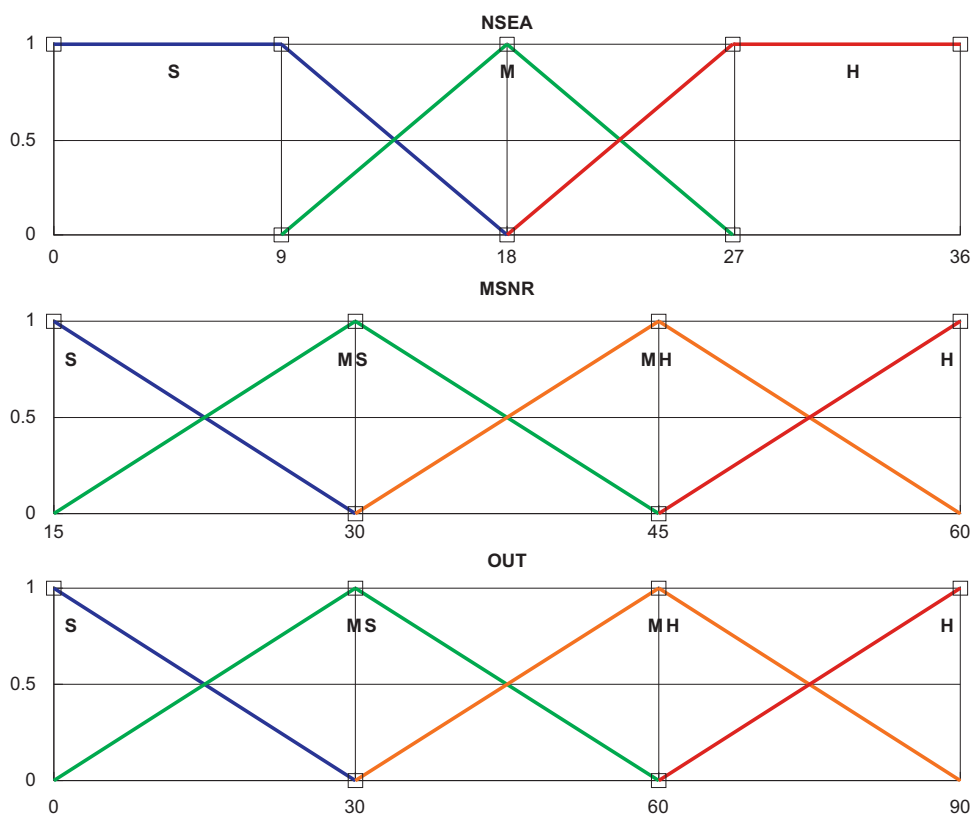


Fig. 4. Fuzzy system membership functions.

where δ_j is the value of elevation for the satellite j, and N_{max} is the maximum number of visible satellites. For movements on the ground $N_{max}=12$.

Parameter 3: Position of the satellites according to their azimuth. It is assumed that depending on the value of the azimuth, satellites fall into one of the following four quadrants: [0,90), [90,180), [180,270) and [270,360). We give greater importance if satellites are positioned uniformly in the quadrants:

$$N_3 = 3\sum_{j=1}^{4}[\min(n_j,N_m)/\max(n_j,N_m)], \tag{8}$$

where n_j is the number of satellites in quadrant j, $N_m=N/4$.

Fuzzy output variable *OUT* gives an estimate for the error in GPS data. The possible values for this variable are as follows: S (Small), MS (Medium Small), MH (Medium High) and H (High). As the value of *OUT* is greater, the stronger must be filtered GPS positions, as noise in them is greater.

Fig. 4 shows the membership functions of the input and output linguistic variables.

To obtain output linguistic variable OUT, 12 fuzzy rules of the following type are used:

if NSEA = value AND MSNR = value then OUT = value.

Fig. 5 shows the fuzzy rule matrix used.

To infere logical products for each rule, the Root-Sum-Square (RSS) method are used. The defuzzification of the data into a crisp output is realized by combining the results of the inference process and the computing of the fuzzy centroid of the area.

