Sketch-on-Map: Spatial Queries for Retrieving Human Locomotion Patterns from Continuously Archived GPS Data

Gamhewage C. de Silva, Toshihiko Yamasaki, Kiyoharu Aizawa

Department of Information and Communication Engineering
The University of Tokyo, 102B2, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan.
{chamds,yamasaki,aizawa}@hal.t.u-tokyo.ac.jp

Abstract. We propose a system for retrieving human locomotion patterns from tracking data captured within a large geographical area, over a long period of time. A GPS receiver continuously captures data regarding the location of the person carrying it. A constrained agglomerative hierarchical clustering algorithm segments these data according to the person's navigational behavior. Sketches made on a map displayed on a computer screen are used for specifying queries regarding locomotion patterns. Two basic sketch primitives, selected based on a user study, are combined to form five different types of queries. We implement algorithms to analyze a sketch made by a user, identify the query, and retrieve results from the collection of data. A graphical user interface combines the user interaction strategy and algorithms, and allows hierarchical querying and visualization of intermediate results. The sketch-based user interaction strategy facilitates querying for locomotion patterns in an intuitive and unambiguous manner.

Keywords: Sketch-based querying, Locomotion patterns, Spatial queries, GPS data, Multimedia retrieval

1 Introduction

Continuously archived location data, obtained by tracking persons or objects, are useful in several application areas such as surveillance, navigational assistance, and behavioral studies. While different types of sensors are used for tracking indoors or in medium-sized outdoor areas, the Global Positioning System (GPS) provides a means of tracking over a very wide geographical area with reasonable accuracy. The tracking data can be indexed by date, time and location, in order to speed up retrieval.

In some applications such as traffic parameter estimation, it is often necessary to search the archived data for a particular pattern of locomotion. Such queries can be categorized as *spatial queries* [8]. This is usually performed by reducing the search space using other criteria and viewing the tracking results manually to retrieve the desired locomotion pattern. Ability to query a collection of tracking data directly by a particular locomotion pattern will greatly enhance the efficiency of retrieval, and allow identification and analysis of long term behavioral patterns. Therefore, developing spatial queries to retrieve locomotion patterns in large collections of tracking data solves an important research problem.

However, facilitating spatial queries on a collection of tracking data from a large geographical area is a challenging task. This task is further complicated when the data are captured over a long period of time. There should be an intuitive and non-restrictive way to input queries on locomotion patterns into a computer. Such queries can entail very different levels of complexity. For example, possible queries might include the following:

- "Which route did I take when I went from Tokyo to Yokohama last month?"
- "How many times have I travelled to places outside Tokyo since January?"

Sketching is a common method used by people to specify or describe patterns of movement. With several common factors such as area, distance and direction, sketching and locomotion has an intuitive mapping between them. Despite different sketching habits and techniques, people are able to interpret sketches made by others. Therefore, sketching is a highly prospective candidate for synthesizing queries on locomotion patterns.

In this research, we propose a system for spatial querying of locomotion patterns in a large collection of continuously archived GPS data. The queries are specified by making sketches with a pointing device on a map displayed on a computer screen. We design a user interaction strategy that facilitates searching for different types of locomotion patterns with sketches that are simple, intuitive and unambiguous. We also design and implement algorithms for searching for segments of GPS data that match the patterns specified by the queries. The system consists of a graphical user interface that combines the user interaction strategy and the algorithms with interactive visualization for efficient retrieval.

The rest of this paper is organized as follows: Section 2 is a brief review of related work; Section 3 outlines the aquisition of GPS data; Section 4 describes the user interaction strategy, and the algorithms for retrieval; Section 5 presents the user interface and describes how the systems is used for retrieval; Section 6 contains a brief discussion regarding the design and implementation issues; Section 7 concludes the paper with suggestions for future directions.

2 Related Work

A number of researches use continuously archived GPS data to associate photos and video with locations and create visualizations on maps [5][9][15]. Morris et al. [12] propose a framework for automatic modeling of recreation travel using GPS data collected at recreation sites. The *Cabspotting Project* [1] analyses a collection of GPS data from a large number of taxi cabs in an urban area in order to discover the invisible dynamics contained within the data. Liao et al. [11] propose a hierarchical Markov model that can learn and infer a person's daily movements using GPS data.

