Real Time Access to Multiple GPS Tracks

Karol Waga, Andrei Tabarcea, Radu Mariescu-Istodor and Pasi Fränti
Speech and Image Processing Unit, School of Computing, University of Eastern Finland
{kwaga, tabarcea, radum, franti}@cs.uef.fi

Keywords: Geographic information systems, GPS tracks, trajectory storage and visualization.

Abstract: Increasing availability of mobile devices with GPS receiver gives users the possibility to record and share a variety of location-based data, including GPS tracks. We describe a complete real-time system for acquisition, storage, querying, retrieval and visualization of GPS tracks. The main problems faced are how to store the data, how to access and how to visualize large amount of data. We propose to reduce the quantity of the data to be visualized, without affecting visualization quality. In order to achieve this, our system uses a fast polygonal approximation algorithm for different map scales along with a bounding box solution.

1. INTRODUCTION

Mobile devices with geo-positioning facilitate the acquisition of location-based data. This allows people to track their outdoor movements while performing physical exercises or when traveling. Companies can manage their geographical information in real-time (Martín et al., 2008) and track the movement of their own vehicles in order to solve problems such as fleet management (Jakobs et al., 2001) or traffic congestion (McCullough et al., 2011). The collected tracks are usually uploaded to an online system in order to be viewed, managed and analyzed. However, accessing and visualizing large amount of data is difficult and time consuming.

We present *MOPSI Routes*, a complete system for storage, retrieval and visualization of GPS tracks that overcomes the most common disadvantages of similar systems. For example, existing real-time systems usually do not have the possibility to plot large number of points and tracks on the map. In such cases, displaying becomes slow and visualizing overlapping tracks is difficult. Other solutions, such as TopoFusion (Morris et al., 2004), propose combining and intersecting GPS tracks in order to create trails and minimize the data needed to be displayed, although the goal, creating a GPS network of trails, is different. Our solution displays all the recorded tracks in real time by reducing the number of points that are plotted. This is done by fast multi-resolution polygonal approximation algorithm described in (Chen et al., 2012), which achieves better approximation result than the existing competitive methods. Furthermore, we minimize the time needed for drawing by using a bounding box solution for plotting only the points that are visible to the user.

MOPSI Routes is available as a part of MOPSI services (cs.uef.fi/mopsi) and addresses the issues of storage, querying, retrieval and visualization of GPS tracks, first described in (Waga et al., 2012b). Users can voluntarily upload their GPS tracks using our mobile application, which is available for most of the modern mobile operating systems (Android, Windows Phone, iOS and Symbian).

Similar research projects include GeoLife (Zheng et al., 2008), the system presented in (Alahakone et al., 2009) and StarTrack (Ananthanarayanan et al., 2009).

GeoLife (Zheng et al., 2008) is a project which focuses on visualization, organization, fast retrieval and understanding of GPS track logs. The main goal of the project is understanding people lives based on raw GPS data. The main contribution is visualizing GPS data over digital maps by indexing the GPS trajectories based on uploading behavior of users. Similarly to MOPSI Routes, tracks are searched using spatial range and time query.

The tool described in (Alahakone et al., 2009) is used for manipulating, integrating and displaying geographical referenced information. The main purposes for the tool are path planning and navigation of mobile objects. The tool can be used in several applications such as: tracking, fleet management, security management and industrial robot navigation. Similarly to our system, a spatial