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002

OUT		MSNR			
		S	MS	MH	H
NSEA	S	H	MH	MH	ML
	M	H	MH	ML	L
	H	MH	ML	L	L

Fig. 5. Fuzzy rule matrix.

2.5. Correction of fuzzy output

When the transportation mode is "not-walking" an adjustment of the response of the fuzzy system is needed because it is possible to over-smooth curves. The adjustment is activated when the STVF value is above 50 km/h:

$$\mathrm{OUT}_i = \mathrm{OUT}_i(1-\mathrm{reduction}), \tag{9a}$$

where

$$\mathrm{reduction} = 0.9 \ \text{ if } \ \mathrm{STVF}_i > 150\,\mathrm{km/h}$$
$$\mathrm{reduction} = [10+0.8(\mathrm{STVF}_i-50)]/100 \ \text{ if } \ \mathrm{STVF}_i > 50\,\mathrm{km/h}. \tag{9b}$$

2.6. Adaptive Kalman Filtering

The Kalman filter is an optimal estimation method that has been widely used in GPS navigation systems (Mohamed and Schwarz, 1999; Kelly, 1994). For GPS navigation the level of measurement noise is environmental and user movement dependent. In this case the Klaman filter is not optimal and may cause unreliable results; for example, over-smoothing in curves. One solution for solving the problem is adaptive Kalman Filtering (AKF). Most existing techniques for on-line identification of the noise covariance matrix (Congwei, et al., 2003; Jwo and Huang, 2007) cannot be realised in real time when the application is started on mobile platforms.

The proposed AKF algorithm is used to filter only the positions of the user. The AKF smoothes positions by recursively modifying measurement error values. Estimation of the measurement noise is obtained in stage 4, "Fuzzy estimation of the random noise in GPS data." The Kalman filter algorithm can be described as a two step process: the prediction step (time update) that makes use of the learned system dynamics to anticipate the next step of the system, and the correct step (measurement update) that takes a new observed measurement and uses it to correct the system dynamics and state.

Discrete Kalman filter equations that describe the prediction step are

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$$
$$P_k^- = P_{k-1}A^T + Q, \tag{10}$$

where k is the discrete time, $\hat{x}_k^-$ is a priori state estimate, $\hat{x}_k$ is a posteriori state estimate, P_k^- is a priori estimate for error covariance, P_k is a posteriori estimate error covariance, A is time transition matrix, u_k is an additional known input parameter and Q is process noise covariance, $p(Q) \sim N(0, q)$.

Discrete Kalman filter equations that describe the correct step are

$$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$$
$$P_k = (I - K_k H)P_k^- \tag{11}$$

where K_k is the Kalman gain matrix, H is the time transition matrix for the observation process, z_k is the observed data (actual measurement) and R is the measurement noise covariance, $p(R) \sim N(0, r)$.

Since only the position (longitude, latitude) is filtered, u_k in (10) becomes zero (one dimensional Kalman filter). GPS data are analyzed in 1 s and therefore time transition matrix A and H are 1 s. In this case process noise was the same as the measurement noise. Thus, Kalman filter equations are reduced to the following form:

$$\hat{x}_k^- = \hat{x}_{k-1}$$
$$P_k^- = P_{k-1} + q \tag{12}$$

$$K_k = P_k^- / (P_k^- + r)$$
$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - \hat{x}_k^-)$$
$$P_k = (1 - K_k)P_k^-. \tag{13}$$

On implementation of the algorithm it is assumed that the process noise variance q is constant, and measurement noise variance r is the value of parameter OUT from the fuzzy system. A filtered position $\hat{x}_k$ is obtained based on an estimation of the difference between measured position z_k and a priori state estimate $\hat{x}_k^-$ (predicted residuals).

2.7. Track simplification

Due to the position filtering with AKF it is possible to use a local algorithm for track simplification. This ensures realization in real time, while maintaining the ability to extract the necessary for navigation process critical track points.