There has been some research towards a framework for spatial querying of locomotion patterns. Egenhofer [4] demonstrated how imprecise spatial queries can be dealt with in a comprehensible manner, using topological relations. A relational algebra is proposed there, for verifying the consistency of the resulting topological representations. Gottfried [7] uses a locomotion base and a set of relations to represent locomotion patterns, with emphasis on healthcare applications. However, an effective user interaction strategy for submitting queries is essential to utilize the above framework to retrieve locomotion patterns.

So far, there has been little research on user interfaces for spatial querying. Ivanov and Wren [8] use simple spatial queries to specify the direction of movement along a corridor for video retrieval from surveillance cameras. Kimber et al. [10] propose a method of object based video playback, that can be used as a means of querying for locomotion patterns. Recently, we proposed three basic types of sketch-based queries for retrieval of human locomotion patterns from a home-like smart environment [2]. However, having been designed for a small area with bounded regions, this system is not scalable to an outdoor environment.

3 Data Acquisition

This research is based on a collection of continuously archived GPS data from a handheld GPS receiver. One of the authors has been continuously carrying a *Garmin*® *GPSmap 60CSx* GPS receiver for data acquisition, since November 2007. Data are collected at all times other than when signals are not received. The author also carried a *SenseCam*® device [16] on certain days, for verification purposes. The following is a summary of the data considered in this work:

– Duration: November 21, 2007 to April 20, 2008
– Area: Covers an approximate land area of 200,000km^2
– Altitudes: 0 to 950 m above mean sea level
– Speeds: Up to 294 km/h
– Total distance travelled: 13500 km (approx.)
– Signal reception: 8-24 hours per day (approx.)

The sampling interval of the GPS receiver is 1 second. Contiguous samples corresponding to the same GPS coordinates are combined to form a *location record* in the format shown in Table 1. The average speed is calculated by dividing the distance between current and previous samples by the difference between the starting timestamps of the two entries. The direction recorded is the angle of the vector from previous location to current, measured clockwise from north. The number of entries during a day varies between approximately 2000 and 20,000, according to the movement of the person and the availability of signals. The average number of location records is approximately 3000 per day.

A few problems arise when extracting location and motion information using GPS data. Due to poor or no reception of signals, the receiver might fail to record data at certain locations. Hence, the collected data form a discrete time sequence that is 'undefined' for certain time intervals. Several factors contribute

Table 1. Format of GPS Location Records

Date and Time	Altitude	Distance	Duration	Speed	Direction	Lattitude	Longitude
2007/11/22 9:22:02	29 m	24 m	0:00:13	7 km/h	199°	N35°44.713'	E139°44.755'
2007/11/22 9:22:15	29 m	39 m	0:00:21	7 km/h	200°	N35°44.700'	E139°44.749'
2007/11/22 9:22:36	29 m	31 m	0:00:17	7 km/h	198°	N35°44.681'	E139°44.741'

to the amount of noise present in GPS data [13], making it difficult to model and eliminate noise. Therefore, the proposed system has to be designed to perform well with incomplete and noisy data.

4 System Description

We take the following approach to facilitate querying the continuously archived location records for locomotion patterns by sketching on a geographical map. First, the GPS data are grouped to form segments according to two basic types of locomotion. The user interaction strategy is designed by identifying basic sketch primitives required for defining locomotion patterns and combining them to form queries. Algorithms are designed for querying the collection of segments using parameters extracted from the sketched queries. A grapical user interface is designed and implemented to integrate the segments, queries and algorithms. The following subsections decribe each of these in detail.

4.1 Segmentation of GPS Data

The result of data acquisition, as described in Section 3, is a large number of location records ordered by time. We intend to combine these records to form a set of non-overlapping *locomotion segments*, representing the nature of the person's movement. The user queries can now be made on the collection of the segments instead of the entire data set, allowing more efficient retrieval. We select the following two classes of locomotion, for segmentation of location records:

1. **Navigating**: instances of locomotion where the person makes a regular change of location with time. Examples are walking, driving or riding in a vehicle.
2. **Non-navigating**: instances of locomotion where a person stays within a small neighborhood. Examples are a room, a bus stop etc.

We employ a constrained hierarchical clustering algorithm [3] to cluster the location records into locomotion segments. This algorithm is capable of classifying location records at an average accuracy of 94.4%. The average accuracy of the timestamps at segment boundaries is approximately 2 minutes and 45 seconds.