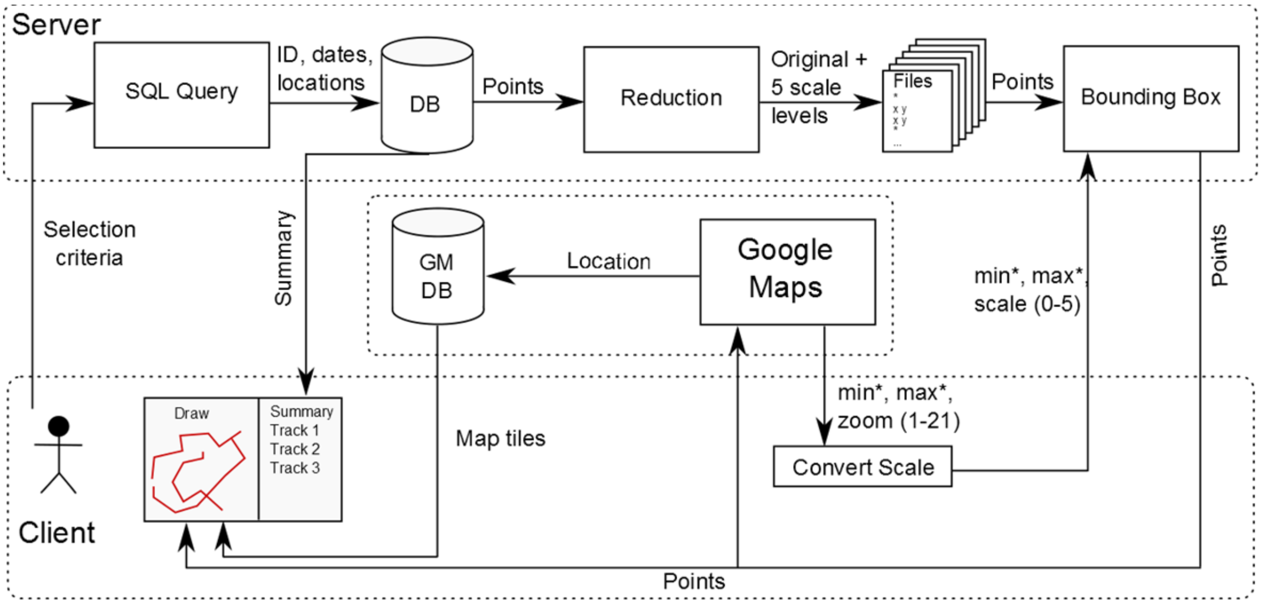

Figure 1. Typical workflow of MOPSI Routes.

database is used for storing tracks and points and Google Maps API to display the tracks. It presents a general approach in handling GPS data and it can be used in a variety of applications that use track recording, navigation and track planning. It requires that the user selects the points and defines the tracks, whereas our application automatically detects and segments the tracks.

StarTrack (Ananthanarayanan et al., 2009) and its improved version in (Haridasan et al., 2010) describe tracks of location coordinates as high-level abstraction for various types of location-based applications. The system supports the following operations: recording, comparison, clustering and querying tracks. Experimental results show that the system is efficient and scalable up to 10.000 tracks. The improved version was extended so that it can operate on collections of tracks, delay query executions and permits caching of query results. Other improvements are canonicalization based on road networks, and using track trees for computing similarity.

2. SYSTEM DESCRIPTION

MOPSI Routes can be accessed at cs.uef.fi/mopsi/routes. The typical workflow of the system is presented in Fig. 1, whereas Fig. 2 shows example of tracks collection from one user.

In the first step, user selects the tracks to be displayed by several criteria such as time, location, duration and length. Tracks that match the criteria are retrieved from database and processed before displaying to the user. During the processing phase, the points belonging to the retrieved tracks undergo approximation process that reduces the number of points needed for specific map scales. Points that are outside the visible area of the map are omitted by applying a bounding box. In the final step, the remaining points are shown on the map and the user can browse through them using map view (panning and zooming) or using list view to see additional information and statistics of each route.

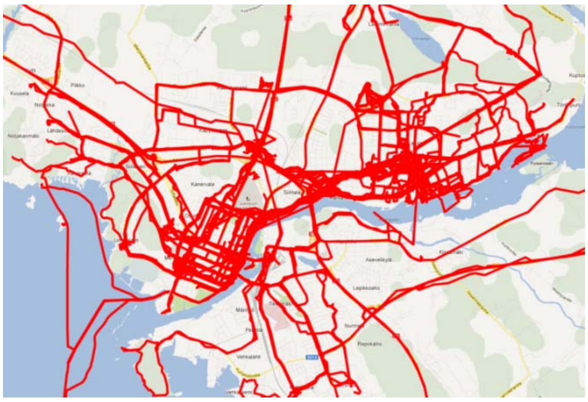

Figure 2. Example of user tracks collection.