The algorithm is based on analysis of relationship between every 3 consecutive points: the last point, belonging to track p_i and the next two points p_{i+1} and p_{i+2} (see Fig. 6). The main task of the algorithm is to decide whether point p_{i+1} belongs to the track or not. This is implemented based on three parameters: distance D between last track point and point p_{i+1}; change in the direction α, $\alpha = |\alpha_i - \alpha_{i+1}|$, and distance d from point p_{i+1} to segment, which points p_i and p_{i+2} form.

The choice of the next point of the track is implemented based on three rules:

Rule 1. Select a candidate point. It is assumed that p_{i+1} is a candidate point if the distance from the last inserted point p_i and point p_{i+1} is greater than D, where

$$D = 15\,\mathrm{HDOP}_{i+1},$$
$$D_{max} = 20 \ \text{ if transportation mode is "walking".} \tag{14}$$

The value of D depends on the current accuracy of GPS receivers. Thus it is possible to filter points, obtained in low GPS accuracy.

Rule 2. Analysis of change in the direction of movements. The analyzed point belongs to a track if the following condition is fulfilled:

$$\alpha > \alpha_{Th1} \quad \text{AND} \quad \alpha < \alpha_{Th2}. \tag{15}$$

The value of the parameter α_{Th1} sets the degree of points reduction. For small values of this parameter, the track is described more accurately, but with a greater number of points. The parameter α_{Th2} has the task to filter the points, obtained due to errors in GPS data for small values of parameter D.

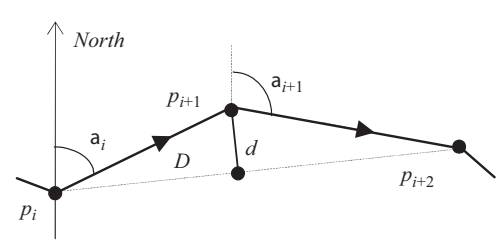


Fig. 6. Track simplification algorithm parameters.

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002

Rule 3. Analysis of the importance of parameter d. Rule 2 may miss critical points for regions containing smooth curves. Rule 3 aims to prevent this. If the distance d is above a predefined threshold, it is assumed that p_{i+1} belongs to the track:

$$d > \mathrm{COFF} \times \mathrm{HDOP}_{i+1}, \tag{16}$$

where the COFF parameter depends on the velocity:

$$\begin{aligned} &\mathrm{COFF} = 0.5 \ \text{ if } \ \mathrm{STVF}_{i+1} < 30\,\mathrm{km/h}, \\ &\mathrm{COFF} = 1.7 \ \text{ if } \ \mathrm{STVF}_{i+1} > 100\,\mathrm{km}/h, \\ &\mathrm{COFF} = 0.7 + (\mathrm{STVF}_{i+1} - 30)/70 \ \text{ if } \ \mathrm{STVF}_{i+1} \geq 30\,\mathrm{km/h}. \end{aligned} \tag{17}$$

Finally, point p_{i+1} belongs to the track if

$$\mathrm{Rule}_1 = \text{true and } (\mathrm{Rule}_2 = \text{true or } \mathrm{Rule}_3 = \text{true}). \tag{18}$$

3. Experimental results

The developed algorithm is used in the Java ME application for navigation along a track for the visually impaired and blind. This application does not require GPS maps and access to a WEB server, and can operate both in the presence and absence of DGPS and SBAS (WAAS, EGNOS and MSAS).

3.1. Settings

In order to analyze the characteristics of the proposed algorithm in the past 6 months, 200 tracks were recorded—50 records for each of the 4 track types. The track parameters are described in Table 3. The optimal number of points which describes the route is obtained manually using the GPS map.

To save tracks, the J2ME application, which records GPS data in the NMEA 0183 format on the flash disk of the mobile terminal at an interval of 1 s is used. The GPS data are analyzed using a specially designed GPS toolkit for Matlab™, which allows: visualization and filtering of GPS data, sky view, obtaining the optimal number of points, describing the track, using GPS maps, program implementation of the proposed track simplification algorithm, the possibility to compare the results obtained by the RSTS and DP algorithms, GPS distance, GPS bearing, etc.