Table 2. Attributes of *Navigating* Segments

Starting Time	Ending Time	Starting Location	Ending Location
2007/11/27 10:32:41	2007/11/27 13:21:20	N35 °37.866', E139 °16.032'	N35 °37.820', E139 °16.039'
2007/11/27 13:27:40	2007/11/27 16:29:11	N35 °37.864', E139 °16.024'	N35 °39.894', E139 °45.695'

Table 3. Attributes of *Non-nagivating* Segments

From	To	Mean Location	Maximum deviation
2007/11/27 10:16:23	2007/11/27 10:32:41	N35 °37.938' ,E139 °16.188'	0.001161
2007/11/27 13:21:20	2007/11/27 13:27:40	N35 °37.844' ,E139 °16.036'	0.000276

The location records are indexed by the results of classification, for efficient querying. Tables 2 and 3 show example entries of indices for navigating and non-navigating segments respectively. The standard deviations of the lattitudes and longitudes of locations in non-navigating segments are calculated and compared, and the larger value is recorded as *maximum deviation* in Table 3. This is intended to be used in visualization of results.

4.2 User Interaction Strategy

Our objective here is to design a user interaction strategy that allows the user to query for locomotion patterns by making a sketch pattern on a map. This should allow the user to submit different types of queries in a simple and intuitive manner. There should be no ambiguity between different types of queries. The relative complexity of queries is also important. Less specific queries, resulting in a large amount of results, should take less effort to sketch. On the other hand, it is fine for specific and more detailed queries to require more time and effort to sketch them. The algorithms should be sufficiently robust to interpret the sketches correctly, despite different sketching habits and speeds. We take a user-centered approach based on that of a previous system we developed for querying locomotion patterns in an indoor environment [2], for designing a strategy that fulfills the above objectives.

In real-life descriptions of locomotion patterns, a person is usually referred to as being within a *region*, or moving along a given *path* within or between such regions. Therefore we identify the entities "region" and "path" as the *query primitives* for locomotion patterns. The size and boundaries of regions are sometimes approximate or even ambiguous, while they can be precisely specified at other times. When describing a path, sometimes only the starting and ending locations are important. On other occasions, the path traversed is important. This diversity of detail is inherent to sketching, making it a strong candidate for specifying locomotion patterns.

We select *sketch primitives* for the query primitives mentioned above, based on the results of a user study on sketching locomotion patterns. During one section of this study, we asked the subjects to sketch multiple locomotion patterns and studied the common notations among sketches. We observed that a region was specified with a closed or near-closed curve in approximately 85% of the sketches. In 75% of the sketches, paths were specified with arrows with arrowheads drawn in different styles. It was also observed that some of the subjects implied the direction of a path by the direction the line was drawn, instead of drawing an arrowhead. Based on these observations, we select the following two sketch primitives for a region and a path. The user draws a closed curve on the map to specify the enclosing region. A path is specified by sketching a line, to which an arrowhead is automatically added. These two sketch primitives form the basis of the following detailed spatial queries:

- **Type 1: staying within a region**: The user specifies the region by sketching a closed curve around it. Figure 1a corresponds to the query “in Ireland.”
- **Type 2: entering a region**: The user specifies the region, and draws a path into the region from outside of it. Figure 1b corresponds to the query “entering Stanford University premises”
- **Type 3: leaving a region**: The user specifies the region, and draws a path from inside of the region to the outside. Figure 1c corresponds to the query “leaving Stanford University premises.”
- **Type 4: moving from one region to another, irrespective of the actual path taken**: The user specifies the two regions (in any order), and then draws a path from the originating region to the destination. Figure 1d corresponds to the query “From Dayton to Columbus.”
- **Type 5: specific path**: The user draws the path that he/she wishes to retrieve, on the map. The path has to be drawn as an open curve, to prevent misdetection as a region. Figure 1e corresponds to a query for a specific path within the premises of the University of Tokyo.

The queries are sketched on images created by extracting map segments from the *Google Maps* Database [6]. A query results in one or more ordered set of points in the form of $P = \{p_1, p_2, ...p_N\}$ where $p_i = (X_i, Y_i)$ are specified in image coordinates. After preprocessing and coordinate conversion (from screen coordinates to latitudes and longitudes), the system analyzes the sketch and determines the type of query. Search algorithms are selected accordingly to extract and display the results. The following section describes these stages in detail.