2.1. Data Acquisition and Storage

MOPSI allows collecting tracking data using smartphones. The mobile application records the user’s location and timestamp at a predefined interval (usually 1-4 seconds). The data is saved to database on server immediately if Internet connection is available, or buffered on device if Internet connection is not available or the application is in offline mode.

Tracks are first saved as individual points in the database, and track objects are created and updated real time when new points are received. Each track object contains not only track points, but includes several basic statistics such as start and end time, bounding box and number of points. Segmentation and classification statistics are also stored. The analysis and classification of GPS tracks is described in details in (Waga et al., 2012a). Furthermore, each track is stored in its original and in a simplified form with reduced number of points. The approximated tracks are computed for 5 different zoom levels in order to speed up the visualization process. This limits the number of points drawn on the map without losing significant information about shape of the GPS track. The analyzed and the approximated tracks are computed immediately when the original GPS track is uploaded.

GPS tracks are created and updated real time and tracking points are handled immediately after they have been uploaded. This process requires maintaining and updating track statistics and information constantly when user is recording a new track. To ensure this, there is a process running constantly on server that checks periodically (every 1 minute) if any track needs to be updated. When new tracking points are uploaded, they are either used for creating a new track object or merged with the existing points and inserted into list of the track’s points in time order. The existing tracks are updated in the case that new tracking points belonging to an older track are received with significant delay caused, for example, by poor Internet connection.

2.2. Different Map Scales

The tracks recorded in our system carry far more data than needed for visualization. Full data is needed for analysis, and therefore, complete GPS tracks must be stored. However, in the rendering process for a web browser, reduced number of points is sufficient to present the shape of track to user. We apply here a multi-resolution polygonal approximation algorithm described in (Chen et al., 2012). The algorithm is fast and achieves good quality approximation of the tracks. It is applied to every uploaded track and returns approximation of the track in different resolution, i.e. map scale. We generate approximated tracks for 5 different scales of the map. The algorithm time complexity is O(N) (Chen et al., 2012) and the results are stored to avoid running algorithm repeatedly when the same track is displayed again.

Figure 3 shows an example of the original and approximated tracks. The original track contains 575 points and it is approximated in different map scales with 44, 13, and 6 points respectively. Suitable approximation error tolerance is selected for each map scale, and the visualization quality is therefore not affected by the approximation, but rendering time is reduced significantly, as shown in the results section.

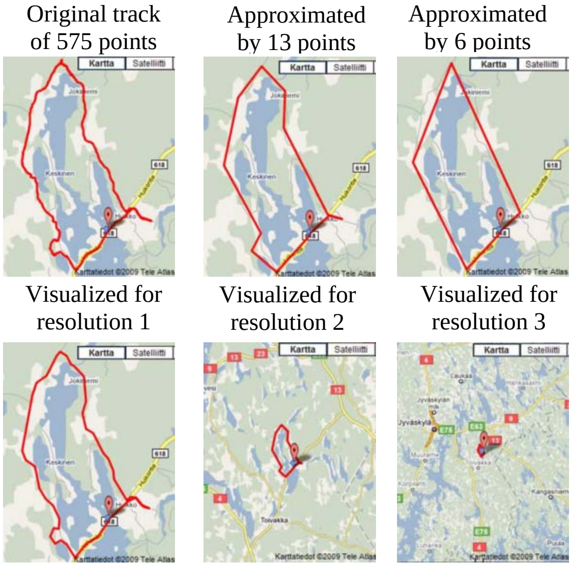
Original track of 575 points
Approximated by 13 points
Approximated by 6 points
Visualized for resolution 1
Visualized for resolution 2
Visualized for resolution 3

Figure 3. Visualization of a sample track.

2.3. Bounding Box

The tracks retrieved from database in the previous steps are ready to be drawn on map. However, we still attempt to minimize number of points passed to drawing module. In order to achieve the goal, we apply a bounding box solution.