Table 4 describes the threshold values of parameters that are used in the implementation of the algorithm.

Table 3
Description of GPS tracks.

Track type	Transportation mode	Optimal number of points	Mean track length (m)	Mean HDOP interval
1	Walk	13	1099	0.9–7.8
2	Walk	16	1267	0.9–5.6
3	Bus+walk	9	1641	1.2–5.5
4	Car	112	13,141	1.1–7.6

Table 4
Parameters threshold.

Algorithm stage	Parameters
Short and long time velocity filtering. Find transportation mode	$\alpha = 0.95$
Knowledge-based GPS data pre-processing and filtering	$M = 8$
Track simplification	$\alpha_{Th1} = 25$, $\alpha_{Th2} = 45$

3.2. Results

The following describes the importance of the various stages of implementation of the proposed algorithm for its characteristics.

3.2.1. Algorithm test, based on visual comparison

The simplification algorithms RSTS and DP are first compared visually. The results obtained are shown in Fig. 7. It is found that the RSTS and DP algorithms provide a similar representation of the shape of the tracks and a similar number of point's reduction.

3.2.2. Knowledge-based GPS data pre-processing and filtering

This stage aims mainly to filter out systematic errors in GPS data. The task is not to remove a maximum number of points, but only those that cannot be compensated in the "Adaptive Kalman filtering" stage. The results obtained are shown in Table 5. The reduction in the number of points is from 3.4% to 6%.

Fig. 8 shows the result obtained after testing the algorithm for finding transitions "movement-stop-movement" for tracks of type 3.

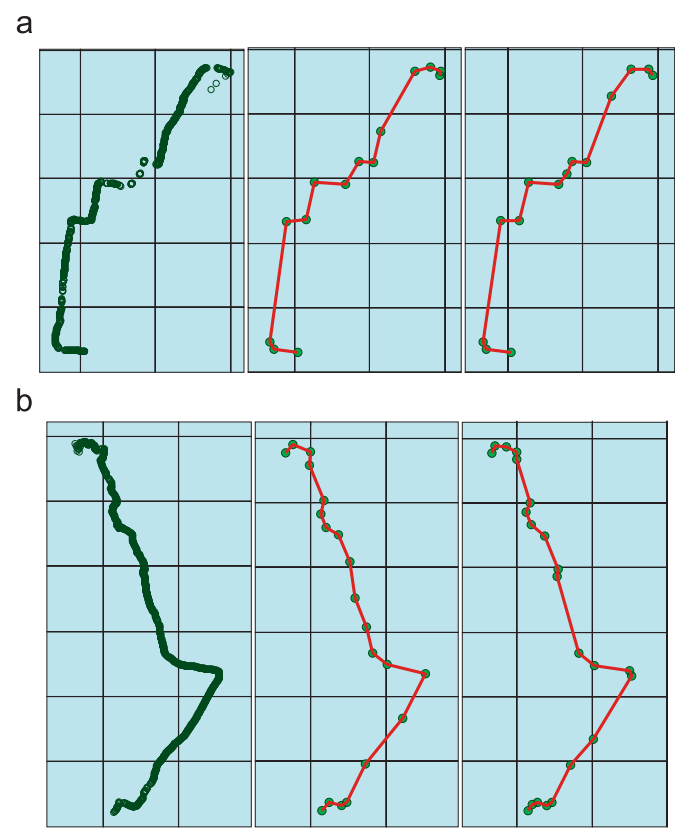


Fig. 7. Visual comparison of the tracks: (a) track type 1 and (b) track type 2 (raw data in the left, DP in the middle, and RSTS in the right).

Table 5
Results after knowledge-based GPS data filtering.

