EFFICIENT MANAGEMENT AND SEARCH OF GPS ROUTES

Radu Mariescu-Istodor

EFFICIENT MANAGEMENT AND SEARCH OF GPS ROUTES

Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences
No xx

University of Eastern Finland
Joensuu
2017

Academic dissertation
To be presented by permission of the Faculty of Science and Forestry for public examination in the Auditorium N100 in the Natura Building at the University of Eastern Finland, Joensuu, on January, 27, 2017, at 12 o'clock noon

Grano Oy
Jyväskylä, 2017
Editors: Pertti Pasanen, Matti Vornanen,
Jukka Tuomela, Matti Tedre
Distribution: University of Eastern Finland / Sales of publications
www.uef.fi/kirjasto
ISBN: xxx (nid.)
ISBN: xxx (PDF)
ISSNL: xxx
ISSN: xxx
ISSN: XXX (PDF)

Author's address: Radu Mariescu-Istodor
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: radum@cs.uef.fi

Supervisors: Professor Pasi Fränti, Ph.D.
University of Eastern Finland
School of Computing
P.O. Box 111
80101 JOENSUU, FINLAND
email: franti@cs.uef.fi

Reviewers: John Krumm, Ph.D.
Microsoft Research
Adaptive Systems and Interaction
14820 NE 36th Street, Building 99
Redmond, WA, 98052, USA
email: jckrumm@microsoft.com

Professor Hassan Karimi, Ph.D.
University of Pittsburgh
School of Computing and Information
135 North Bellefield Avenue
Pittsburgh, PA, 15260, USA
email: hkarimi@pitt.edu

Opponent:

Mariescu-Istodor, Radu
Efficient Management and Search of GPS Routes
Joensuu: University of Eastern Finland, 2017
Publications of the University of Eastern Finland
Dissertations in Forestry and Natural Sciences 2017; xx
ISBN: xxxx (print)
ISSNL: xxxx
ISSN: xxxx
ISBN: xxxxx (PDF)
ISSN: xxxx (PDF)

ABSTRACT

This research was focused on routes recorded using global positioning system (GPS) and examines methods of recording, storing, processing and visualizing those routes on a map. All of the methods discussed in this dissertation have been implemented in Mopsi, a location-based service with a collection of over 10,000 routes (11 million points).

I first discuss the creation of a system capable of working with a large amount of data, by applying two point-reduction methods. The two methods are cropping and polygonal approximation. These techniques allow users to load and visualize a large amount of data that would otherwise typically overload a browser.

Secondly, we used a grid to define four route measures: similarity, inclusion, novelty and noteworthiness. These measures feature in applications that deal with route search, ride-sharing and identifying taxi fraud. The similarity measure, C-SIM, allows real-time search on the Mopsi database. Our results showed that it is helpful for users who track their sports activities.

Navigation software is essential nowadays when visiting a large city. Our final contribution is CellNet, a method that uses the route database to infer the road network in an area, which is essential for navigation devices to function correctly. Using CellNet, we obtained higher quality results than those obtained by three conceptually different popular alternatives.

Universal Decimal Classification:

CAB Thesaurus: GPS route, grid, visualization, analysis, similarity, road network.

ACKNOWLEDGEMENTS

The work presented in this thesis was carried out in the Machine Learning Group at the School of Computing, University of Eastern Finland, Finland, between 2013 and 2017.

I would like to express my sincere gratitude to my supervisor, Professor Pasi Fränti, who has guided me throughout my studies. I especially appreciate his dedication to doing sports, which has influenced me and has improved my overall wellbeing and ability to work.

I am grateful to the entire Machine Learning Group, especially to Andrei Tabarcea and Karol (and Katalin) Waga, who helped me to understand the concepts required in my field of study and showed me how to do good research. Special thanks also to Najlah Gali and Sami Sieranoja for commenting on my thesis.

I would like to thank my parents, Camelia and Liviu Popescu, who have supported me and my passions in life, even though this meant I had to travel far from home to study. I often remember my grandmother, Constanta Istodor, and my aunt, Elena Vurfenescu, who both played a big role in raising me and helping to form my personality.

Lastly, I want to thank my girlfriend Iida Pirinen. I love her and all the things we do together, especially our travels – which have taught me a lot about the world. I'm also grateful to Iida's parents, Ritva and Risto, for showing me many different things to do in Finland.

Joensuu, 25th May 2017
Radu Mariescu-Istodor

LIST OF ABBREVIATIONS

GPS	Global Positioning System
WGS	World Geodetic System
PNG	Portable Network Graphics
LCSS	Longest Common Subsequence
EDR	Edit Distance on Real Sequence
ERP	Edit Distance with Real Penalty
DTW	Dynamic Time Warping
MGRS	Military Grid Reference System
UTM	Universal Transverse Mercator
UPS	Universal Polar Stereographic
XML	Extensible Markup Language
API	Application Programming Interface

LIST OF ORIGINAL PUBLICATIONS

This thesis is based on the following articles, referred to by the Roman Numerals I to V.

I Waga K., Tabarcea A., Mariescu-Istodor R. & Fränti P. 2013. Real Time Access to Multiple GPS Tracks, *International Conference on Web Information Systems & Technologies*, Aachen, Germany, pp. 293-299.

II Mariescu-Istodor R., Tabarcea A., Saeidi R. & Fränti P. 2014. Low complexity spatial similarity measure of GPS trajectories, *International Conference on Web Information Systems & Technologies*, Barcelona, Spain, pp. 62-69.

III Mariescu-Istodor R. & Fränti P. 2017. Grid-based method for GPS route analysis for retrieval (submitted).

IV Mariescu-Istodor R. & Fränti P. 2016. Gesture input for GPS route search, *Joint International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, Merida, Mexico, pp. 439-449.

V Mariescu-Istodor R. & Fränti P. 2017. CellNet: Inferring road networks from GPS trajectories (to submit).

AUTHORS' CONTRIBUTIONS

I) The author implemented the system and performed the experiments together with his colleagues. The idea originated with Karol Waga. The co-authors wrote most of the paper and refined the methodology.

II) The idea originated with the author and was jointly refined through discussion with the co-authors. The author implemented the method, conducted the experiments and wrote the paper.

III) The idea was developed by both authors jointly. The author implemented the methods and performed the experiments, which were later refined by the co-author. The paper was written by the author.

IV) The idea originated with the author and was polished together with the co-author. The method was implemented by the author. The experiments and authorship of the paper were handled jointly by both authors.

V) The idea originated with the author and was later refined by the second author. Implementation and experiments were performed by the author. The two authors wrote the paper jointly.

CONTENTS

1 INTRODUCTION

In recent years, global positioning system (GPS) technology has become widely available. As a result of this increase in availability, there has been a boom in the amount of location-based data that are recorded, stored and downloaded on a daily basis. Such data include geo-tagged photos, videos, service locations and GPS trajectories. The trajectories are referred to as *routes*.

Location information often represents a point on the surface of the Earth. It is typically defined using the world geodetic system (WGS) coordinates: latitude and longitude. This information is obtained by GPS sensors available in many mobile devices nowadays, such as mobile phones, smart watches and tablets.

Location information can be used in many ways. Some examples include finding the location of lost or stolen items such as bicycles, cars and mobile phones. Pets or loved ones are also often tracked in case they go missing. Many people who play sports feel safer when sharing their location so that others know their whereabouts. Geo-tagged photo albums allow the grouping of large picture collections by location.

Sequences of GPS locations may be recorded to form routes. Various applications store, manipulate and display routes for several purposes – such as sports tracking, ski-track maintenance, vehicle tracking, fleet management, road maintenance and wildlife surveillance. Figure 1 shows some examples of these applications.

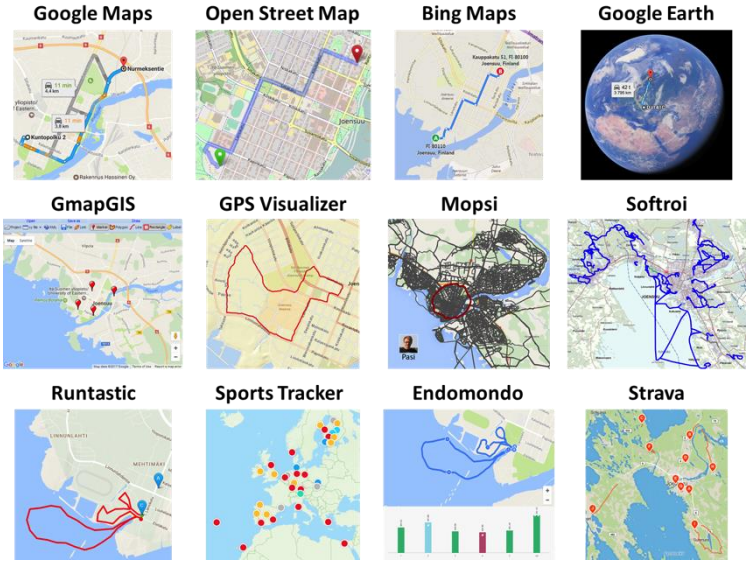


Figure 1. GPS routes displayed by various software.

Routes contain a lot of information and handling them is not trivial. In this thesis, I present efficient methods for storing, analysing, searching and visualizing routes. The methods have been implemented and tested inside Mopsi, a real-world environment.

1.1 MOPSI

Mopsi is a social network that helps people to discover who and what is around them. Its features include photo sharing, live tracking and chatting with friends. Mopsi can be found on the web at http://cs.uef.fi/mopsi and mobile applications exist for all major platforms (iOS, Android, Windows Phone, Symbian). They can be downloaded from the respective stores or from http://cs.uef.fi/mopsi/mobile.php.

Mopsi was developed by the Machine Learning Group, School of Computing at the University of Eastern Finland. It provides location-based services, such as search, recommendation, route tracking, geo-tagged photo collection and bus schedules. Mopsi has more than –

- 2,400 registered users
- 35,000 geo-tagged photos
- 400 points of interest
- 10,000 GPS routes.

The main topics of research to date involving Mopsi are as follows:

- route management and visualization **[I]**
- route search **[II, III, IV]**
- road network inference **[V]**
- transport mode detection [Waga et al. 2012]
- location-based recommendations [Fränti et al. 2011, Waga et al. 2011, Waga et al. 2012]
- web page summarization [Rezaei et al. 2015, Gali et al. 2015, Gali and Fränti 2016].

O-Mopsi [Tabarcea et al. 2013] is a mobile orienteering game built using data and modules in Mopsi.

1.2 RESEARCH CHALLENGES

Mobile users typically have many routes in their collection. Such route collections are difficult to manage. Certain challenges in processing routes are caused by GPS inaccuracies, missing points, different recording intervals and varying movement speed. Another challenge is the large size; to store the routes for fast retrieval and to display

them on a map is difficult without overwhelming the browser. In **[I]** we grappled with all of these challenges and provided methodological solutions.

The most popular route similarity measures are slow (quadratic time complexity) and unintuitive for the average user. Current approaches are point-based and borrow concepts from text matching, such as edit distance [Chen et al. 2005, Chen and Ng 2004], longest common subsequence [Vlachos et al. 2002] or time series analysis [Hamilton 1994, Berndt & Clifford 1994] such as dynamic time warping [Zheng and Zhou 2011]. Such point-based measures are unintuitive to typical sports tracking users, who understand routes as curves or shapes on the map. For example, to the user, a perfectly straight route consisting of 10 points is identical to the same route with 8 midpoints removed. However, the similarity as scored by the above measures is low.

Frechet [Eiter and Mannila 1994] is a similarity measure between two curves. It can be described as the minimum length of a leash an owner needs to walk a dog, when the owner travels on one curve and the dog on the other. While useful in applications such as route clustering, this type of measure is not what sports-tracking users expect. They are more interested in seeing whether the routes are recorded in the same area so they can objectively compare performances. Two routes belonging to two different users may be the same except for the start or end parts, which depend on the users' homes. Such two routes will have a low Frechet similarity even though they can be compared.

We defined a fast, linear time similarity measure called C-SIM **[III]**, which focuses on the spatial aspect of routes. C-SIM is inspired by the Jaccard set similarity coefficient; it measures the area two routes have in common as a proportion of the total space covered by the two routes. C-SIM is fast enough to allow real-time route similarity searches in large databases. To allow for fast computation, we used a grid to represent routes as sets of cells. We investigated and discussed methods of defining a good measure using the grid **[II]**.

Another challenge is searching for routes. Traditional solutions present routes as a time-ordered list or display the routes one-by-one on the map. Users often forget the date when a route was recorded. In this situation, users are forced to search in the list, one by one, to find a specific route. Showing the collection on the map successfully limits the data in the region a user is interested in; however, routes often overlap and become difficult to distinguish. We propose to use route similarity as a tool for efficient searching in large collections. As a result, users are able to search for routes based on the route shape. This shape input can be a similar route obtained from the database **[III]** or a free-form shape **[IV]** drawn by the user on the map.

The final challenge we address is to automatically generate a road network using GPS routes. Current methods are based either on satellite (aerial) image analysis [Tavakoli and Rosenfeld 1982, Hu et al. 2007, Barsi and Heipke 2003] or on GPS route analysis. Many conceptually diverse methods use GPS routes; for example, route merging methods [Cao and Krumm 2009], clustering methods [Edelkamp and Schrödl 2003] and visual methods [Davies et al 2006]. These methods contain a list of parameters that need to be carefully chosen, depending on the properties of the route dataset. Optimizing these parameters is time consuming and can make a dramatic difference to the quality of the outcome, as demonstrated by Biagioni and Eriksson [2012]. Moreover, a road network generated by these methods is unnecessarily complex; relatively straight road segments have many points in their definition. To aid in these aspects, we developed a new two-step method called CellNet **[V]**. The method uses a grid to find road intersections and then connects the intersections to obtain the resulting network. It produces higher accuracy than other state-of-the-art methods. The network generated by CellNet is optimized in terms of size, requiring 75% less storage space than any other method.

All methods presented in this thesis were implemented and tested within the real datasets provided by Mopsi users. Figure 2 summarizes the components and methods used in this research.