The goal of the bounding box is to draw on the map only the points that are visible to user as shown in Fig. 4. Therefore, we select only points that user will see using the current map scale and location (*bounding box* of the map) at the moment of query. In addition, we draw also points that are outside the bounding box, but within immediate neighborhood (50% extension of screen size). In this way, we prepare for fast panning and zooming.

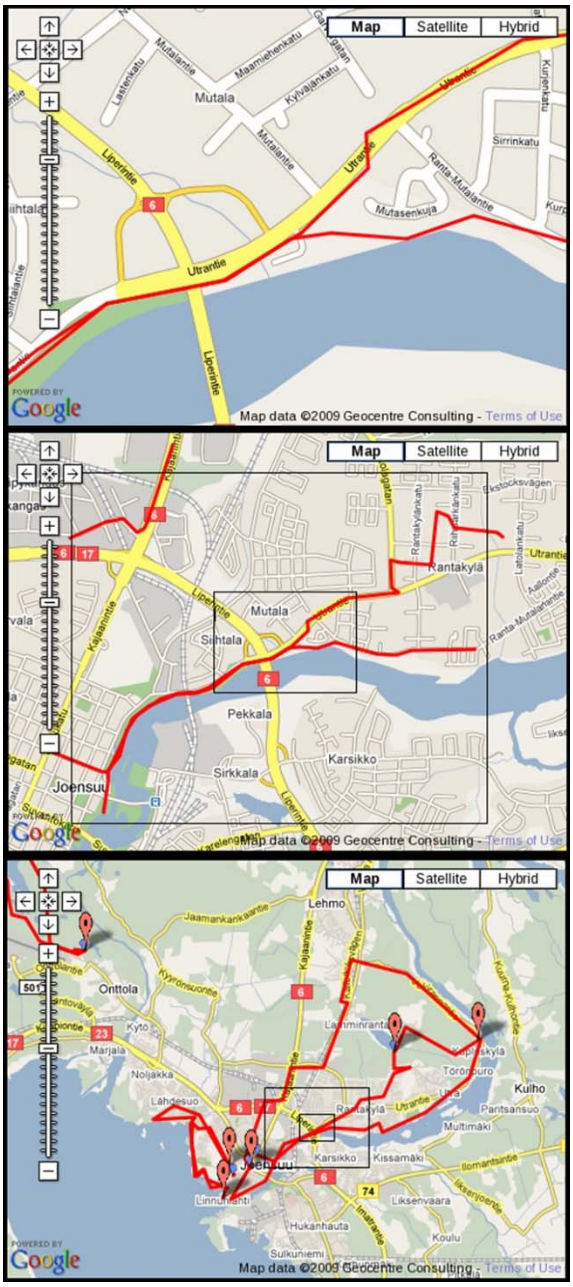

Figure 4. How the bounding box works (from top to bottom): what user sees on screen, what is drawn on map, all tracks selected.

The bounding box is implemented as a function that gets coordinate of north, east, south and west of the map visible on the screen. Map scale is also passed, so that points from the correct approximation can be selected. The function is applied to every track that is to be displayed on the map. For every point it checks if the point lies inside the bounding box. Time complexity of this solution is linear. The bounding box is computed entirely on server.

2.4. Displaying Tracks on Map

In MOPSI, we use Google Maps to visualize the data (see Fig. 5). However, user can select different type of maps that are displayed as overlay over Google Maps. We support OpenStreetMaps and detailed orienteering maps in Joensuu area where the most of the MOPSI users come from.

There are several search options available. The main search criterion is time, thus only tracks in the selected time period are shown. In addition, other criteria can be applied. For example, tracks can be filtered by minimum and maximum length and duration. Moreover, it is possible to search for tracks that start and end around a certain location.

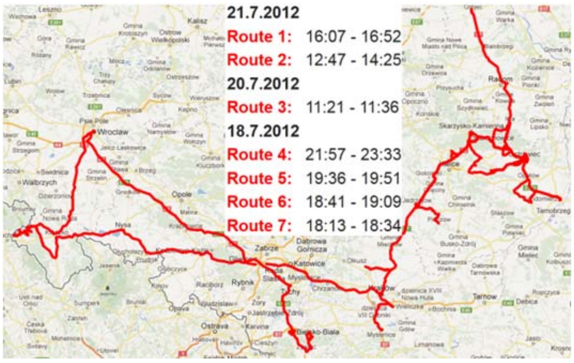

Figure 5. Displaying tracks on the map.