Track type	Raw data		Filtered data	
	No. of points	Track length (m)	No. of points	Track length (m)
1	1234	1192	1230	1087
2	1056	1409	1019	1280
3	352	1804	331	1762
4	1395	14,123	1348	13,140

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002

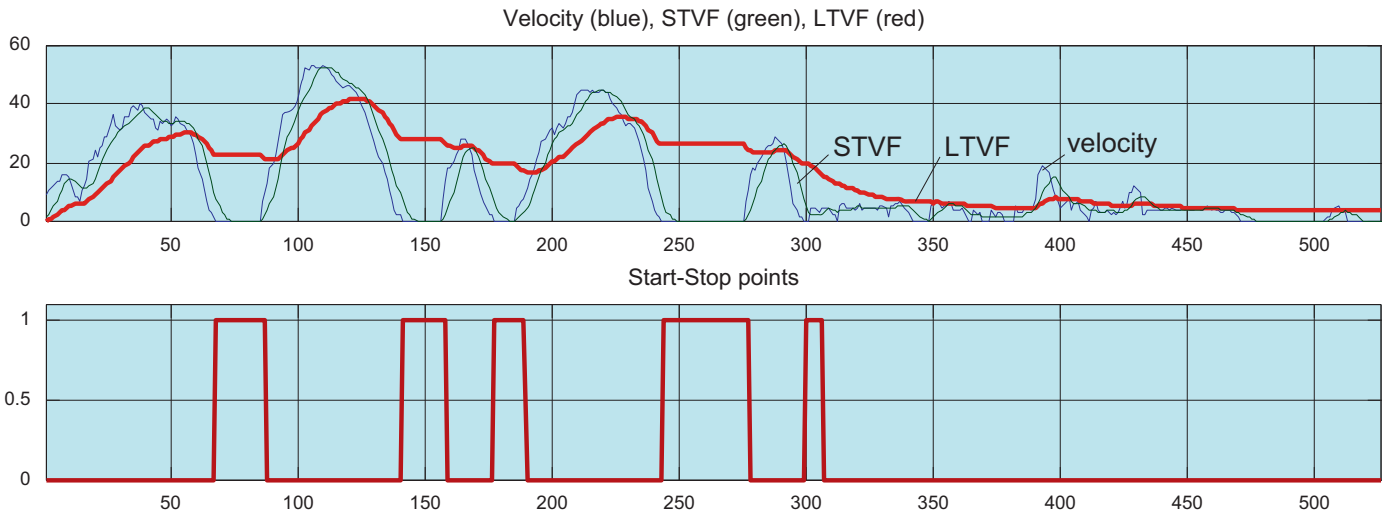


Fig. 8. Find start and stop points of transition "movement-stop-movement" regions.

Table 6
Importance of parameter OUT.

Track type	Deviation from optimal number of points (%)	
	Fuzzy OUT	OUT=const
1	7.1	28
2	23.8	42.9
3	10	40
4	3.6	23.8

Algorithm accurately detects over 95% of the positions of start and stop points when transportation mode is "not-walking." Errors are greatest in regions where transportation mode is changed.

3.2.3. *Fuzzy estimation of random noise in GPS data*

An analysis of the importance of parameter OUT, which gives an estimate for the GPS noise, is presented. Table 6 shows deviation from the optimal number of points, when fuzzy estimation of noise in GPS data is enabled and disabled. When using fuzzy logic, accuracy in determining the number of critical points is improved 4 times.

3.2.4. *Adaptive Kalman Filtering*

Since the RSTS track simplification algorithm is a local type, it is sensitive to errors in the GPS positions. The importance of the "Adaptive Kalman Filtering" stage for the quality of the algorithm RSTS is analyzed. For this purpose, two parameters, "number of critical points" and "positional error after track simplification" (rms error) are compared when AKF is active and inactive. The results are shown in Table 7.

We use distance-based measure to obtain positional error after track simplification. We calculate root mean square (rms) deviation between the original track and its simplified versions as a sum of normalized perpendicular distances for all points, describing the original track.

Experiments confirm that the RSTS algorithm is more sensitive to noise in GPS positions when compared to the DP algorithm. If AKF is active the results obtained by algorithms RSTS and DP are comparable. The number of points is reduced by an average of 96%, without changing the shape of the tracks.