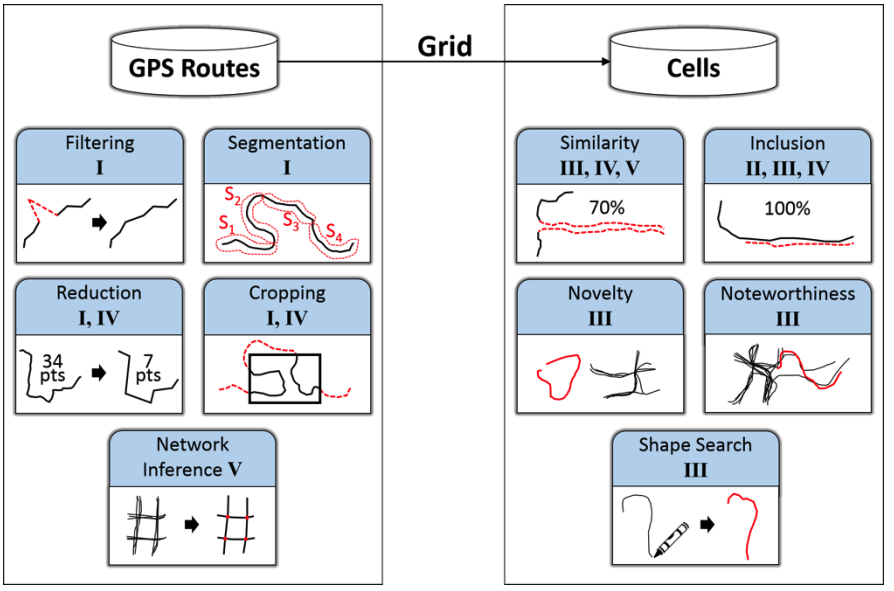


Figure 2. Different route operations available in Mopsi. They are grouped depending on whether they work with the cell approximation or with the routes themselves.

2 ROUTE HANDLING

Storing, accessing and visualizing large amounts of data on maps is computationally time-consuming. Several systems aim at providing such functionality [Alahakone and Ragavan 2009, Almer and Stelzl 2002, Follin et al. 2003, Horozov et al. 2006, Lehtimäki et al. 2008, Zheng et al. 2008]. StarTrack, described in Ananthanarayanan et al. [2009] and Haridasan et al. [2010], is most similar to the one we developed. StarTrack was tested with up to 10,000 routes. However, it does not address the problem of displaying the routes in real time; nor does it attempt to detect the transportation mode.

One of the most popular route collecting services are sport trackers, such as Sports Tracker, Endomondo, Runtastic and Strava. They allow users to record routes and evaluate their performance through comparison with past activities, or by comparing the user's performance to that of other users. In this chapter I describe current state-of-the-art methods for route management and compare these methods with the way routes are handled in Mopsi (Figure 3).

Figure 3. Mopsi user Pasi's route collection between 2008 and 2014, consisting of 915 routes with a total of 1,798,685 points. Pasi travelled a total of 11,775 kilometres, accumulating 500 hours of data. Map is centred in Joensuu, Finland.

2.1 ROUTE RECORDING

A GPS location is defined as a point **p** = {latitude, longitude, timestamp}, where the first two values represent the WGS coordinates on the surface of the planet and the last value is the unix timestamp at the moment the location is recorded. A sequence of such points forms a route $\mathbf{R} = \{\mathbf{p}_1, \mathbf{p}_2 \ldots \mathbf{p}_n\}$. The route points are usually presented in the order they were recorded in. This feature facilitates certain operations, such as drawing the points on a map.

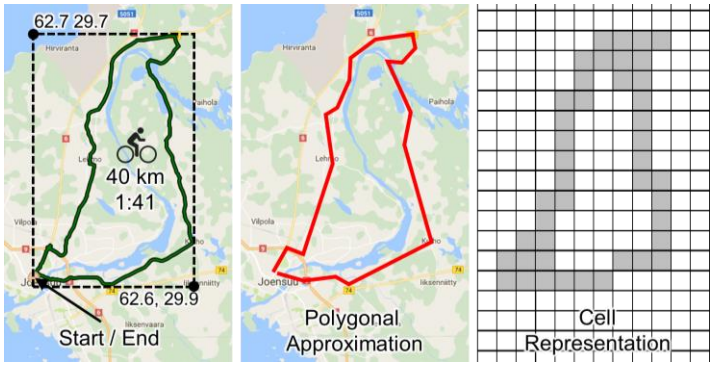
In Mopsi, points are recorded at a fixed time interval. The interval is usually between 1 and 4 seconds but can be changed in the application settings. Recorded points are buffered in the device's internal memory and they are periodically uploaded to the server database if internet connection is available. Many applications (including Sports Tracker and Endomondo) do not allow changing this parameter, but their recording interval is usually within the same 1-to-4-second range.

2.2 ROUTE STORING

When new points are uploaded to the server, route objects are created or updated as follows. If the uploaded points are recorded within a 3-minute time period and within a distance of 1.5 km from the last point of a route, the route is updated with the new points; otherwise a new route object is created. Unlike other systems (Sports Tracker, Endomondo, Runtastic and Strava), this setup enables users to cope with battery limitations or a device or application error (requesting restart) by enabling the use of multiple devices to record a single route. Routes can be manually edited later in Mopsi, but the 3-minute, 1.5 km default segmentation is usually enough to cover typical situations. In case of such error, other applications allow manual processing – such as merging two routes – on the website. Routes can be stored efficiently using the method described by Chen et al. [2012].

Once uploaded, for every route we computed and stored the following features in a MySQL database:

- start and end point
- bounding box
- distance
- movement type
- polygonal approx.
- cell representation.



The MySQL table contains pointers to the files containing the route points. StarTrack uses XML files to store the route points. We used simple text files in which each row contained the latitude, longitude and timestamp. The XML file structure is useful to attach notes, photos or other information to a specific route as metadata. We stored the points in simple text files because these occupy about a third of the space relative to the XML formatting standard (Figure 4). Additional metadata, such as the transportation mode, are stored in the MySQL database. The filtered polygonal approximated and segmented variants are also stored as files that MySQL points to.

Figure 4. A route consisting of 3 points represented in XML format (left panel) and text format (right panel). The XML file contains 335 characters whereas the text file has only 104.

2.3 ROUTE ANALYSING

In Mopsi, any route can be analysed. Segmentation is performed together with automatic transport mode detection for each segment [Waga et al. 2012]. Each segment is coloured differently according to the transportation mode. All popular sports tracking programs allow to segment using a fixed length or duration. In Mopsi, the segments are defined in a way that minimizes the speed variance of each segment. Such segments are useful when displaying routes with multiple transportation modes. Figure 5 shows that the running segment in the middle is perfectly isolated and accurate statistics can be viewed for only that section. In Runtastic it is only possible to segment the route at fixed length or duration, therefore, the running portion falls under the fourth and fifth segments making it difficult to interpret.

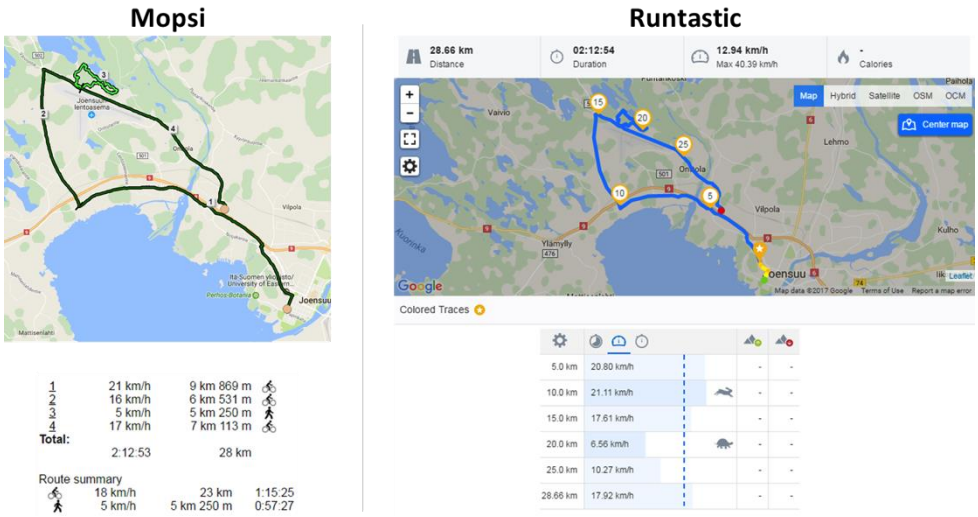


Figure 5. A route that features cycling, orienteering and cycling, shown in Mopsi (left panel) and Runtastic (right panel). The orienteering (running) segment is separated from the cycling segments in Mopsi.

Other features such as stop detection, roundness computation and showing photos taken by users on the route are also available in the analysis screen. Routes are filtered to remove outlier points caused by fluctuations in GPS accuracy. These points are computationally simple to identify and remove because they typically deviate away from the actual location, causing an impossibly high speed.

Mopsi lacks certain popular features, such as calorie and power output display, which other sports tracking software collects through additional pieces of hardware connected to the user's body or bicycle.

2.4 ROUTE VISUALIZING

We accessed routes in Mopsi by selecting a user and a time period (Figure 6). The time interval can be chosen to show the most recent day or last week, last month or last year's activity; showing the entire collection or choosing a user-defined time interval is also supported. This is efficiently done by querying the timestamp of the start point of the routes. This feature is available in all sports tracking software. Other programs (Sports Tracker and Runtastic) also allow the user to search or group the results based on transportation mode, distance travelled and duration.

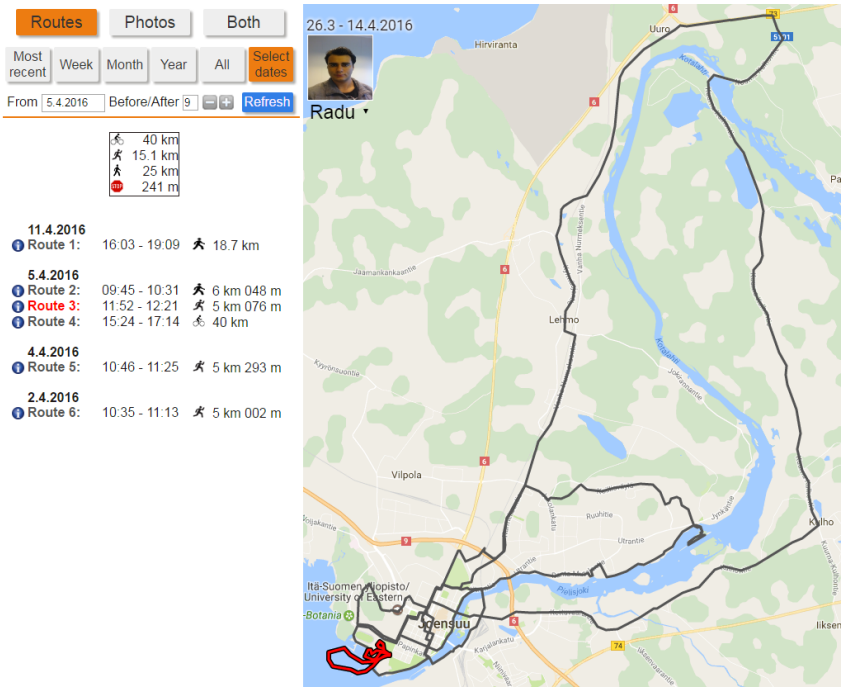


Figure 6. Mopsi tracking activity of user Radu over a few days. The entire route shapes are preserved. Routes are clickable inside the list or on the map.

Figure 7. The same route collection as in Figure 6, displayed by two sports tracking programs; routes are represented by their start points. Sports Tracker colours the points based on the transportation mode. Runtastic can display one route at a time when clicking on the start point marker.

In Mopsi, the selected route collection is presented in two ways: in a list and on the map. The list is ordered with respect to time. Routes recorded on the same day are grouped together to enhance usability. On the map, routes are shown as grey lines. A route can be selected by clicking it on the map or in the list, and a selected route is

highlighted in red. The movement type and distance are shown for each route in the list. A summary of the data collection per transportation mode is shown at the top of the list.

Visualizing route collections on the map is a difficult task because of the large number of points they contain. Therefore, many other applications lack this feature and can display only one selected route at a time (Table 1). The map system used is typically Google Maps[1] or Open Street Map[2] (OSM). Runtastic also uses Open Cycle Map[3] (OCM). Mopsi allows for the display of specially designed orienteering maps as well, provided by Kalevan Rasti[4], as overlays on top of the Google map.

Table 1. Features of popular sports tracking applications.

	Mopsi	**Endomondo**	**Runtastic**	**Sports Tracker**	**Strava**
Map type	Google OSM Kalevanrasti	Google	Google OSM OCM	OSM	Google OSM
Displays collection	✓		✓	✓	
Displays single route	✓	✓	✓	✓	✓

To avoid overwhelming the map, some applications show the starting locations only (Figure 7). In contrast, Mopsi can display large route collections by showing the full shape of the routes, which enables users to better understand their collection (Figure 6). In this way, a user can see at a glance that the collection is not limited to the city centre, as suggested by Sports Tracker and Runtastic; several routes actually pass through different towns. When displaying collections, the problem of overlapping route segments also becomes apparent. It is common that the starting points of routes overlap near the user's home or workplace. Clustering these start points could help to improve the user's experience [Rezaei and Fränti, 2017].

Our solution is to limit the route points using two strategies. First, when users browse a collection on the map, they need to see the overall shape of the route but not every detail. The shape can be well preserved by applying polygonal approximation [Chen et al. 2012], which reduces the number of points. We used approximations with varying reduction levels, which are used at different zoom levels of the map (Figure 8).

[1] https://www.google.fi/maps
[2] http://www.openstreetmap.org
[3] https://www.opencyclemap.org
[4] http://wp.kalevanrasti.fi

Figure 8. Polygonal approximation at three different levels. The 10-point approximation is suitable for the current zoom level. Original route has 110 points.

The second strategy is to crop the collection according to the screen boundary. Only points within the current map borders are loaded, together with an immediate neighbourhood (50% extension of screen size). This allows panning the map by a small amount without the need to reload new data (Figure 9). The cropping process is hidden from the user and does not interfere with usability.