3. RESULTS

We measure time spent between sending request to the system and presenting the result to the user. The time elapsed from user's query to the time of displaying the tracks on the screen using our system is compared with the same system that does not have such features as track reduction or bounding box. Moreover, we study the performance and time benefit of the features on different sizes of track collection.

In all measurements, we ignore the time needed for data transfer. However, in weak Internet access this might become bottleneck, and therefore, we design the system so that it minimizes data transfer. That allows using the system on computers with slower Internet connection as well as on tablets that usually have slower connection and limited bandwidth.

Table 1. Collections used for our experiments.

User	Tracks	Points	Length (km)	Duration (h)
Pasi	784	1,216,039	8,535	669
Karol	650	1,015,939	9,655	442
Radu	429	613,684	4,604	188

We present measurements for 3 sample users from Table 1. The original tracks consist of large number of points. In MOPSI, there are users having over one million points, which shows the need for reducing the number of points being displayed as none of the browser could handle such large number of points (Chen et al., 2009). In Table 2, we show the number of points from the original tracks within the selected time period and the number of points from the approximated tracks. The zoom level of the map is selected in such manner that all the tracks are visible on the map using the formula:

$$z = ceil\left(\log_2 \frac{\pi \cos\left(min(|lat_1|, |lat_2|)\right)}{256 \cdot min(H_r, V_r)}\right)$$
$$H_r = \frac{1}{w} \operatorname{asin}\left(\sin\frac{\Delta lon}{2}\sqrt{\cos(lat_1)\cos(lat_2)}\right) \qquad (1)$$
$$V_r = \frac{\Delta lat}{2h}$$

where:

- w and h are map width and height in pixels
- map tile size is 256 pixels
- route bounding box is defined by the points (lat_1, lon_1) and (lat_2, lon_2) in radians, having Δlat and Δlon height and width
- H_r and V_r are bounding box height and width in meters divided by earth's radius

Figure 6 presents the time needed to display tracks in a selected period for three test users. The process is divided into three phases: querying database, computing bounding box and drawing in browser. Results show that the time needed for showing all the tracks of the user with the biggest collection is about 2.5 seconds.

Table 2. Number of points in original (left) and in the approximated tracks (right) in the selected time period for user Pasi.

	Original	**Approximated**
all	1,216,039	9,064
year	424,709	3,088
month	46,669	331
week	11,204	903
recent	3,328	141