3.2.5. *Analysis of flash memory used*

The proposed algorithm optimizes the use of both DRAM and flash memory. Table 8 shows the degree of reduction of flash memory used.

The flash memory is reduced 32.11 times when the GPX format for input and output files is used and 32.31 times for the CSV format.

3.2.6. *Analysis of battery life*

For mobile applications it is important how long they can be used without recharging the battery. To extend the tracking time the following are taken: An external GPS receiver with Bluetooth™ interface is used; the floating point arithmetic is minimized; application runs in the background; and intensive use of software threads, which are activated only when expected events occur.

Fig. 9 shows how the level of battery energy is changed over the time. During the benchmarking test, incoming and outgoing calls are not performed. The battery energy level was obtained at intervals of 1 min until the battery was completely discharged. The test was carried out with a mobile terminal Nokia N95 (battery type BL-5F 950 mAh).

Battery life test shows that when an external Bluetooth™ GPS receiver is used, tracking time is up to 15 h.

4. Conclusions

In this article, we have presented the GPS track simplification algorithm called RSTS. Algorithm allows for on-line realization for platforms with limited resources and is adaptive to the accuracy of the GPS receiver, noise in GPS data, and transportation mode.

We evaluate the RSTS algorithm using 4 types of tracks collected over 6 months. The test results show that the RSTS algorithm outperforms or is similarly efficient to the Douglas–Peucker algorithm, which is regarded as the best one for track simplification.

The algorithm is used in low-cost and widely accessible Java ME application (Ivanov, 2009) for navigation along a track for people who are visually disabled and blind. The algorithm is designed to reduce redundant track points and to find critical navigation process points. Maximum tracking time is up to 15 h, which is commensurate with the time of the battery life of most GPS receivers. The reduction of number of points that describe the

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002

Table 7
Importance of stage "Adaptive Kalman Filtering".

Track type	Parameters	Parameter values			
		AKF=ON		AKF=OFF	
		RSTS	DP	RSTS	DP
1	Critical points (reduction (%))	14 (98.87)	14 (98.87)	25 (97.97)	16 (98.70)
	Track length (m) (error (m))	1067 (32)	1069 (30)	1141 (42)	1151 (52)
	rms error (m)	0.82	0.712	1.84	2.36
2	Critical points (reduction (%))	21 (98.01)	20 (98.11)	31 (97.06)	25 (97.63)
	Track length (m) (error (m))	1259 (8)	1258 (9)	1262 (5)	1341 (74)
	rms error (m)	0.958	0.962	0.97	0.99
3	Critical points (reduction (%))	10 (97.16)	14 (96.02)	18 (94.89)	23 (93.47)
	Track length (m) (error (m))	1647 (6)	1650 (9)	1675 (34)	1755 (114)
	rms error (m)	1.16	0.94	0.88	1.71
4	Critical points (reduction (%))	108 (92.26)	104 (92.54)	246 (82.37)	132 (90.54)
	Track length (m) (error (m))	13,032 (109)	13,080 (61)	13,851 (710)	14,012 (871)
	rms error (m)	1.10	1.15	1.26	1.41

Table 8
Flash memory reduction.

Track type	Raw data file size (KB)			Output data file size (KB)	
	NMEA	GPX	CSV	GPX	CSV
1	464	317	70	7	2
2	391	402	89	10	2
3	138	154	34	5	1
4	506	570	126	47	8

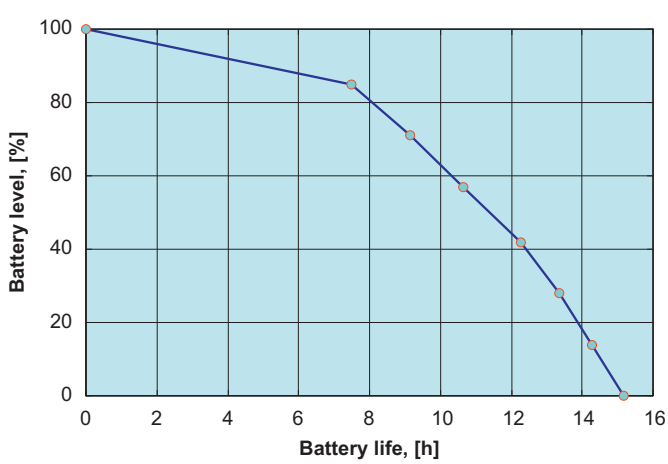


Fig. 9. Battery level benchmarking.