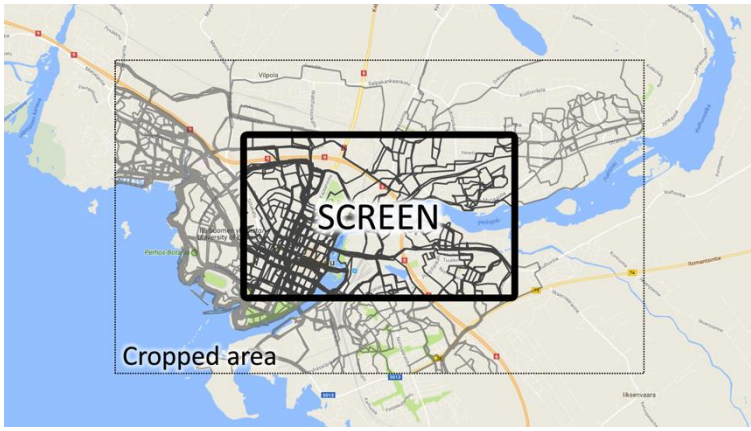


Figure 9. Cropping of a route collection. Only route segments inside the screen area and screen neighbourhood are plotted on the map.

An approach by Morris et al. [2004] aims to minimize the data displayed by combining the route segments that overlap. In Mopsi, we avoid this solution so that users are allowed to interact with each individual route by clicking it on the map.

To understand how effective the cropping and reduction process is for limiting the amount of data, we first investigated three highly active Mopsi users and their route collections. The data are shown in Table 2.

Table 2. Tracking statistics for 3 active users in Mopsi, grouped by time period.

User		Week	Month	Year	All
Pasi	routes	3	17	230	1,704
	points	2,030	43,635	548,379	3,648,923
	length (km)	33	230	8,040	27,384
	time (hours)	3	21	306	2,030
	size (MB)	221	1,719	22	145
Radu	routes	2	13	82	1,235
	points	650	14,376	100,542	1,383,318
	length (km)	8	140	1,258	23,138
	time (hours)	0.7	15	114	1,034
	size (MB)	24	566	3.8	53
Matti	routes	9	14	148	412
	points	2,086	5,875	98,237	293,207
	length (km)	39	90	1,642	4,160
	time (hours)	3	8	138	350
	size (MB)	74	210	3.5	11

Figure 10 illustrates the process of querying and displaying these collections in full, without any reduction. Figure 11 shows the process with polygonal approximation and cropping applied. We disregarded the time required to download the points from the server, as this duration varies depending on factors such as internet speed and bandwidth. For the display, we used the zoom level that allowed all routes to be visible on the map.

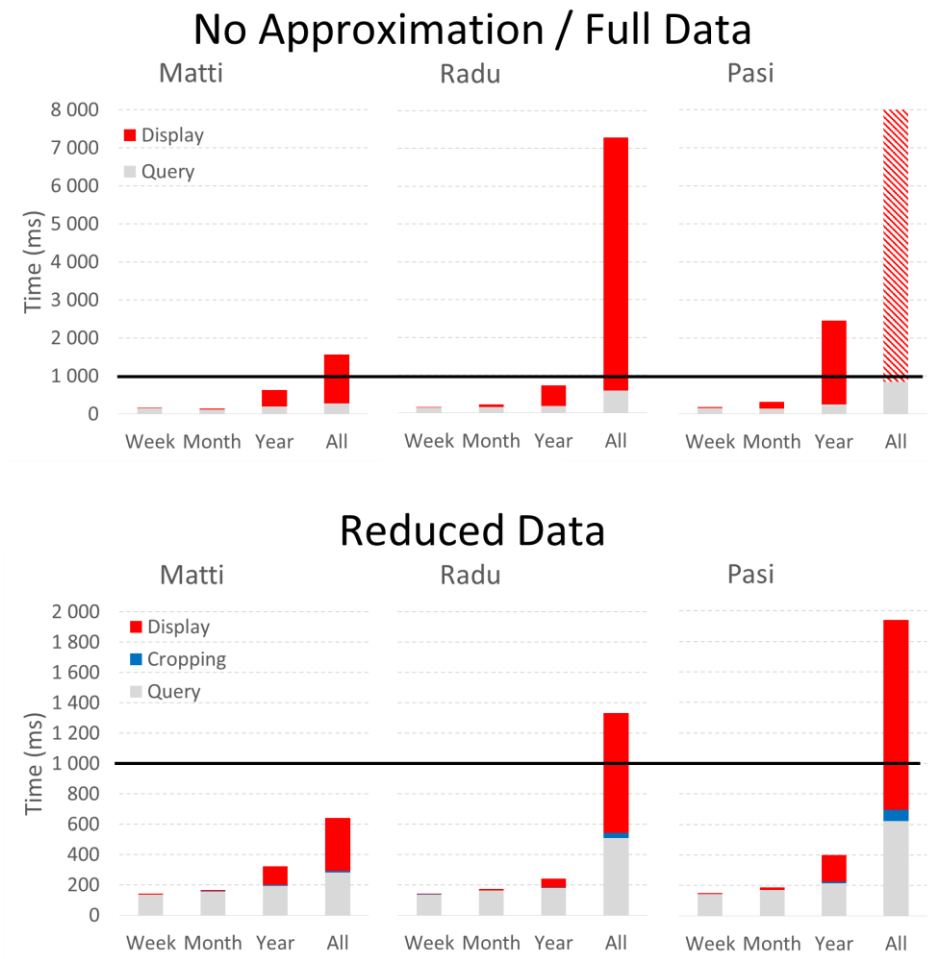


Figure 10. Query and display times for route collections of varying sizes, obtained for users Matti, Radu and Pasi. Pasi's entire collection crashed the browser.

The display time was most affected by reducing the number of points. In fact, the browser crashed when trying to display the entire data for user Pasi if no reduction was applied. After polygonal approximation, the points were reduced as shown in Table 3. We used five different approximations at different map zoom levels. In the experiment illustrated in Figure 10, R1 was used because it allowed all routes to be seen on the map. At this zoom level only ~1% of the points were required to preserve the route shape, making the download time a fraction of that needed for the unreduced data. The query processing time was 6% faster. This is because each route is stored in separate files, and forming a collection requires accessing multiple files. The process can be slow as it requires relocation of the read–write head of the hard-disk. The files are not large, therefore the required time depends mainly on the number of routes rather than the number of points. The cropping process is also performed;

however, the map shows all routes in this experiment, meaning that cropping would not be effective at all.

Table 3. Effectiveness of polygonal approximation.

User	R1	R2	R3	R4	R5
Pasi	0,8%	2%	4%	9%	22%
Radu	0,9	2%	4%	9%	21%
Matti	1,5%	3%	5%	15%	50%

Note: The values are measured as the proportion of points remaining after reducing all routes of a user. Values are shown for five different reduction levels (R1 to R5).

Table 3 shows that the efficiency of the reduction was similar for users Pasi and Radu. However, for Matti the reduction was less effective, especially at the higher zoom levels. Matti uses Android whereas Pasi and Radu use iPhone and Windows Phone respectively. Most Android devices do not represent GPS coordinates with sufficient accuracy, resulting in a zig-zag effect, as illustrated in Figure 12. The Android route therefore requires more points to be represented accurately.

Figure 12. Two walking routes recorded on different sides of a street. The top route was recorded using Windows Phone. The lower route was recorded using an Android device, which uses lower precision to represent coordinates.

The cropping step works in linear time with respect to the number of points in a collection. Using the polygonal approximation first causes the cropping step to process less data (Figure 13).

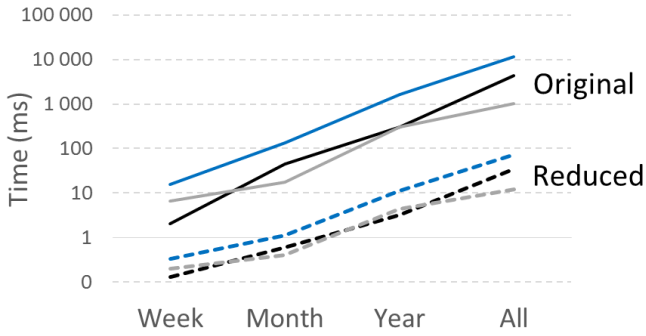


Figure 13. The speed of the cropping process on the original route (solid line) and reduced route (dotted line), for each of the three users: Matti (grey), Radu (black) and Pasi (blue).

After selecting the routes, the user can continue to look around by panning and zooming the map. Only the cropping and displaying operations are performed at this stage. To see how effective the system was in this situation, we performed the following experiment. For each of the three users, we loaded the entire route collection. Then we programmatically panned the map by matching the screen borders to the bounding box of every route. We recorded the number of points and the time required for the processing and the display. This experiment was designed to stress our method, by simulating a user moving the map to see the different regions where he or she expected to find routes.

The results are presented in Table 4, which shows that analysing a data collection requires to load on average, 1% to 5% of the data. Although Matti's collection was far smaller than Radu's, a similar amount of data was retrieved (on average). This is because Radu was recording routes in different cities, whereas Matti's data was mostly obtained in two cities, Kuopio and Tampere. This discrepancy resulted in their data density being roughly the same when zooming at the city level. The cropping time complexity was linear with respect to the total number of points. The time required by Google Map to display the points appears to be linear with respect to the number of points after cropping.

Table 4. Average amount of data and processing times when moving the map.

	Data		Processing Time	
User	**Size (KB)**	**Points**	**Cropping (ms)**	**Display (ms)**
Pasi	1,144	56,827 (2%)	407	210
Radu	325	15,989 (1%)	241	141
Matti	300	14,901 (5%)	93	128

To give more meaning to the amount of data being transferred, we compared the amount of data loaded by Google Maps to show the map tiles. Tiles were portable network graphic (PNG) images, typically 256 x 256 pixels, and their size varied considerably depending on the amount of information displayed. We recorded the size of each tile loaded in the panning experiment described earlier and the average was 10.5 KB per tile. A screen of 1920 x 1200 can load 36 tiles, equivalent to about 378 KB. This value is comparable to the amount of route data loaded when panning the map to browse Radu's and Matti's collections. To load Pasi's route data meant loading roughly three times the amount of data contained in the map tiles.

Using reduction and cropping not only improved the speed but also prevented the browser from crashing. For example, loading Pasi's entire collection without applying any reduction or cropping caused the browser to crash. Online utilities, such as GPSVisualizer[5], GmapGIS[6] and many sports tracking programs – which also use Google Maps (see Table 1) – are incapable of displaying these data, as they lack a similar data-reduction strategy.

[5] http://www.gpsvisualizer.com
[6] http://www.gmapgis.com

3 GRID-BASED OPERATIONS

Using a grid, we defined four route operations that were useful for solving different problems. These operations were:

- Similarity
- Inclusion
- Novelty
- Noteworthiness.

Similarity is probably the most common operation performed on routes. For instance, Ying et al. [2010] demonstrated that meaningful friend recommendations can be issued in social networks by analysing users' similar routes. Another case where route similarity is helpful is when giving trip recommendations. In Shang et al. [2012] a route is recommended when a set of intended places and textual attributes that describe the user's preferences is given as input. The similarity measure has also been used successfully to identify ideal places to build new bicycle paths [Evans et al. 2013]. Route similarity is used as an inverse distance function for clustering applications [Pelekis et al. 2010, McCullough et al. 2011, Ying et al. 2009] in various applications – for instance, to identify traffic congestion.

Finding similar route(s), also known as "k nearest neighbour search" in a database, is the most typical use for the similarity operation [Agrawal et al. 1993, Frentzos et al. 2007, Ni and Ravishankar 2007, Wang and Liu 2012, Yanagisawa et al. 2003]. In Mopsi, this feature enables users to find a similar route recorded in the past in order to compare the routes in terms of speed. The feature also allows comparison with the data of other users who have recorded similar routes.

Many measures for computing route similarity exist:

- *longest common subsequence* (LCSS) [Vlachos et al. 2002]
- *edit distance on real sequence* (EDR) [Chen et al. 2005]
- *dynamic time warping* (DTW) [Zheng and Zhou 2011]
- *edit distance with real penalty* (ERP) [Chen and Ng 2004]
- *Hausdorff distance* [Rockafellar and Wets 2009, Chen et al. 2011]
- *Frechet distance* [Eiter and Mannila 1994].

These measures typically require quadratic time to be computed. Some approximate and more complicated variants exist, such as *FastDTW* [Salvador and Chan 2004]. *Euclidean distance* (*L2-norm*) [Gradshteyn and Ryzhik 2000] is an example of a simple linear time approach to compute route similarity; however, the method works well

only if routes are aligned at their start and are of similar length. This degree of congruence happens rarely in a real database.

We considered two routes to be similar if they overlapped. The amount of overlap measured how similar the routes were. We defined a fast and linear time similarity measure (C-SIM), which focuses on the spatial aspect of the routes. C-SIM is inspired by the Jaccard set similarity coefficient, and measures the amount two routes have in common divided by the total space covered by the two routes. C-SIM is fast enough to allow real-time route similarity searches in large databases. The equation is:

$$S(C_A, C_B) = \frac{|C_A \cap C_B| + |C_A \cap C_B{}^d| + |C_B \cap C_A{}^d|}{|C_A| + |C_B| - |C_A \cap C_B|}, \quad (1)$$

where C_A and C_B are the cell representations of two routes. $C_A{}^d$ and $C_B{}^d$ are the dilated regions of the two routes respectively.

The second operation is *Inclusion*. It measures how much one route is contained inside the other. The equation is:

$$I(C_A, C_B) = \frac{|C_A \cap C_B| + |C_A \cap C_B{}^d|}{|C_A|}, \quad (2)$$

where C_A and C_B are the cell representations of two routes. $C_B{}^d$ is the dilated region of the second route.

Unlike similarity, inclusion is not symmetric. The measure is useful for solving drive-sharing problems, by identifying users who –

- can pick up somebody along the user's route, or
- can be picked up by somebody else on their route.

In Mopsi, inclusion is used to search for routes that pass through a region manually specified by the user on the map **[IV]**.

Novelty measures the amount of unique parts of a route compared with other routes in a database. Novelty can be useful in several applications. For instance, it may be considered an alternative to iBAT [Zhang et al. 2011] with regard to identifying taxi fraud, namely when a taxi driver takes a longer route than necessary to arrive at the destination. The given route is compared with other past routes that start and end at the same locations. If the new route has a high novelty measure, the route is labelled as fraudulent. Alternatively taxi driver safety can be addressed as in [Karimi and

Lockhart 1993]. Another application for novelty is to automatically update GPS navigation systems that exist in many cars nowadays. If a recent route shows novelty compared with the existing road network, the roads in the region have changed; in such instances, the database updating methods described by Fathi and Krumm [2010] and Cao and Krumm [2009] should be applied [**V**].