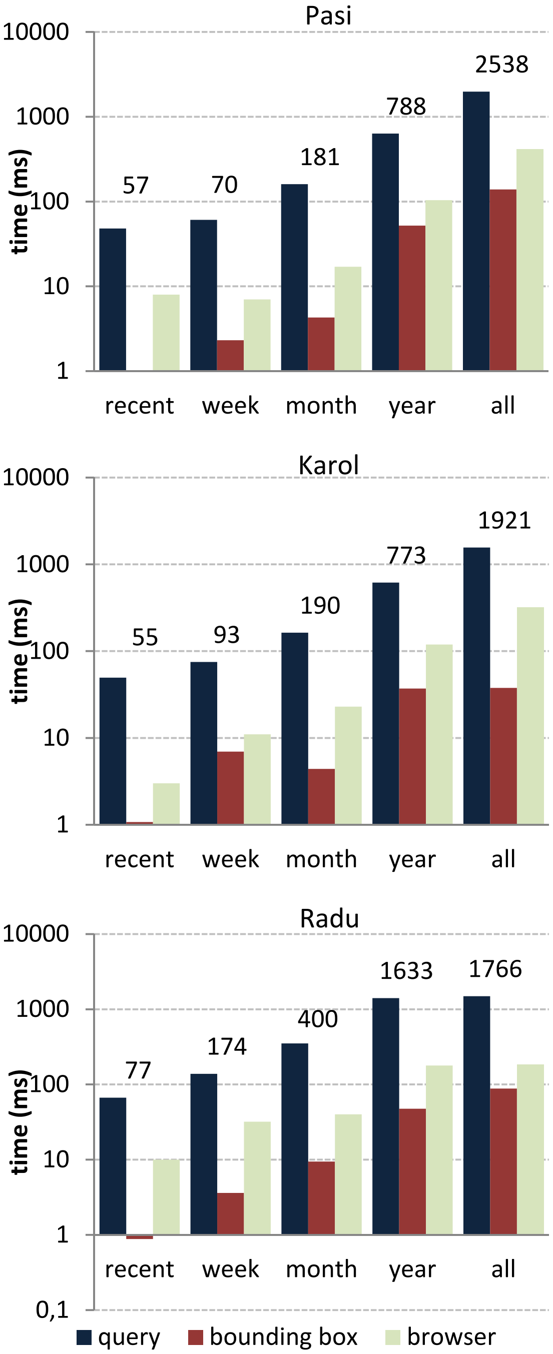

Figure 6. Display times of track collection for users Pasi, Karol and Radu.

Figure 7 shows average time percentages spent in each of the three phases. Querying data takes most of the time. Calculating bounding box is a fast process that additionally speeds up drawing tracks on map, so that it takes only 14% of time.

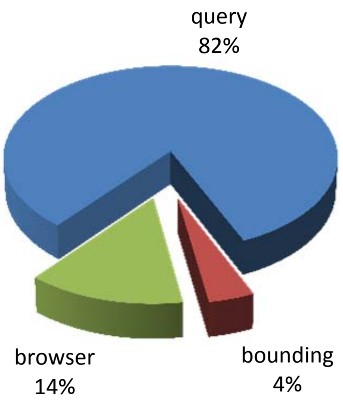

Figure 7. Average time percentage used for performing each operation of the system

The approximation algorithm is necessary to reduce the number of points displayed. Without it, it is not possible to display all tracks because the web browser would crash. The number of points browsers can handle depends on available resources. Displaying thousands of points significantly slows down web browsers. Nevertheless, even if browser can display all the points in tracks, the time needed for the process increases (see Fig. 8).

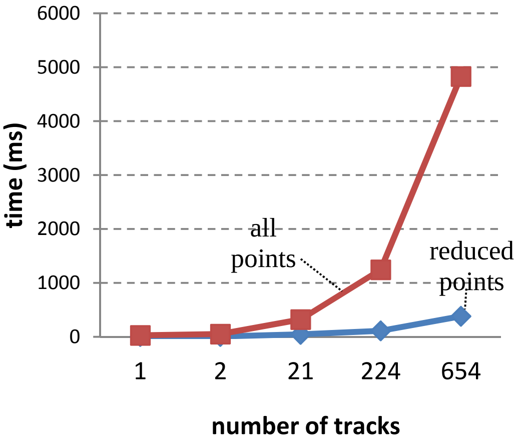

Figure 8. Time comparison between the system with and without track approximation for user Karol.

Table 3. Size of files (in bytes) with original and approximated tracks for user Karol.

	Original	Approximated
recent	14.000	148
week	14.000	148
month	346.000	2280
year	4.056.000	69.000
all	11.595.000	129.000

Bigger number of points slows down the bounding box algorithm and often leads to memory issues. Moreover, bandwidth needed to retrieve the points from the server is much bigger as shown in Table 3.

Experiments show that applying bounding box decreases time needed to draw tracks on map. Figure 9 shows a sample case from the experiments. In this case the same set of tracks was requested at the same zoom level, but the map was focused in two different places, Finland and Poland. In Finland the collection of tracks is big, whereas in Poland there are only several tracks available. Because of applying the bounding box solution, not all the tracks have to be displayed and the time to show the tracks when map shows a smaller number of tracks (Poland area) is significantly shorter. Figure 9 also shows how reducing number of points affects the display time.