4.3 Search Algorithms

First, the sets of points are processed to identify the type of sketch primitive they belong to. If a set of points forms a closed curve, it is identified as a region primitive and trimmed to remove the parts that do not enclose the region. If a set of points does not form a closed curve, it is identified as a path primitive.

The type of the query is determined after preprocessing using a hierarchical decision making process, following the columns in Table 4 from left to right. This

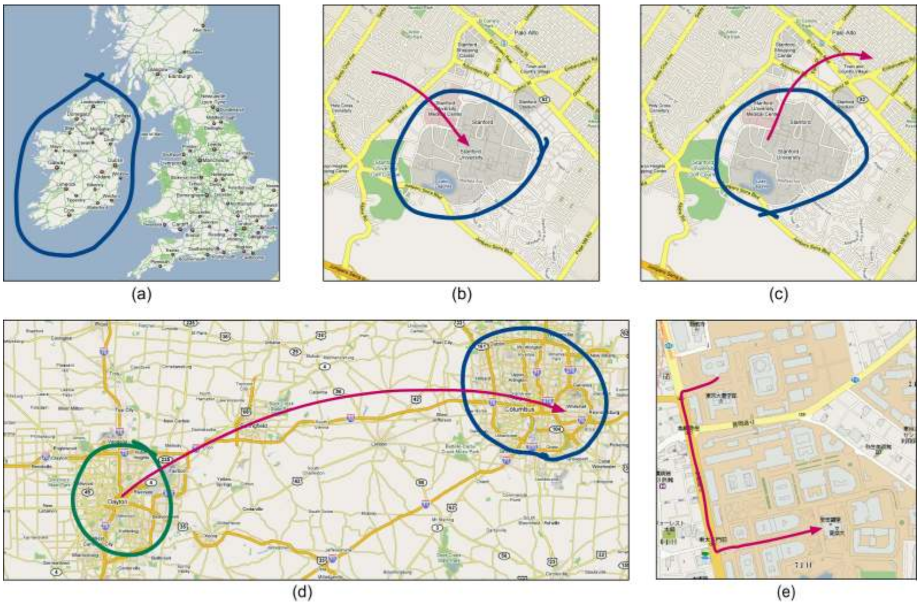


Fig. 1. Different types of spatial queries.

Table 4. Identifying the Type of Query

No. of Primitives	No. of Regions	Relationship between the path and region/s	Query Type
1	1	N/A	1
1	0	N/A	5
2	1	Path starts inside the region	2
2	1	Path finishes inside the region	3
3	2	Path starts inside one region and finishes inside the other	4

method prevents ambiguities in interpretation of query types. The possibilities that are not listed in the table are not recognized as valid queries. However, the decision making process can be expanded to cover such possibilities, where necessary. For example, a query with multiple regions and no path can be used to query for segments within only those regions.

In the next stage, the points are converted from image coordinates to geographical coordinates (latitudes and longitudes). Since Google Maps use Mercator's projection with an adjustment, we calculate these points using the inverse Mercator projection [14]. After conversion, a path is represented as an ordered set of points $P = \{p_1, p_2, ..., p_N\}$ on the path. A region is represented by a set of points $R = \{r_1, r_2, ..., r_K\}$ along its perimeter. The detected region(s) and path are submitted as input to the search algorithm for the appropriate query type. The following is a description of the search algorithms we propose:

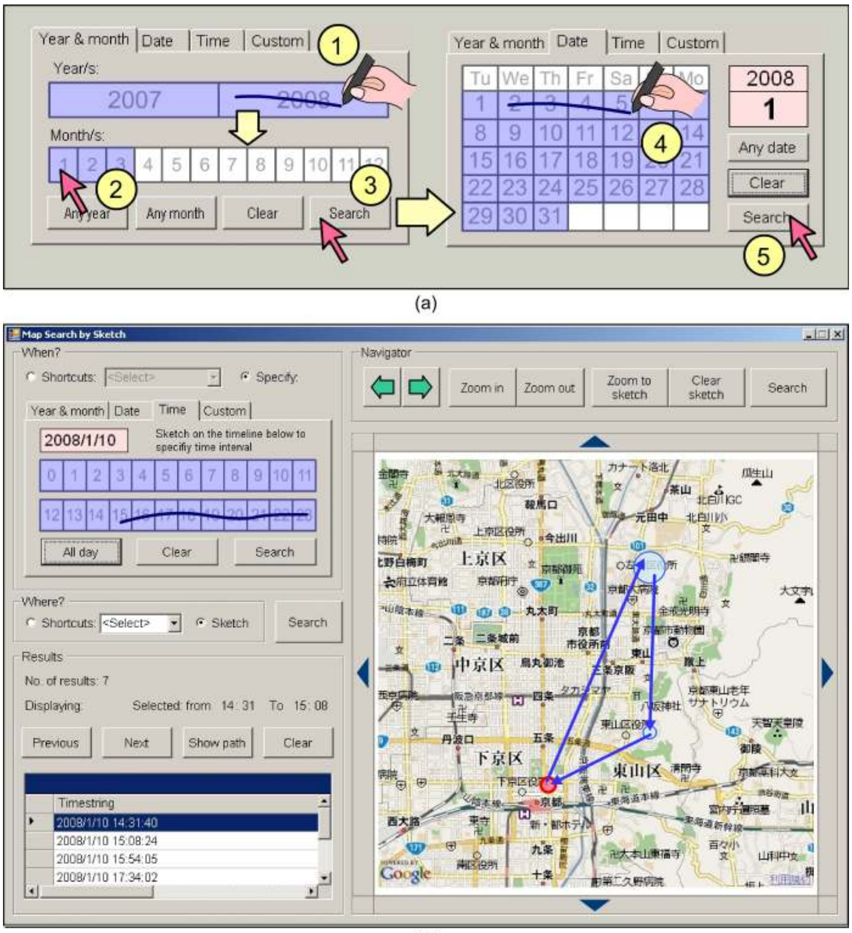
– **Query type 1**
  To perform this query type, we retrieve all segments (both navigating and non-navigating) that are contained within the region specified by the sketch. The results are ordered by the starting time of the segments.
– **Query type 2**
  We retrieve all navigating segments with their starting points contained in the sketched region. The results are ordered by the starting time of the segments.
– **Query type 3**
  We retrieve all navigating segments with their end points contained in the sketched region. The results are ordered by the starting time of the segments.
– **Query type 4**
  Let R_1 be the region where the path starts from, and R_2 be the region where the path ends. First, we extract navigating segments that have starting points in R_1. From this set of segments, the set of segments with end points in R_2 are extracted. The results are ordered by the starting time of the segments.
– **Query type 5**
  We define a *search area* by expanding the dimensions of the bounding box of the sketched path by 10% in each direction. We extract a set of *candidate paths* by selecting all navigating segments contained within this search area. For each candidate path $C = \{c_1, c_2, ..., c_M\}$ selected as above, we apply the following directional matching algorithm.

1. Set overall mean distance $D = 0$
2. for the first point c_1 in the selected candidate path C, find the closest point p_a in P
3. Add the geographical distance between c_1 and p_a to D
4. Repeat steps 2 and 3 for the next point in C and $P' = \{p_a, p_{a+1}, ..., p_N\}$ until all points in C are used in the calulation
5. Divide D by M and record the overall mean distance

This algorithm looks for navigating segments that are similar to the sketched path, while preserving direction. The results are presented to the user in ascending order of the overall mean distance. The common approach is to set a threshold value and remove the results that have higher distances than the threshold value. However, given that sketches can be imprecise, we believe that it is desirable to display all the results after ordering them accordng to the similarity.

5 User Interface Design

We design a graphical user interface based on the above strategy for retrieval of locomotion patterns. The interface is designed in such a way that only a pointing device is necessary to use it. The interface facilitates hierarchical segmentation of the data collection interactively using spatial queries, temporal queries or a combination of them to retrieve the desired results. The following sections describe how the user interface allows these types of queries.



(a)

(b)

Fig. 2. Submitting temporal queries by sketching.

5.1 Temporal Querying

The temporal queries also can be specified using sketches, making the interaction consistent with the query-by-sketch user interaction strategy for spatial queries. The users can sketch on a calendar-like interface to select a duration to retrieve data from. Figure 2a shows how a user queries for the duration "from the 2^{nd} to the 5^{th} of January, 2008". Where only one item is selected, clicking can be used in place of sketching, facilitating faster interaction.