Noteworthiness is closely related to novelty. It measures the amount of rarely visited parts instead of focusing only on unique parts. This measure is useful in places that have a high density of routes that have extremely few novel regions. In Mopsi, novelty and noteworthiness are used to inform users when their route passes through places they have never visited before. It is also verified whether other users have frequented the area (Figure 14).

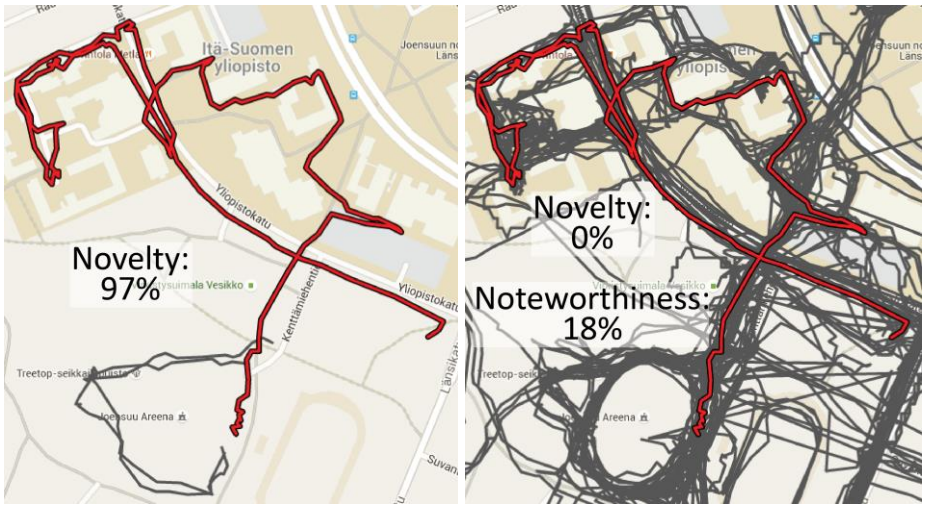


Figure 14. A route is 97% novel to the user (left panel). The same route is not novel at all with respect to all users (right panel), but 18% of the route is noteworthy. The selected route is shown in red and other routes in the collection are grey.

3.1 GRID

Grids have been used to represent geographical data in past studies. In Pang et al. [2012] and [Zhang et al. [2011], grids were used to find patterns in taxi data. In Wei et al. [2012], popular routes were constructed using the frequency information of grid cells. In Zheng et al. [2010] and Bao et al. [2012] the grid was used to infer stay areas, which in turn are used to detect points of interest. In Krumm and Horvitz [2006] grids were shown to be useful when predicting the destination of moving vehicles.

The abovementioned examples used grids to perform frequency analysis in sub-regions of a given area. We extended the use of grids to define a similarity measure

between routes and to perform similarity-based retrieval in route databases. To enable this, we required a grid with equal cell size spanning the entire planet, which is not trivial to do [Kennedy and Kopp 2001]. The existing applications create grids by segmenting the latitude and longitude values [Bao et al. 2012] for which the cells gradually change size when one moves in the north–south direction. Another way grids have been defined is by focusing only on a small region, such as a city – as in Zhang et al. [2011], Krumm and Horvitz [2006], Pang et al. [2012] and Wei et al. [2012] – and dividing that region into equal-sized cells. When computing the similarity of routes, the grid needs to be finer than for other applications, which typically use cell sizes in the scale of 100 m to 1 km.

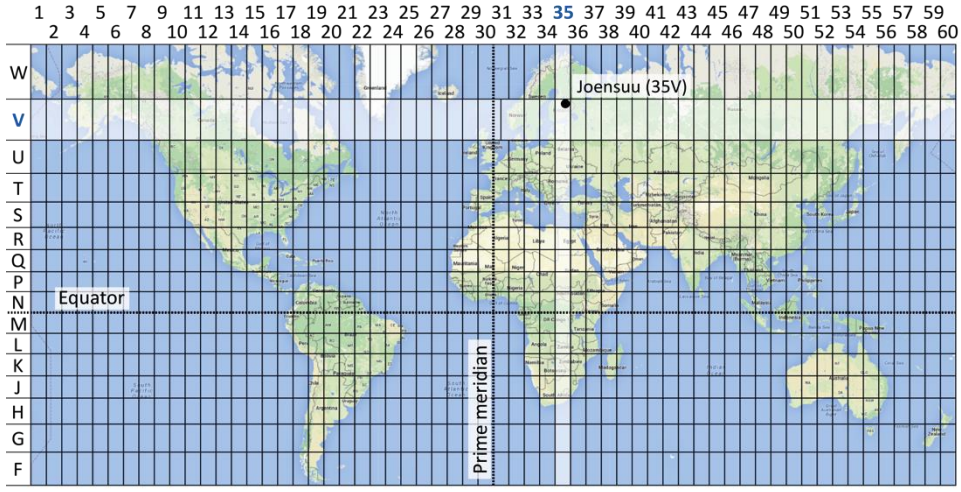


Figure 15. MGRS grid zones. Joensuu is in UTM zone 35 and latitude band V.

To generate the grid, we used the military grid reference system (MGRS[7]), in which cells of equal size fill up specially defined zones that cover the entire planet. These zones do not usually follow the north-south orientation. This aspect allows the zones to wrap around the planet. Then, each zone is divided into cells of the same size in square metres.

MGRS is an alpha–numeric two-dimensional coordinate system in which locations are identified independent of their elevation. MGRS divides Earth into projection zones and computes easting and northing in metres, within a designated zone. The Universal Transverse Mercator (UTM) is used to divide the planet into 60 zones, each being 6° of longitude wide. For the polar regions (above 84°N and below 80°S), the Universal Polar Stereographic (UPS) convention is used instead of UTM. For the perpendicular segmentation, bands of latitude (8° high) are used.

[7] http://builds.worldwind.arc.nasa.gov/worldwind-releases/1.4/docs/api/overview-summary.html

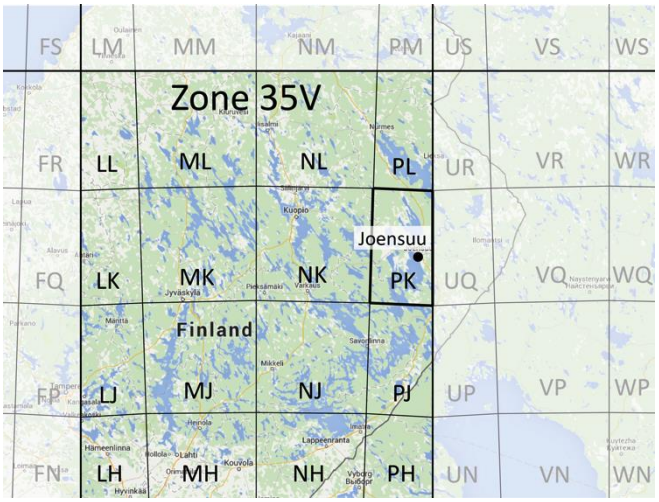


Figure 16. 100 kilometre squares. Joensuu is in square PK of zone 35V.

The first three characters of the MGRS value for the city of Joensuu, Finland, are 35V (Figure 15). The next pair of characters identifies a 100 km × 100 km square within each of the grid zones. Joensuu is located in region 35VPK (Figure 16). The remaining part consists of numeric easting and northing values within the 100-km square. MGRS allows one of five predefined precision levels when choosing the cell length: 1 m, 10 m, 100 m, 1 km or 10 km. However, any specific degree of precision can easily be obtained if the desired cell length can be perfectly divided into 100,000 (metres). Limited by the average GPS error, we chose a 25 m × 25 m cell size. As shown in Figure 17, we identified the centre of a small park as being 35VPK16461774.

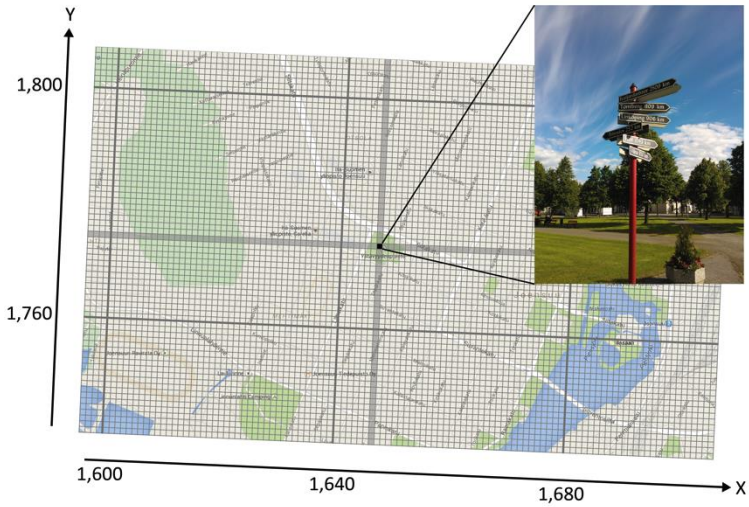


Figure 17. 25 m x 25 m cells in the Ystävyydenpuisto (Freedom Park) in Joensuu.

3.2 EVALUATION

We evaluated our proposed grid-based similarity measure, C-SIM, to see how it compares with other approaches. We used the Mopsi2014[8] dataset, which is a subset of all routes in the Mopsi database collected between 2008 and 2014. It contains 6,779 routes recorded by 51 users having 10 or more routes (Table 5). Some users have more data than others (Figure 18).

Table 5. Mopsi2014 dataset summary.

Routes	Points	Kilometres	Hours
6,779	7,850,387	87,851	4,504

The dataset consists of routes with a wide range of activities, including walking, cycling, hiking, jogging, orienteering, skiing, driving, and travelling by bus, train, or boat. Even though such ground truth is not available, using the method of Waga et al. [2012] we automatically computed the movement types and showed a distribution of these activities by transportation mode (Figure 19). Routes exist on every continent except Antarctica allowing a good test for MGRS, which seems to work well in all regions where test data is available. Most routes are in Finland, in the city of Joensuu, which creates a very dense area suitable for extensive testing of the methods.

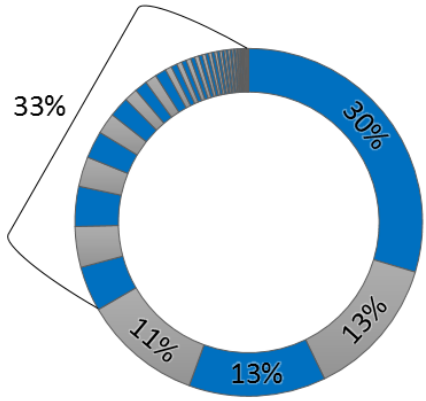


Figure 18. The distribution of the data per user. The four most active users had recorded approximately two thirds of the data.

We computed a 25 m × 25 m cell representation for all 6,779 routes using MGRS. The cell size was decided based on experimentation and by observing typical GPS inaccuracies. The cell database entries included cells obtained from interpolation and dilation [**II**], which are required for the operations. Statistics are shown in Table 6. Typically, point databases are indexed by using tree structures such as R-tree [Guttman

[8] http://cs.uef.fi/mopsi/routes/dataset

1948] to make range queries possible. As a comparison, if R-tree is applied, Mopsi2014 would require approximately 1 GB of space. Roughly the same space is required by the cell database when indexed using B-tree [Cormen 2009]. In **[III]** we showed that the Hash index [Cormen 2009] can also be used and is about 50% faster than the B-tree index, with a 40% increase in memory requirements.

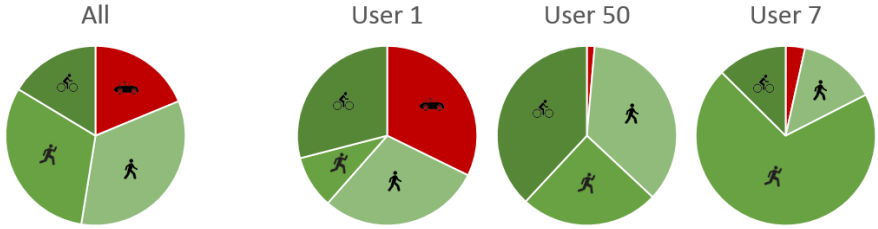


Figure 19. The distribution of all walking, running, cycling and car routes in Mopsi2014 dataset. The distributions for three sample users are also shown.

Table 6. Space requirements for Mopsi2014[9] dataset.

	Entries	Index	Total
Point Database	7,850,387 (329 MB)	R-tree (650 MB)	979 MB
Cell Database	11,477,506 (525 MB)	B-tree (429 MB)	954 MB
		Hash (788 MB)	1,313 MB

We investigated how various route similarity measures are affected by the following transformations:

- increasing sampling rate (adding points)
- decreasing sampling rate (removing points)
- adding noise
- random shifting of points
- synchronized shifting of points.

We extended the evaluation performed by Want et al. [2013] by adding C-SIM and a few other similarity measures to the comparison. We selected 1,000 random routes from Mopsi2014, and analysed the behaviour of the similarity measures when the five artificial transformations were applied. We assumed that these transformations might occur naturally in a route database due to the use of different devices, varying GPS accuracy and other influences. Therefore, the similarity between the transformed route and the original was expected to be high (100%); alternatively, the distance should be 0 for distance-based measures. The trends for the similarity measures are illustrated in Figure 20.