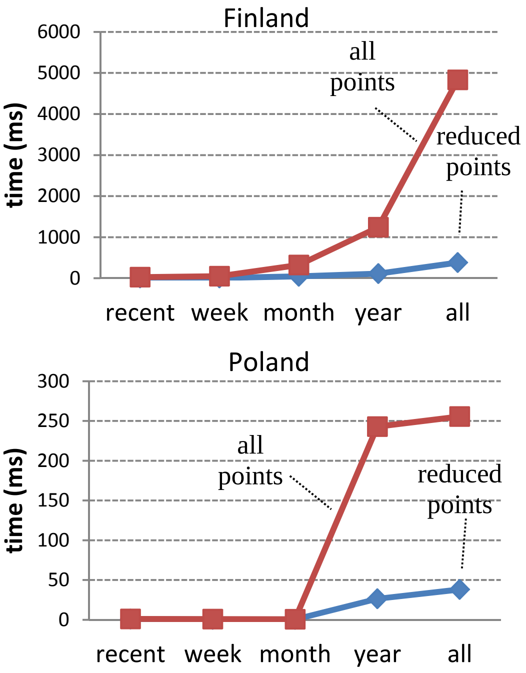

Figure 9. Example of querying the same track collection with map set to the same zoom level and focused in Finland with large collection of tracks (top) and Poland with small collection of tracks (bottom).

4. SUMMARY

We presented a complete real time system to collect and visualize GPS tracks. Our motivation is to offer a system that is capable of handling large amount of GPS data so that user can access them in real time. The results show that our system is efficient even with large point collection. The most important part of the process is the approximation algorithm reducing the number of points to be displayed. Combined with a bounding box solution, the requested tracks can be accessed within about 2.5 seconds and the collection can be browsed by panning and zooming with no delay caused by drawing many points. The developed track handling system is used as a basis for the systems that offer more advanced analysis of GPS tracks, such as similarity and movement type analysis of tracks.

Although, the system is efficient, there are still ways to improve it. For instance, now we reduce the number of points of one track only, but not when multiple tracks are overlapped. Further improvement could be achieved by clustering partial track segments. Moreover, the query phase should be optimized to minimize time needed to retrieve data.

REFERENCES

Alahakone, A. U., Ragavan, V. Geospatial Information System for Tracking and Navigation of Mobile Objects. ICAIM 09. Singapore, July, 2009.

Ananthanarayanan, G., Haridasan, M., Mohomed, I, Terry, D., Chandramohan, A. T. StarTrack: a Framework for Enabling Track-Based Application. ICMAS 09. Kraków, Poland, June 2009.

Chen, M., Xu, M., Fränti, P. A Fast O(N) Multi-resolution Polygonal Approximation Algorithm for GPS Trajectory Simplification. IEEE Transactions on Image Processing 21(5). 2012.

Chen, Y., Jiang, K., Zheng, Y., Li, Ch., Yu, N. Trajectory Simplification Method for Location-based Social Networking Services. International Workshop on Location Based Social Network. Seattle, USA, November 2009.

Haridasan, M., Mohomed, I., Terry, D., Chandramohan, A. T., Li, Z. StarTrack Next Generation: A Scalable Infrastructure for Track-Based Applications. 2010

Martín S., Cristóbal E.S., Gil R., Díaz G., Oliva N., Castro M., Peire J.. Finding the Way: Services for a Multi-View and Multi-Platform Geographic Information System. 2008, WEBIST (2), pp.267~270

Waga, K., Tabarcea, A., Chen, M., Fränti, P. Detecting Movement Type by Route Segmentation and Classification. ICCCNAW ‘12, Pittsburgh, USA, October 2012.

Waga, K., Tabarcea, A., Mariescu-Istodor R., Fränti, P. System for Real Time Storage, Retrieval and Visualization of GPS Tracks. International Conference on System Theory, Control and Computing. Sinaia, Romania, October 2012.

Zheng, Y., Wang, L., Zhang, R., Xie, X., Ma, W.-Y. GeoLife: Managing and Understanding Your Past Life over Maps. International Conference on Mobile Data Management. Beijing, China, April 2008.