Figure 2b shows the results retrieved from a temporal query. Once a selection is made, the map is scaled and scrolled to show only the regions where data have been captured. The non-navigating segments are shown as circles with the mean location as the center and scaled maximum deviation as the radius. The radius of the circle visualizes the confidence of the location estimation. This helps the

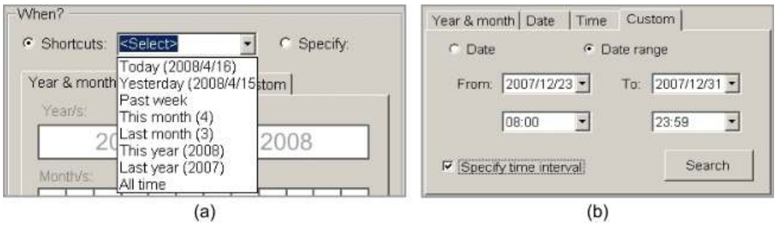


Fig. 3. Additional methods for submitting temporal queries.

user to identify the exact location with his memory and knowledge, where the accuracy of the data is less. The navigating segments are visualized with arrows. The detailed results for the segments are shown to the left of the map, and can be selected one at a time. The selected result is shown in red on the map.

Temporal querying is facilitated using two additional methods, to allow easier input. The user can choose some frequently-used time intervals directly from a combo box (Figure 3a). While the above methods are easy to use, sometimes it might be necessary to query the data for more precise time intervals. To allow this, controls for custom querying are provided in a separate tab (Figure 3b).

5.2 Spatial Querying

The user can start a spatial query by navigating the map to the region that he/she wishes to query. In addition to zooming, unzooming and scrolling, which are common methods for navigating to a location on an electronic map, we include the facility to *zoom to sketch*. The user can draw a closed curve to specify a region and zoom in to that region directly, as illustrated by Figure 4a. This is much faster and more efficient than the conventional method of zooming and scrolling, as both the location and the amount of zoom required can now be specified using a single interaction.

We implemented the user interaction strategy described in Section 4.3, to facilitate spatial querying. After the user reaches the desired area of the map, he can sketch spatial queries and retrieve the results. The results for non-navigating segments are shown in the same format as that of temporal querying. For navigating segments, the actual GPS data points are plotted on the map, joined by lines. The color of the data points and the line segments change from blue to red with time, to indicate direction. Figures 4b and c show spatial queries of Types 4 and 5 respectly, together with the retrieved results.

The user can perform more detailed searches, by combining temporal and spatial searches. A temporal query reduces the search space on the map by scrolling and zooming in to the area where GPS data were recorded for the specified interval. Instead of browsing the retrieved segments (described in Section 5.1), the user can submit a sketch and filter the results.

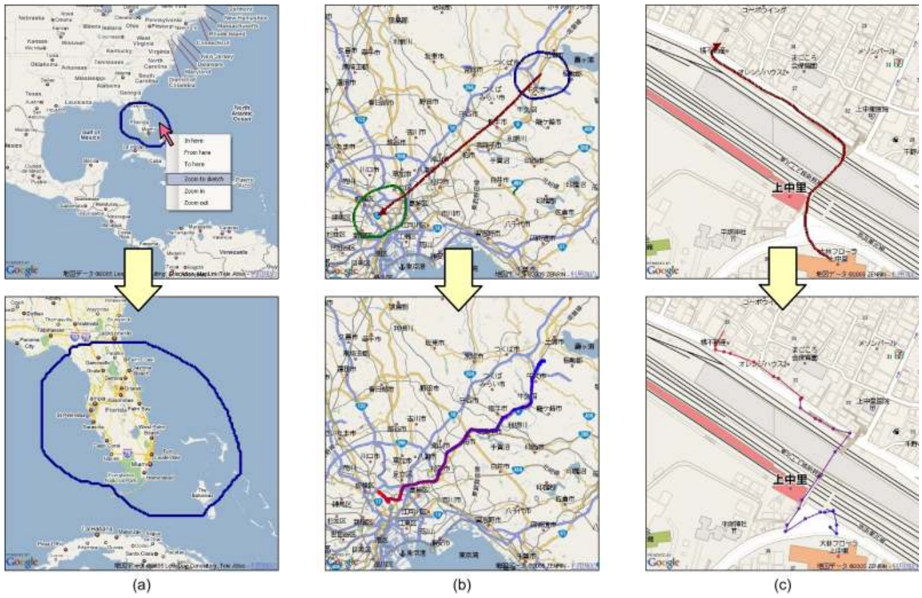


Fig. 4. Example queries and results.