[9] http://cs.uef.fi/mopsi/routes/dataset

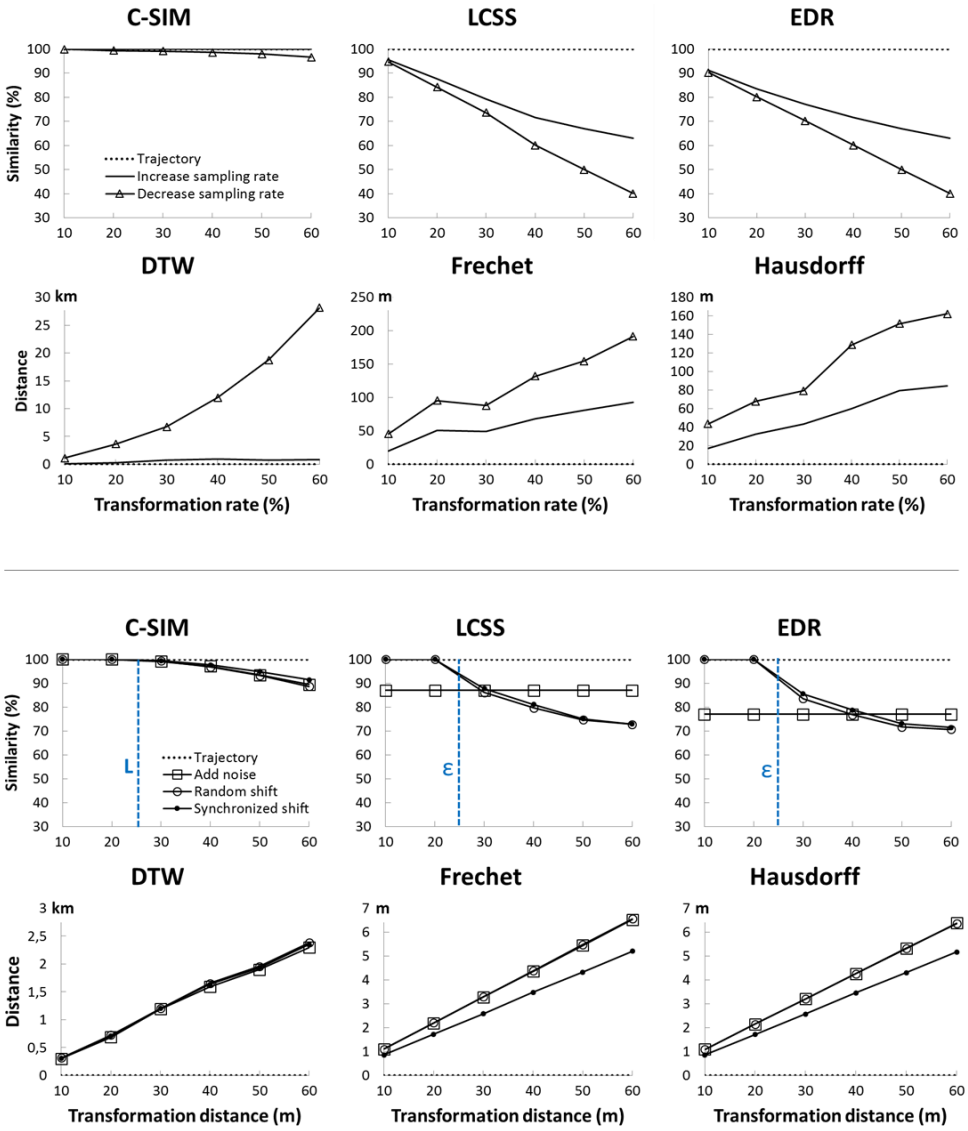


Figure 20. Six route similarity measures affected by sampling rate transformations (upper panel) and by noise and point shifting (lower panel).

C-SIM performed well when points were added or removed. The measure is not affected by increasing the sampling rate, because the cell representation remains identical due to the interpolation step. Decreasing the sampling rate had a minor effect on similarity, because of the inability of interpolation to correctly predict the missing parts of the route. However, the effect was far smaller than that of the other methods.

Among the measures, LCSS and EDR were the most sensitive to a decreased sampling rate, although an increase in sampling rate had a milder effect. The C-SIM, LCSS and EDR measures are not affected by point shifting if the transformation distance is small (L = ε = 25 m, in our experiments). For higher distances, C-SIM decreases in proportion to the transformation distance. LCSS and EDR similarities do not decrease proportionally to the distance; ε is a threshold when two points are considered identical. The similarity is higher when transformation distance is small, but will be above ε because points shifted only little more than ε metres away are still likely to match with other points in the vicinity. DTW did not vary with an increase of the sampling rate but was highly sensitive to a decrease. Hausdorff and Frechet were both sensitive to changes in sampling rate.

Table 7. Summary of the effectiveness of the 6 route similarity measures.

Function	Sampling rate		Add noise	Point shifting	
	Increase	Decrease		Random	Sync.
C-SIM	Robust	Robust	Fair	Fair	Fair
LCSS	Sensitive	Fair	Sensitive	Fair	Fair
EDR	Sensitive	Fair	Sensitive	Fair	Fair
DTW	Robust	Sensitive	Sensitive	Sensitive	Sensitive
Hausdorf	Sensitive	Sensitive	Sensitive	Sensitive	Fair
Frechet	Sensitive	Sensitive	Sensitive	Sensitive	Fair

Noise affected LCSS and EDR more than the other measures because it caused a change in the length of a transformed trajectory. DTW was sensitive to all transformations. Frechet and Hausdorff were sensitive to noise and point shifting, but less so if the points were shifted in the same direction (synchronized). The similarity depends linearly on the transformation distance. The results are summarized in Table 7.

4 ROUTE SEARCH METHODS

Routes can be searched for various reasons, such as finding, comparing and reviewing:

- find a past route in order to compare any progress
- compare one's effort with that of others on a similar track
- review statistics of a route recorded in the past
- memorize a specific tour to make revisiting a place easy

In large collections, finding a specific route is difficult. Traditionally, sports tracking applications offer a time-ordered listing and/or map plotting of the collection. Recently, thumbnail listing and segment-based searches have also become supported by certain applications. We introduced two novel means of searching for routes: similarity search **[III]** and gesture search **[II]**. These approaches are discussed in greater detail here.

4.1 TIME-ORDERED SEARCH

All sports tracking applications offer some kind of time-based ordering of a route collection. The options to display the information are a calendar, a list and – more recently – a list with route thumbnails (see Figure 21 and Table 8).

The *calendar* is familiar and intuitive to many users; however, it can show numerous empty locations, meaning the user must perform many clicks to access the data. In addition, calendars impose a limit on the number of activities per day. The calendar is large and wide and it cannot coexist with a map on a typical screen.

The *list* is more useful because it contains no blanks. In Mopsi, list items are grouped by the date. The duration, movement type and distance are shown. Other applications often include additional information, such as calorie burning and power output. In Mopsi the user lacks access to this information, which typically requires separate hardware in addition to the mobile phone.

Both the calendar and list formats have a weakness when searching for routes. They do not show the shape of the route although shape is a feature that users easily recognize. For this reason, all major applications now show a thumbnail list, which provides a greater amount of information but with the drawback that the list becomes longer. If the date is unknown, these methods are no longer useful and would imply sequential searching through every item in the list.

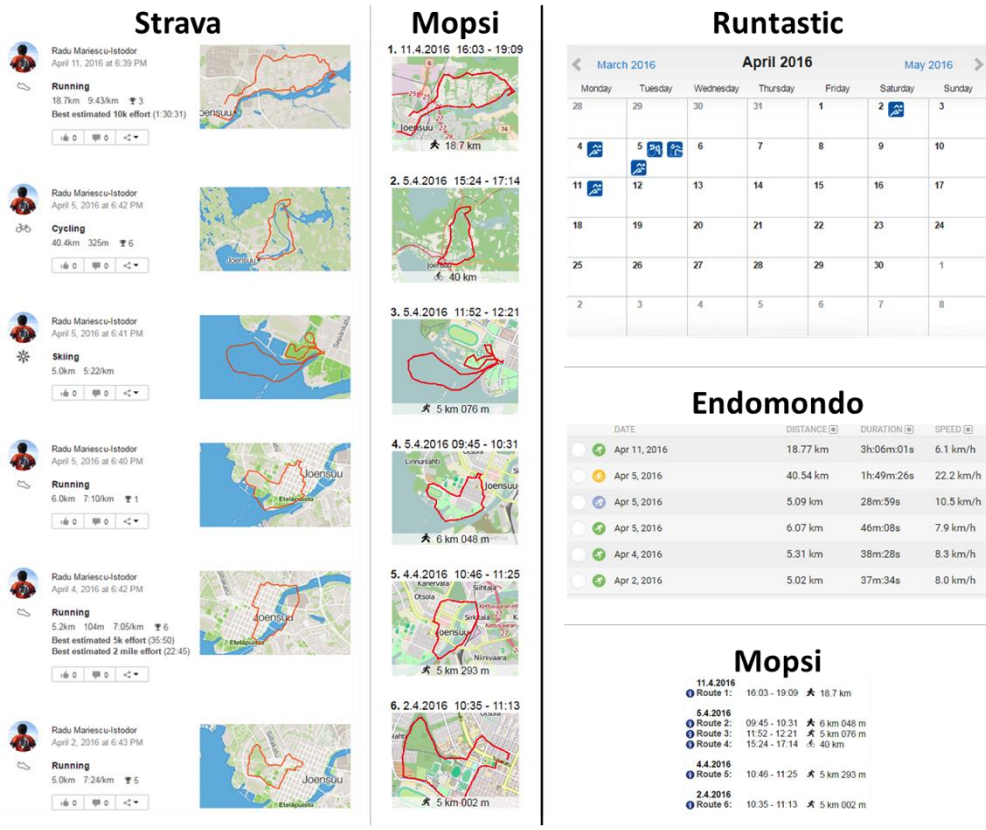


Figure 21. The same route collection visualized in different ways. Strava and Mopsi demonstrate the thumbnail view (left panel), Runtastic shows the calendar view (top right panel), and Endomondo and Mopsi show the list view (lower right panels).

Table 8. Time-based route visualization methods and their availability in sports tracking software.

	Mopsi	Endomondo	Runtastic	Sports Tracker	Strava
List	✓	✓	✓	✓	
Calendar		✓	✓	✓	✓
Thumbnails	✓	✓	✓	✓	✓

4.2 MAP-BASED SEARCH

Some applications show the collection on the map. In this way, routes can be identified by their location (Figure 22). The Sports Tracker application represents routes by their starting points so that the map is not overwhelmed by too many points. The route representatives are coloured with respect to their transportation mode. The disadvantage of this method is that it hides much of the information. Also, typically routes start at the same location – the user's home – for activities such as cycling and running. This commonality makes the three running routes hard to distinguish. In Mopsi, the entire route shapes are shown. The amount of data is minimized using the reduction. Problems occur if routes overlap so much that they become indistinguishable.

The benefit is that routes can be identified quickly, unless there is a massive amount of data for the region. In the latter scenario, the data should first be limited based on time.

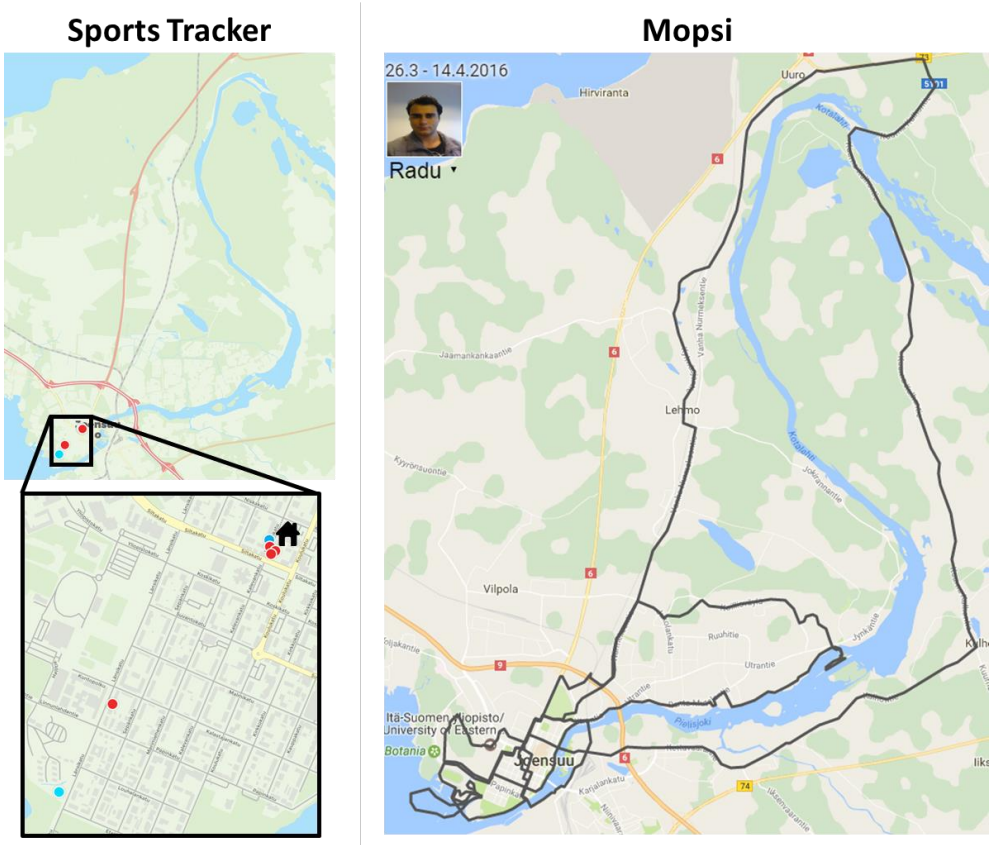


Figure 22. The same route collection displayed on a map by Sports Tracker (left panel) and Mopsi (right panel).

Strava does not display route collections on a map, but it does display a collection of user-defined segments of interest (Figure 23). This application provides an easy way for people to compete with others. The segments are manually defined by users via their start and end points and by the intermediate locations which can be selected from the user's routes. Once a segment is defined, users passing through that area will be clocked and ranked in a list, providing another way to search for routes. Because segments are manually defined, some regions may lack them and it is impossible for users to conduct a search in such areas.

Figure 23. Strava segments in Joensuu (left panel) and the first 5 athletes on segment H (right panel).

4.3 SIMILARITY SEARCH

Route similarity can be used as a method to search the database (Figure 24). Starting with a reference route, Mopsi allows users to perform *route similarity ranking* (RSR). The application shows a list of routes that are spatially similar to the reference route, with results ranked in decreasing order of similarity. For each route, the ranking shows the user who recorded that route, the transportation mode used, the similarity value and the date. Figure 24 shows only the first 16 elements of the ranking whereas the full list contained 1196 routes.

The user can compare the reference route with any similar route in the list. The analysis can also be localized to a chosen segment of the reference route.