track in all transportation modes is more than 90%, and the output file size is reduced by over 30 times.

Future research will focus on the adaptive estimation of the value of parameter α_{Th1} that controls the degree of reduction of the number of points.

References

Barbeau S, Labrador MA, Perez A, Winters P, Georggi N, Aguilar D, et al. Dynamic management of real-time location data on GPS-enabled mobile phones. In: Proceedings of the second international conference on mobile ubiquitous computing, systems, services and technologies; 2008. p. 343–48.

Chen Y, Jiang K, Zheng Y, Li Ch, Yu N. Trajectory simplification method for location-based social networking services. In: Proceedings of the international workshop on location based social networks; 2009. p. 33–40.

Cheung Ch, Shi W. Positional error modeling for line simplification based on automatic shape similarity analysis in GIS. Journal of Computers and Geosciences 2006;32:462–75.

Congwei H, Chen W, Chen Y, Liu D. Adaptive Kalman filtering for vehicle navigation. Journal of GPS 2003;2(1):42–7.

Jwo Dah-Jing, Huang Chung-Min. An adaptive fuzzy strong tracking Kalman filter for GPS/INS navigation. In: Proceedings of the 33rd annual conference IECON'07; 2007. p. 2266–71.

Douglas D, Peucker T. Algorithm for the reduction of the number of points required to represent a digital line or its caricature. Journal of Canadian Cartographer 1973;10:112–22.

Hastie T, Tishiram R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. Springer; 2001.

Hershberger J, Snoeyink J. Speeding up the Douglas–Peucker line simplification algorithm. In: Proceedings of the fifth international symposium on spatial data handling, vol. 1; 1992. p. 134–43.

Ivanov R. Mobile GPS navigation application, adapted to visually impaired people. Journal of Information Technologies and Control 2009;1:20–4.

Kelly A. A 3D space formulation of a navigation Kalman filter for autonomous vehicles. Technical report CMU-RI-TR-94-19; 1994. Available from: 〈http://www.ri.cmu.edu/publication_view.html?pub_id=338〉.

McMaster RB. The geometric properties of numerical generalization. Journal of Geographical Analysis 1987;19(4):330–46.

Mohamed H, Schwarz KP. Adaptive Kalman filtering for INS/GPS. Journal of Geodesy 1999;73:193–203.

Ogle J, Guensler R, Bachman W, Koutsak M, Wolf J. Accuracy of global positioning system for determining driver performance parameters. Journal of Transportation Research Board 2002;1818:12–24.

Parkinson BW, Spilker JJ. Global positioning system: theory and applications. American Institute of Aeronautics; 1996.

Shi W, Cheung Ch. Performance evaluation of the simplification algorithms for vector generalization. Gartographic Journal 2006;43(1):27–44.

Verhoel P. EGNOS service definition document, open service, version 1.0; 2009. Available from: 〈http://ec.europa.eu/transport/egnos/programme/doc/2009_egnos_sdd_os_v1.pdf〉.

Visvalingam M, Whyatt JD. Line generalization by repeated elimination of points. Cartographic Journal 1993;30:46–51.

Zhung Y, Li Q, Xie X, Ma WY. Understanding mobility based on GPS data. In: Proceedings of the UbiComp'08; 2008. p. 312–21.

Please cite this article as: Ivanov R. Real-time GPS track simplification algorithm for outdoor navigation of visually impaired. Journal of Network and Computer Applications (2012), doi:10.1016/j.jnca.2012.02.002