6 Discussion

While there exist other researches that classify GPS data into activities at higher semantic levels, we decided to segment the data into only two basic classes of locomotion. The main reason for this decision is the logical mapping of simple sketches to such classes. The sketch primitive "path" maps naturally to *navigating* segments. Both *navigating* and *non-navigating* segments can be referred to as contained within a "region".

Speed is one of the important aspects in locomotion patterns. While regions and paths map naturally from sketches to locomotion patterns, specifying the speed seems not so straightforward. The system will be much more versatile if speed can be incorporated to the queries in a way that is easy to specify.

There were several programming issues in implementing the proposed user interaction strategy on publicly available application programming interfaces for mapping. For example, Google Maps API does not allow sketching in the same way as a typical drawing application (that is, by holding the mouse button down and moving the pointer to make the sketch). We use several programming workarounds to enable free hand sketching on the map.

7 Conclusion

We have developed a system for retrieving human locomotion patterns from continuously archived GPS data. The proposed sketch-based interaction strategy provides an intuitive way of querying the large collection of data. Five types of queries have been designed by combining two simple sketch primitives, making effective and unambiguous querying possible. The clustering algorithm and the user interaction strategy are integrated using an interface that supports both spatial and temporal querying, facilitating fast retreival of results. The clustering algorithm and the user interaction strategy can be applied to other applications based on GPS data, such as vehicle fleet monitoring and interfaces for navigational support systems.

The user interaction strategy can be enhanced by both designing new types of queries using the current primitives, and adding new primitives. Creating a formal model for the queries including time and speed will increase the versatility of spatial queries. We are working on designing user studies for evaluating the usability of the proposed system and obtaining feedback.

References

1. Cabspotting. Exploratorium: Museum of science, art and human perception. *http://cabspotting.org*, 2006.
2. G. C. de Silva, T. Yamasaki, and K. Aizawa. Spatial querying for retrieval of locomotion patterns in smart environments. In *Proc. ACM Multimedia 2007*, pages 803–806. ACM SIGMM, September 2007.
3. G. C. de Silva, T. Yamasaki, and K. Aizawa. Human Locomotion Segmentation by Constrained Hierarchical Clustering of GPS Data. *To appear in Proc. ITE 2008*, August 2008.
4. M. Egenhofer. Spatial query by sketch. *Journal of Visual Languages and Computing*, 8(4):403–424, August 1997.
5. J. Gemmell, A. Aris, and R. L. Roger. Telling stories with mylifebits. In *IEEE ICME 2005*, July 2005.
6. Google. Google maps api - google code. *http://code.google.com/apis/maps/*, 2008.
7. B. Gottfried. Spatial health systems. In *Pervasive Health Conference and Workshops*, pages 81–7. ACM SIGMM, December 2006.
8. Y. A. Ivanov and C. R. Wren. Toward spatial queries for spatial surveillance tasks. In *proc. Pervasive PTA 2006*, pages 803–806. ACM SIGMM, September 2006.
9. I.-J. Kim, S. C. Ahn, H. Ko, and H.-G. Kim. Persone : Personalized experience recording and searching on networked environment. In *ACM CARPE*, October 2006.
10. D. Kimber, T. Dunnigan, A. Girgensohn, F. Shipman, T. Turner, and T. Yang. Trailblazing: Video playback control by direct object manipulation. In *IEEE ICME 2007*, pages 1015–1018, July 2007.
11. L. Liao, D. J. Patterson, D. Fox, and H. Kautz. Learning and Inferring Transportation Routines. In *Artificial Intelligence* 171:5-6, pages 311–331, April 2007.
12. S. Morris, R. Gimblett, and K. Barnard. Probabilistic travel modeling using GPS data. In *Int'l Congress on Simulation and Modeling*, pages 61–66, December 2005.
13. B. W. Parkinson and S. W. Gilbert. Navstar: Global positioning system - ten years later. *Proceedings of the IEEE*, 71(10):1177–1186, October 1983.
14. J. P. Snyder. *Map Projections - A Working Manual, U.S. Geological Survey Professional Paper.* United States Government Printing Office, Washington, D.C., 1987.
15. D. Tancharoen, T. Yamasaki, and K. Aizawa. Practical experience recording and indexing of life long video. In *ACM CARPE 2005*, pages 61–66, November 2005.
16. K. Wood, R. Fleck, and L. Williams. Playing with SenseCam. In *UbiComp*, 2004.