1	Radu	🚲	100%
2	Radu	🚲	99%
3	Radu	🚲	96%
4	Radu	🚲	96%
5	Radu	🚲	96%
6	Radu	🚲	96%
7	Radu	🚲	96%
8	Radu	🚲	96%
9	Radu	🚲	96%
10	Radu	🚲	95%
11	Radu	🚲	95%
12	Radu	🚲	95%
13	matti	🚲	95%
14	Radu	🚲	94%
15	Radu	🚲	92%
16	Radu	🚲	92%
17	Radu	🚲	90%
18	Radu	🚗	90%
19	Radu	🚲	89%
20	Radu	🚲	84%
21	Radu	🚲	84%
22	Radu	🚲	81%
23	Radu	🚲	77%
24	Andrei	🚲	77%
25	Radu	🚲	77%
26	Radu	🚲	76%

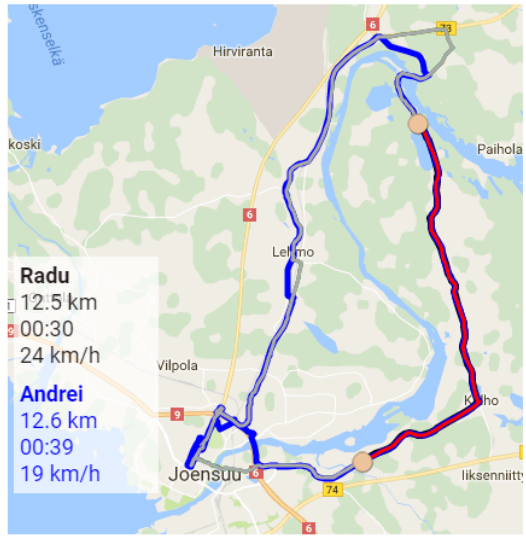


Figure 24. User Radu's reference route (grey) and a list of highly similar routes from the database. The user name, inferred movement type and similarity values are shown for each similar route. A route of user Andrei (blue) was 77% similar. A manually selected common segment (red) was selected and analysed, and Radu's segment was shown to be 5 km/h faster. The reason for the large speed difference was a strong tailwind from the north.

4.4 GESTURE SEARCH

Gestures have been used as a means to access menu items without the need to traverse large hierarchies [Kristensson and Zhai 2007, Li 2010] by providing users with various types of shortcuts. We proposed using gestures to access routes in large collections. The gesture represents hand-drawn input in the form of a free shape drawn on a map; the shape approximates the locations through which the targeted route passes. According to Cirelli and Nakamura [2014] and Karam and Schraefel [2015], this gesture is classified as symbolic and triggers a command, namely the search for spatially similar routes.

Typically, gesture-based systems require the user to learn a set of symbols [Cirelli and Nakamura 2014]. In our study, the user was expected to remember the spatial characteristics of the route and to be able to read maps, because roads, buildings and terrain elements (such as forests, lakes, and rivers) provide key information when giving the input. For example, a user can draw the input by following a river front, road, or other landmark visible on the map. Users who have a large route collection benefit most from the gesture search. It is therefore fair to assume that these users also have the necessary skills to understand maps.

Gesture search has two modes: similarity (Figure 25) and inclusion (Figure 26), which use the two operations respectively. To enter the gesture input mode, the user presses a hotkey (Ctrl for similarity, Shift for inclusion). While the key is pressed, the map changes colour to show which mode is active and further acts as a canvas on which to draw. The drawing is completed when the hotkey is released and search is then initiated with the drawn shape being used as the input.

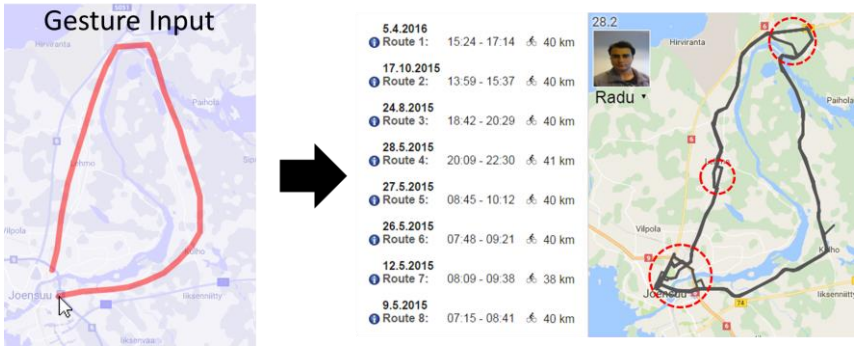


Figure 25. Gesture search using Similarity. Eight routes resembling the drawn shape were found and returned to the user. The eight routes overlapped perfectly on the map, except in three highlighted regions where the road network allowed variation.

The similarity search retrieves route candidates that are similar to the drawn shape, whereas the inclusion search retrieves candidates that contain or include the drawn shape. The latter is similar to the segments feature of Strava in the sense that routes passing through the drawn segment are retrieved. The benefit is that segments do not need to be created and stored in the system. Users can draw any segment at any time. The downside is that users do not become aware of places in which other people compete, as they do in Strava.

Figure 26. Gesture search using Inclusion. Five routes that pass through the drawn region are found and presented to the user.

The precision of drawing the gesture should be independent of the zoom level of the map. When the zoom level is decreased by one unit the content of the map becomes

half of its previous size, and consequently the regions on the map become twice as difficult to read. We created 10 grids with different resolutions and stored the routes at each of these approximation levels (Table 9).

Table 9. A mapping from zoom level to the grid resolution. The statistics are for Mopsi2014 Route dataset using each of the grid resolutions.

Zoom level	≤ 6	7	8	9	10	11	12	13	14	≥15
Grid resolution	0	1	2	3	4	5	6	7	8	9
Cell size (km)	12.8	6.4	3.2	1.6	0.8	0.4	0.2	0.1	50 m	25 m
Number of cells	7×10^4	9×10^4	1×10^5	2×10^5	4×10^5	7×10^5	1×10^6	3×10^6	5×10^6	1×10^7
Memory (MB)	3.5	4.5	6.5	9.5	16.5	30.6	59.6	118.6	238	486
B-tree Index (MB)	8.5	9.5	13.5	21.5	35.6	66.7	131.8	263.1	526	1.1 GB

The finest grid has a cell size of 25 m × 25 m. Finer grids are not needed because at this level, GPS error becomes apparent and the route approximations become unreliable. The number of cells needed increases exponentially when finer grids are produced. Therefore, we did not compute unnecessary levels for no purpose. The sparsest grid had a cell length of 12.5 km. At lower levels (≥ 25 km) the cell size becomes so large that even the longest routes are represented by only a few cells.

4.3 EVALUATION

We studied the efficiency of the gesture search from a usability point of view. We compared the average time a user spends on searching a randomly chosen route using the gesture search versus using the traditional system. Eleven volunteers were asked to search randomly selected routes using a tool[10] built for this purpose, as follows:

1. A target route was shown on the map but no date, length or duration were shown. The user could study and memorize the route for as long as he or she wanted to.
2. When the user pressed the *Start* button, he or she was (randomly) directed either to the traditional system or to the new gesture search. The timer was started.
3. The task was to find the route and input its date and then press the *Stop* button. If the date was correct the timer was stopped. If the user considered the task too difficult, he or she could press the *Give-up* button.

[10] http://cs.uef.fi/mopsi/routes/gestureSearch/qual.php

Each volunteer was asked to repeat the test at least 10 times or for as long as he or she found the task enjoyable.

In total, 82 routes were found using the traditional system, and 89 routes using the gesture search. Traditional searches were given up on 50% more often than the gesture search, with 24 traditional searches being abandoned, compared with only 16 gesture searches. Gesture search was 41% faster, on average. The individual performance differences are shown in Figure 27. Traditional searches were slower on average than gesture searches for all users except one.

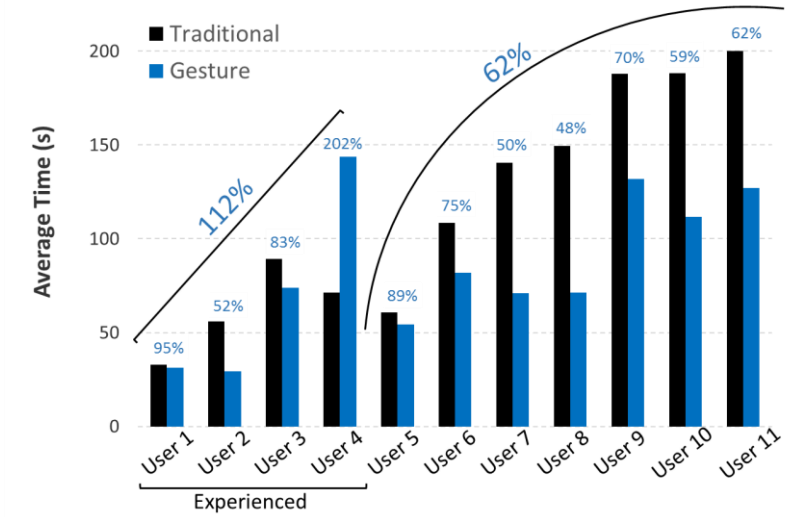


Figure 27. Average search times, showing the superiority of the gesture search relative to the traditional search. Results are shown for every user who participated in the experiment.

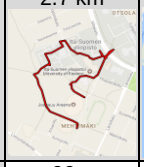
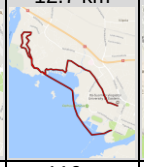
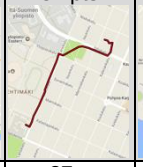
The search time was also affected by factors such as complexity and length of the route, and density of the areas the route passes through. We next grouped the results by these three factors. Complexity was calculated as the number of points used by the polygonal approximation [Chen et al. 2012] to represent the route at the maximum zoom level at which the route could still be seen in its entirety. Density was calculated as the proportion of cells that were frequented by many other routes; density values are the converse of the noteworthiness value in **[III]**. The results, shown in Table 10, indicated that although shorter, less complex routes in low-density areas were faster to find, the gesture search outperformed the traditional approach in all cases.

The volunteers were asked if they liked the gesture search and which method they would prefer to use for similar search tasks. Ten volunteers rated the gesture search as good and one as excellent. Most (nine volunteers) preferred the gesture search,

none preferred the traditional search, and two people said they would not use either method. Written comments included “I really liked it” and “It was fun”.

The four volunteers whose data are shown on the left-hand bars in Figure 27 had previously been familiar with the traditional search method. Even in that group, the gesture search yielded faster results for 75% of cases. This result was above our expectations because we assumed that previous experience in using the traditional method would bias the results. Less experienced users seem to find the routes faster using the gesture search than the traditional search. This result indicates that the gesture search is a more intuitive method.

Table 10. Average search times, grouped by various factors.

	Length		**Complexity**		**Density**	
	Short 2.7 km	Long 12.7 km	Low 31 pts	High 128 pts	Low 12 %	High 75 %
						
Traditional	90 s	116 s	87 s	120 s	90 s	117 s
Gesture	64 s	78 s	65 s	77 s	54 s	88 s
Reduction	30%	33%	25%	36%	30%	24%

5 INFERRING ROAD NETWORKS

In large cities, navigation using traditional means – a paper map – has become almost impossible. Road networks are becoming increasingly complex and large roads rarely offer the chance to pause and study the situation if one gets lost. As a result, navigators such as TomTom[11] and Garmin[12] are present in most cars nowadays and most smartphones have navigation capabilities. Road networks may also be used to offer personalized navigation such as safe routing [Krumm and Horvitz 2017] or accessible routing [Kasemsuppakorn and Karimi 2009]. For such applications, up-to-date and accurate information is essential.

The current acquisition and updating of road networks is characterized by a large amount of manual work, which is costly and slow. There have been two main ways of automatizing the process: aerial image processing [Tavakoli and Rosenfeld 1982,

[11] https://www.tomtom.com
[12] http://www.garmin.com

Hu et al. 2007, Barsi and Heipke 2003] and GPS route processing [Edelkamp and Schrödl 2003, Davies et al. 2006, Cao and Krumm 2009].

Using aerial images has limitations because roads have varying features such as colour, intensity, shadows and variable widths (Figure 28). Buildings cause further difficulties and this issue was addressed by Tavakoli and Rosenfeld [1982]. In that study, categorization was performed using edge features to separate roads from buildings and other structures. The method described by Hu et al. [2007] for finding roads begins with several initial guesses. A road tree is then built for each initial guess by tracking along road segments in one or more directions. By merging the resulting trees, a road network is created. Obtaining the direction of travel for the roads is not possible using image data.

Figure 28. Aerial images showing part of a city (left) and a countryside area (right).

GPS routes are easier and less costly to collect than aerial images. Route databases collected for various purposes, such as the OSM traces[13] and Mopsi2014[14], are already available and growing. The routes have fewer artefacts than the aerial images and the only issue is the error caused by tall buildings and other obstructions. Routes have the added benefit of preserving the direction of travel and can be used to produce a directed graph. Because of these advantages, inferring a road network from GPS routes has become an attractive area of research, and several conceptually distinct approaches have emerged. In addition to road networks, other types of networks, such as pedestrian networks [Kasemsuppakorn and Karimi 2013] which can be inferred from walking routes.

[13] https://www.openstreetmap.org/traces
[14] http://cs.uef.fi/mopsi/routes/dataset

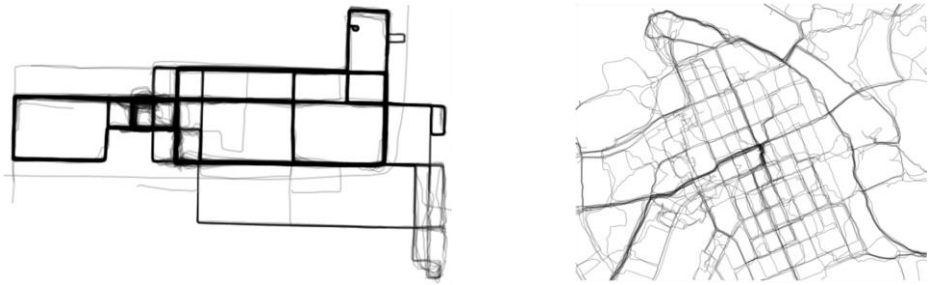


Figure 29. GPS routes recorded in Chicago (left) and Joensuu (right).

Visual methods [Chen and Cheng 2008, Davies et al. 2006] use route data to form binary images, which are processed using image-processing techniques such as contour finding, morphological operations, skeletonization and density-based thresholding.

Route merging methods [Niehoefer et al. 2009, Cao and Krumm 2009] combine routes one by one to form a graph. If a route segment is already part of the graph, a weight corresponding to that particular segment is increased. Finally, segments with too low weights are removed from the network. Merging methods typically filter GPS data in order to better handle the noise.

Clustering-based methods also exist to infer road networks [Edelkamp and Schrödl 2003, Schrödl et al. 2004]. This approach typically begins by considering only the points of the GPS routes; connectivity is omitted. Then, equally spaced representatives are placed over the point data. The representatives are optimized using *k*-means, and finally the network is formed using the point connections from individual routes.

Some studies have focused on the task of locating the road intersections [Barsi and Heipke 2003, Fathi and Krumm 2010], and machine learning is used to achieve this goal. A classifier is trained using positive and negative samples obtained from data containing ground truth, typically OSM.

The visual, merging and clustering methods perform poorly in places where GPS accuracy is low. In those regions, numerous intersections are incorrectly found and many spurious segments disrupt the quality of the network. The filtering process employed by the merging methods is insufficient to handle abundant GPS error. Visual methods can handle the problem through setting a higher value for the density threshold parameter. The downside is that regions of the map having a low number of routes will also be omitted from the process. The existing clustering methods do not attempt to solve GPS errors at all.

We argue that accurately obtaining the location of road intersections is crucial for generating high-quality maps. Therefore, in **[V]** we proposed a new method entitled CellNet, which works in two steps:

1. finding the road intersections
2. generating the in-between road segments.

CellNet has two parameters: L and R, which can be interpreted respectively as the expected average GPS error in the dataset (L) and the minimum distance between two intersections (R). The method does not lead to substantial differences when these parameters are altered, and we expected it to work well with our recommended values of 25 and 80. A visual representation of the method output using these values is shown in Figure 30.

Figure 30. Joensuu road network as inferred by CellNet.

Unlike other intersection finding methods [Barsi and Heipke 2003, Fathi and Krumm 2010], CellNet does not require training data. It finds the intersections using a *split descriptor*, which checks to see whether at a certain location, routes head into more than two general directions. To check this a set of data points is created, as described in Figure 30. Then, clustering is performed separately with two and three clusters, using the random swap algorithm [Fränti and Kivijärvi 2000]. The two clusterings are inspected using the silhouette coefficient [Rousseeuw and Kaufman 1990] to deduce the correct number of clusters. If three clusters provide the best solution, a split is concluded.

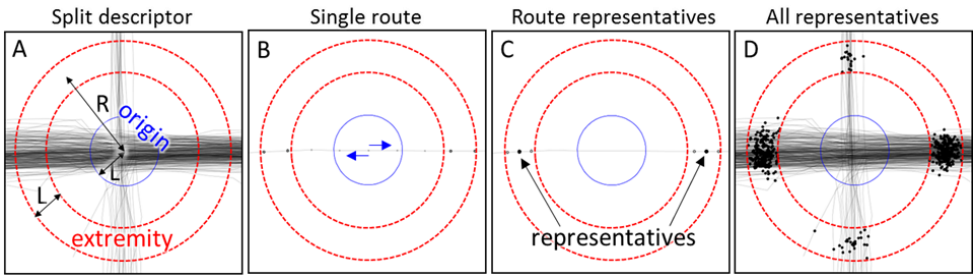


Figure 31. A, the split descriptor composed of the origin and the extremity. B, a sample route traversing through the point of interest; points inside the extremity are chosen. C, the selected

points are averaged in each of the two directions to create the representatives. D, representatives of all routes passing through the point of interest.

Once the intersections are found, the in-between road segments are selected by checking every subsequent pair of intersections that every route passes through. If more routes link the same intersections, all segments are kept and are used in the following optimization step. We used the method in Hautamäki et al. [2008] to obtain a representative for all segments between every pair of intersections (Figure 32). We excluded segments that were not 100% spatially similar according to C-SIM similarity measure **[III]**. Unlike Hautamäki et al. [2008], we did not initialize the optimization method using the medoid. Instead, we used the shortest segment under the assumption that it has less GPS error, which would make it a good initial guess. In addition, we used the FastDTW algorithm [Salvador and Chan 2004]. Our results were no worse than those obtained when the medoid was used and the optimum DTW was calculated but the speed had improved by 99%.

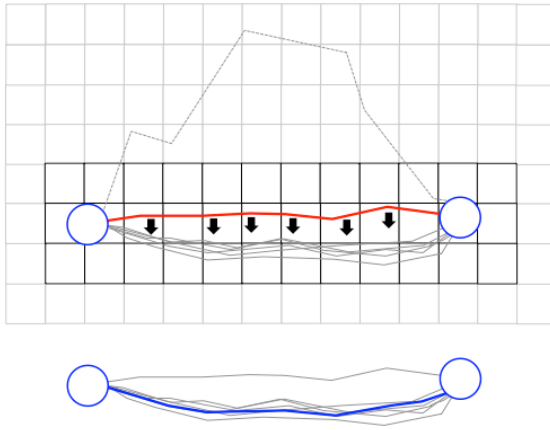


Figure 32. The minimum length segment (red) was improved by the averaging method using other segments that were spatially similar. The dashed-line segment (top) was not spatially similar and was therefore excluded from the process. The result was the fine-tuned blue segment (bottom).

Once the links were optimized, we noted that some became redundant. This was the result of a route missing one or more intersections due to GPS error. We removed these links using the following strategy. For every link segment, we found all segments that were contained inside its spatial region and marked them as valid. To do this, we used the inclusion measure from **[III]**. If a valid path existed between the two intersections, the direct link was removed because it was probably redundant. This strategy is an improvement over the one presented in Fathi and Krumm [2010], which takes into consideration only the physical length of the segments. Using only the length implies that the direct segment is removed in both situations presented in Figure 33.

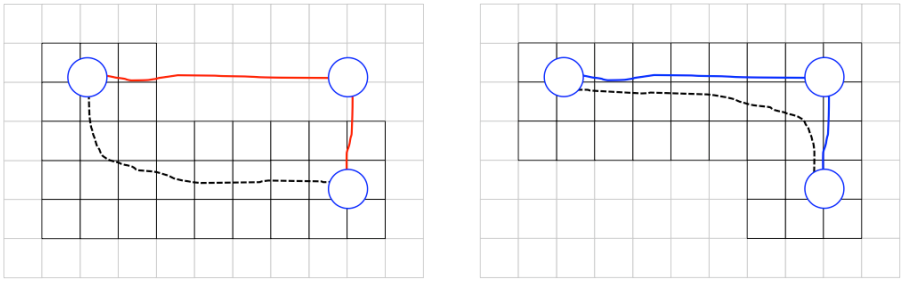


Figure 33. The specified segment (dashed line) is shown together with the dilated cell representation (dark cells). In the example on the right, two other road segments are included in the region defined by the dilated cells and are valid. The example on the left has no valid segments.

A question that has not yet been answered is how to score the quality of a generated road network. Virtually all studies to date have relied on visual inspection of the results, with generated maps being compared with existing maps or satellite imagery. We propose two novel ways of comparing a generated map with ground truth obtained from OSM. First, we evaluate the intersections using the same technique that is used to compute clustering quality in Fränti et al. [2014].

We next propose a way to evaluate the road segments connecting the intersections. We use grid cells to measure whether the ground truth segments are properly identified. The measure is also sensitive to redundant segments (Figure 34).

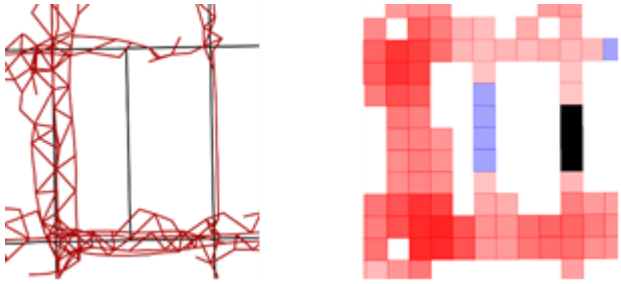


Figure 34. Left panel: ground truth segments (black) and inferred segments (red). Right panel: red cells show over-emphasis proportional to the colour intensity; blue cells show segments that were not represented at all; black cells mean correct representation.

Table 11 shows summary data on the quality of the road network produced by CellNet and a comparison with the three other methods. The visual method yielded the highest precision for the Chicago dataset, because the high density of routes in that

dataset produced good visual features. The recall was low because the portion of the dataset that fell below the density threshold was omitted. The clustering and merging methods displayed high recall, because – unlike the visual approach – they did not drop out part of the dataset according to a threshold. However, the precision of the clustering and merging methods was low because they detected too many intersections in regions with many routes and low GPS accuracy. CellNet achieved the most balanced results in terms of precision and recall, and produced the best result in terms of the F-score.

Table 11. Comparison of CellNet with other popular road network inference methods.

Chicago

	Intersections			Links		
Method	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Visual	97%	27%	42%	97%	27%	42%
Clustering	14%	94%	24%	17%	94%	28%
Merging	5%	90%	10%	7%	70%	10%
CellNet	77%	90%	84%	81%	68%	75%

Joensuu

	Intersections			Links		
Method	**Precision**	**Recall**	**F-score**	**Precision**	**Recall**	**F-score**
Visual	54%	63%	58%	56%	38%	46%
Clustering	22%	85%	36%	16%	92%	27%
Merging	22%	52%	31%	13%	28%	18%
CellNet	71%	68%	69%	68%	49%	58%

A potential challenge is the memory requirements of the generated network. The compared methods produce unnecessarily complex segments, which could be simplified by using polygonal approximation [Chen et al. 2009, Chen et al. 2012, Pikaz and Dinstein 1995]. We used the technique in Chen et al. [2012] and obtained networks of only 25% of the size of those produced by other methods.

6 SUMMARY OF CONTRIBUTIONS

This chapter summarizes the contributions of our five publications. In publication **[I]** we studied how routes are recorded, stored and visualized. In publications **[II]** and **[III]** we explained how a grid-based representation is useful when implementing four commonly used functions: similarity, inclusion, novelty and noteworthiness. Publication **[IV]** presented an application of the fast grid-based similarity search, namely gesture search, which allows users to search for routes by drawing a free-form shape on the map. Publication **[V]** presents a novel way of generating a road network from a route dataset.

In **[I]**, we proposed a method for recording GPS routes which allows online and offline capability and live tracking, and is efficient in terms of internet and battery usage. Using the polygonal approximation and cropping strategies allows the Mopsi system to query and display routes consisting of over 3.5 million points in under 2 seconds. As far as we know, no other online system even exists to achieve this and all systems show only the start points or just a single route at a time.

In **[II]**, we presented a new fast and intuitive way of computing route similarity using a grid-based approximation of the routes and set-based operations. The method is equipped with interpolation and dilation of the grid cells in order to cope with missing points and to handle the arbitrary grid division into cells.

In **[III]** we introduced four grid-based route operations: similarity, inclusion, novelty and noteworthiness. The methods were analysed in terms of their space requirements, computational complexity and indexing strategy. In that work, the similarity measure presented in **[II]** was redefined as "inclusion" and a new, improved similarity measure was introduced. Using the new similarity measure, a route similarity search strategy was presented and was shown to work in real time on a real-world dataset. We built an interactive tool for comparing and understanding different similarity measures and offered an application programming interface (API) for calling our newly presented measure. The API also supports calls to the other similarity measures. These are available in the web page[15] of **[III]**.

In **[IV]** we showed that the similarity search method can be used to search for a route if the user remembers the approximate shape but not the time. This feature improves the user's experience when searching routes in large data collections, compared with the traditional interface described in **[I]**.

[15] http://cs.uef.fi/mopsi/routes/grid

In **[V]** we present a new method for road network extraction, CellNet, which produces accurate results without the need to optimize parameters. We show that CellNet produces higher quality results than three conceptually different state-of-the-art methods.

7 CONCLUSIONS

We presented efficient ways to record, store and visualize route collections and demonstrated their efficiency within the real-world environment of Mopsi. We showed that polygonal approximation and cropping are very useful in reducing the amount of data, and these techniques also allow the display of large route collections on the map. In addition, we showed that the system is capable of displaying routes consisting of over 3.5 million points in less than 2 seconds.

Many popular route similarity measures exist, inspired by methods based on various fields – such as string matching, time-series analysis, curve comparison and set matching. Most of these methods are slow and are not intuitive for average users, who perceive routes as being shapes on a map. Our proposed similarity measure, C-SIM, uses the grid-based representation of routes to output a fast and intuitive measure of similarity. It was combined with an indexing strategy, which was demonstrated to perform similarity searches in real time on a database containing over 5,000 routes.

Searching for routes is not easy in large collections. We proposed gestures to be used for this purpose. We built a working system that allows users to draw the approximate shape of a route on the map; then, spatially similar routes are retrieved. This method is preferred over the traditional approach when the user cannot remember the date of the searched route.

To date, managing road networks still requires intensive manual editing. Our proposed method, CellNet, provides more accurate results on different datasets compared with other popular approaches, without the need for parameter optimization. In addition, the size of the generated network is reduced by using polygonal approximation to produce maps that require a quarter of the storage space needed by other automatically generated maps.

BIBLIOGRAPHY

Agrawal R., Faloutsos C. & Swami A. 1993. Efficient similarity search in sequence databases. *International Conference on Foundations of Data Organization and Algorithms (FODO 1993)*, Chicago, Illinois, USA, pp. 69-84

Alahakone A. U. & Ragavan V. 2009. Geospatial Information System for tracking and navigation of mobile objects. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2009)*, Singapore, pp. 875-880.

Almer A. & Stelzl H. 2002. Multimedia visualisation of geo-information for tourism regions based on remote sensing data. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(4), pp. 436-441.

Ananthanarayanan G., Haridasan M., Mohomed I., Terry D. & Thekkath C. A. 2009. Startrack: a framework for enabling track-based applications. *ACM international conference on Mobile systems, applications, and services (MobiSys 2009)*, Kraków, Poland, pp. 207-220.

Bao T., Cao H., Yang Q., Chen E. & Tian J. 2012. Mining significant places from cell id trajectories: A geo-grid based approach. *IEEE International Conference on Mobile Data Management (MDM 2012)*, Bengaluru, India, pp. 288-293

Barsi, A. & Heipke, C. 2003. Artificial neural networks for the detection of road junctions in aerial images. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences, 34(3/W8), pp. 113-118.

Berndt D. J. & Clifford J. 1994. Using dynamic time warping to find patterns in time series. *KDD workshop*, Vol. 10, No. 16, pp. 359-370.

Biagioni, J. & Eriksson, J. 2012. Inferring road maps from global positioning system traces. Transportation Research Record: Journal of the Transportation Research Board, 2291(1), pp. 61-71.

Cao L. & Krumm J. 2009. From GPS traces to a routable road map. *ACM SIGSPATIAL international conference on advances in geographic information systems (ACM SIGSPATIAL GIS 2009)*, Seattle, Washington, USA, pp. 3-12

Chen C. & Cheng Y. 2008. Roads digital map generation with multi-track GPS data. *IEEE In International Workshop on Education Technology and Training, 2008 and 2008 International Workshop on Geoscience and Remote Sensing. (ETT and GRS 2008)*, Vol. 1, pp. 508-511.

Chen J., Wang W., Liu L. & Song J. 2011. Clustering of trajectories based on Hausdorff distance. *IEEE International Conference on Electronics, Communications and Control (ICECC 2011)*, Ningbo, China, pp. 1940-1944.

Chen L., Özsu M. T. & Oria V. 2005. Robust and fast similarity search for moving object trajectories. *ACM SIGMOD international conference on Management of data and Symposium on Principles Database and Systems (SIGMOD/PODS 2005)*, Baltimore, MD, USA, pp. 491-502

Chen L. & Ng R. 2004. On the marriage of lp-norms and edit distance. *International Conference on Very Large Data Bases-Volume (VLDB 2004)*, Toronto, Canada, pp. 792–803

Chen M., Xu M. & Fränti P. 2012. A fast multiresolution polygonal approximation algorithm for GPS trajectory simplification. *IEEE Transactions on Image Processing*, 21(5), pp. 2770-2785.

Chen M., Xu M., & Fränti P. 2012. Compression of GPS trajectories. *IEEE Data Compression Conference (DCC 2012)*, pp. 62-71.

Chen Y., Jiang K., Zheng Y., Li C. & Yu N. 2009. Trajectory simplification method for location-based social networking services. *ACM International Workshop on Location Based Social Networks*, Seattle, USA, pp. 33-40.

Cirelli M. & Nakamura R. 2014. A Survey on Multi-touch Gesture Recognition and Multi-touch Frameworks. *ACM Conference on Interactive Tabletops and Surfaces (ITS 2014)*, Dresden, Germany, pp. 35-44.

Cormen T. H. 2009. Introduction to algorithms, *MIT press*.

Davies, J. J., A. R. Beresford & A. Hopper. 2006. Scalable, Distributed, Real-Time Map Generation. IEEE Pervasive Computing, Vol. 5, No. 4, pp. 47–54.

Edelkamp, S. & Schrödl S.. 2003. Route Planning and Map Inference with Global Positioning Traces. In Computer Science in Perspective, Springer Berlin Heidelberg, Vol. 2598, pp. 128–151.

Eiter T. & Mannila H. 1994. Computing discrete Fréchet distance. *Tech. Report CD-TR 94/64, Information Systems Department*, Technical University of Vienna

Evans M. R., Oliver D., Shekhar S. & Harvey F. 2013. Fast and exact network trajectory similarity computation: a case-study on bicycle corridor planning. *ACM SIGKDD International Workshop on Urban Computing (UrbComp 2013)*, Chicago, IL, USA, 9

Fathi A. & Krumm J. 2010. Detecting road intersections from gps traces. *International Conference on Geographic Information Science (GIScience 2010)*, Zurich, Switzerland, pp. 56-69

Follin J. M., Bouju A., Bertrand F. & Boursier P. 2003. Management of multi-resolution data in a mobile spatial information visualization system. IEEE International Conference on Web Information Systems Engineering Workshops, 2003. pp. 92-99.

Frentzos E., Gratsias K. & Theodoridis Y. 2007 .Index-based most similar trajectory search. *IEEE International Conference on Data Engineering (ICDE 2007)*, Istanbul, Turkey, pp. 816-825

Fränti P. & Kivijärvi J. 2000. Randomized local search algorithm for the clustering problem. *Pattern Analysis and Applications*, 3 (4), pp. 358-369.

Fränti P., Chen J. & Tabarcea A. 2011. Four Aspects of Relevance in Sharing Location-based Media: Content, Time, Location and Network. *In WEBIST*, pp. 413-417.

Fränti P., Rezaei M. & Zhao Q. 2014. Centroid index: Cluster level similarity measure, *Pattern Recognition*, 47 (9), pp. 3034-3045.

Gali N. & Fränti P. 2016. Content-based title extraction from web page. International Conference on Web Information Systems and Technologies (WEBIST 2016), Rome, Italy, pp. 204-210.

Gali N., Tabarcea A. & Fränti P. 2015. Extracting representative image from web page. International Conference on Web Information Systems & Technologies (WEBIST 2015), pp. 411-419.

Gradshteyn I. S. & Ryzhik I. M. 2000. Tables of Integrals, Series, and Products, 6th ed. San Diego, *CA: Academic Press*, pp. 1114-1125.

Guttman A. 1984. R-trees: a dynamic index structure for spatial searching. 1984. *ACM SIGMOD international conference on Management of data (SIGMOD 1984)*, New York, NY, USA, 47-57

Hamilton J.D. 1994. Time series analysis (Vol. 2). *Princeton: Princeton university press*

Haridasan M., Mohomed I., Terry D., Thekkath C. A. & Zhang, L. (2010, October). StarTrack Next Generation: A Scalable Infrastructure for Track-Based Applications. *ACM/USENIX Symposium on Operating Systems Design and Implementation (OSDI 2010)*, Vancouver, Canada, pp. 409-422.

Hautamäki V., Nykänen P. & Fränti P. 2008. Time-series clustering by approximate prototypes, *IAPR International Conference on Pattern Recognition (ICPR'08)*, Tampa, Florida, USA, December 2008, pp ???.

Horozov T., Narasimhan N. & Vasudevan V. 2006. Using location for personalized POI recommendations in mobile environments. IEEE International symposium on Applications and the internet, (pp. 6-pp). ???.

Hu, J., Razdan, A., Femiani, J. C., Cui, M. & Wonka, P. 2007. Road network extraction and intersection detection from aerial images by tracking road footprints. IEEE Transactions on Geoscience and Remote Sensing, 45(12), pp. 4144-4157.

Karam M. & Schraefel M. C. 2015. A taxonomy of Gestures in Human Computer Interaction. *ACM Transactions on Computer-Human Interactions*, 2015. (in press)

Karimi H.A. & Lockhart J.T. 1993. GPS-based tracking systems for taxi cab fleet operations. *IEEE Conference on Vehicle Navigation and Information Systems*, pp. 679-682.

Kasemsuppakorn P. & Karimi H.A. 2009. Personalised routing for wheelchair navigation. *Journal of Location Based Services*, 3(1), pp.24-54.

Kasemsuppakorn P. & Karimi H.A. 2013. A pedestrian network construction algorithm based on multiple GPS traces. *Transportation research part C: emerging technologies*, 26, pp. 285-300.

Kennedy M. & Kopp S. 2001. Understanding Map Projections. *ESRI Press*.

Kristensson P. O. & Zhai S. 2007. Command strokes with and without preview: using pen gestures on keyboard for command selection. *SIGCHI Conference on Human Factors in Computing Systems (CHI 2007)*, New York, USA, pp. 1137-1146.

Krumm J. & Horvitz E. 2006. Predestination: Inferring destinations from partial trajectories. *In Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp '06)*, Orange County, CA, USA, pp. 243-260.

Krumm J. & Horvitz E. 2017. Risk-Aware Planning: Methods and Case Study for Safer Driving Routes. *In Twenty-Ninth Innovative Applications of Artificial Intelligence Conference*, pp. 4708-4714.

Lehtimäki T. M., Partala T., Luimula M. & Verronen P. 2008. LocaweRoute: an advanced route history visualization for mobile devices. ACM working conference on advanced visual interfaces, pp. 392-395.

Li Y. 2010. Gesture search: a tool for fast mobile data access. *ACM Symposium on User interface software and technology (UIST 2010)*, New York, USA, pp. 87-96.

McCullough A., James P. & Barr S. 2011. A Service Oriented Geoprocessing System for Real-Time Road Traffic Monitoring. *Transactions in GIS*, 15(5), 651-665.

Morris S., Morris A. & Barnard K. 2004. Digital trail libraries. *ACM/IEEE Conference on Digital Libraries (ICDL 2004)*, New Delhi, India, pp. 63-71.

Ni J. & Ravishankar C. V. 2007. Indexing spatio-temporal trajectories with efficient polynomial approximations. *IEEE Transactions on Knowledge and Data Engineering*, 19(5).

Niehöfer B., Lewandowski A., Burda R., Wietfeld C., Bauer F. & Lüert O. 2010. Community Map Generation based on Trace-Collection for GNSS Outdoor and RF-based Indoor Localization Applications. *International Journal on Advances in Intelligent Systems* Volume 2, Number 4, 2009.

Pang L. X., Chawla S., Liu W. & Zheng Y. 2013. On detection of emerging anomalous traffic patterns using GPS data. *Data & Knowledge Engineering (DKE)*, 87, pp. 357-373.

Pelekis N., Kopanaki, I., Kotsifakos E. E., Frentzos E. & Theodoridis Y. 2011. Clustering uncertain trajectories. *Knowledge and information systems*, 28(1), pp. 117-147.

Pikaz A. & Dinstein I. 1995. An algorithm for polygonal approximation based on iterative point elimination, *Pattern Recognition Letters*, 16 (6), 557–563, Jun. 1995.

Rezaei M., Gali N. & Fränti P. 2015. ClRank:a method for keyword extraction from web pages using clustering and distribution of nouns", IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2015), pp. 79-84.

Rockafellar R. T. & Wets R. J. B. 2009. *Variational analysis (Vol. 317). Springer Science & Business Media*

Rousseeuw P. J. & Kaufman L. 1990. *Finding Groups in Data.* Wiley Online Library.

Rezaei M. & Fränti P. 2017. Clustering large geo-referenced data on maps for clutter removal (*submitted*)

Salvador S. & Chan P. 2004. FastDTW: Toward accurate dynamic time warping in linear time and space. *ACM International Conference on Knowledge Discovery and Data Mining Workshop on Mining Temporal and Sequential Data (SIGKDD 2004)*, Seatle, Washington, USA, pp. 70–80.

Shang S., Ding R., Yuan B., Xie K., Zheng K. & Kalnis P. 2012. User oriented trajectory search for trip recommendation. *ACM International Conference on Extending Database Technology*, Berlin, Germany, pp. 156-167.

Tabarcea A., Wan Z., Waga K. & Fränti P. 2013. O-mopsi: Mobile orienteering game using geotagged photos. CONFERENCE, pp. 300–303.

Tavakoli, M. & Rosenfeld, A. 1982. Building and road extraction from aerial photographs. IEEE Transactions on Systems, Man, and Cybernetics, 12, pp. 84-91.

Vlachos M., Gunopulos D. & Kollios G. 2002. Robust similarity measures for mobile object trajectories. *International Workshop on Database and Expert Systems Applications (DEXA 2002)*, Aix en Provence, France, pp. 721-726

Vlachos M, Kollios G. & Gunopulos D. 2002. Discovering similar multidimensional trajectories. *IEEE International Conference on Data Engineering (ICDE 2002)*, pp. 673-684

Waga K., Tabarcea A., Chen M. & Fränti P. 2012. Detecting movement type by route segmentation and classification. *IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2012)*, Pittsburgh, USA, pp. 508-513.

Waga K., Tabarcea A. & Fränti P. 2011. Context aware recommendation of location-based data. *IEEE International conference on System Theory, Control, and Computing (ICSTCC 2011)*, pp. 1-6.

Waga K., Tabarcea A. & Fränti P. 2012. Recommendation of points of interest from user generated data collection. *IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2012)*, Pittsburgh, USA, pp. 550-555.

Wang H. & Liu K. 2012. User oriented trajectory similarity search. *ACM SIGKDD International Workshop on Urban Computing (UrbComp 2012)*, Beijing, China, pp. 103-110

Wang H., Su H., Zheng K., Sadiq S. & Zhou X. 2013. An effectiveness study on trajectory similarity measures. *Australasian Database Conference (ADC 2013)*, Adelaide, Australia, pp. 13-22.

Wei L., Zheng Y. & Peng W. 2012. Constructing popular routes from uncertain trajectories. *In Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD '12)*, Beijing, China, pp. 195-203.

Yanagisawa Y., Akahani J. & Satoh T. 2003. Shape-based similarity query for trajectory of mobile objects. *International Conference on Mobile Data Management (MDM 2003)*, Melbourne, Australia, pp. 63-77

Ying J. J. C., Lu E. H. C., Lee W. C., Weng T. C. & Tseng V. S. 2010. Mining user similarity from semantic trajectories. *ACM SIGSPATIAL International Workshop on Location Based Social Networks (ACM SIGSPATIAL GIS 2010)*, San Jose, CA, USA, pp. 19-26

Ying X., Xu Z. & Yin W. G. 2009. Cluster-based congestion outlier detection method on trajectory data. *IEEE International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2009)*, Tianjin, China, pp. 243-247

Zhang D., Li N., Zhou Z. H., Chen C., Sun L. & Li S. 2011. iBAT: detecting anomalous taxi trajectories from GPS traces. *ACM international conference on Ubiquitous computing (UbiComp 2011)*, Beijing, China, pp. 99-108

Zheng V. W., Zheng Y., Xie X & Yang Q. 2010. Collaborative location and activity recommendations with gps history data. *In Proceedings of the 19th ACM International Conference on World Wide Web (WWW '10)*, New York, NY, USA, pp. 1029-1038.

Zheng Y. & Zhou X. 2011. Computing with spatial trajectories, *Springer Science & Business Media*

Zheng Y., Wang L., Zhang R., Xie X. & Ma W. Y. 2008. GeoLife: Managing and understanding your past life over maps. *IEEE International Conference on Mobile Data Management (MDM 2008)*, Beijing, China, pp. 211-212.