Following Human Mobility Using Tweets

Mahdi Azmandian, Karan Singh, Ben Gelsey,
Yu-Han Chang, and Rajiv Maheswaran

Information Sciences Institute
University of Southern California
Marina del Rey, CA 90292
{azmandia,karans,gelsey}@usc.edu,
{ychang,maheswar}@isi.edu

Abstract. The availability of location-based agent data is growing rapidly, enabling new research into the behavior patterns of such agents in space and time. Previously, such analysis was limited to either small experiments with GPS-equipped agents, or proprietary datasets of human cell phone users that cannot be disseminated across the academic community for followup studies. In this paper, we study the movement patterns of Twitter users in London, Los Angeles, and Tokyo. We cluster these agents by their movement patterns across space and time. We also show that it is possible to infer part of the underlying transportation network from Tweets alone, and uncover interesting differences between the behaviors exhibited by users across these three cities.

1 Introduction

Location-based agents are becoming increasingly prevalent, and the data generated by these agents is a rich domain for data mining and interaction research. This involves an important issue, i.e. mining agent data to enhance agent performance, an important topic in agent mining [1,2]. Agents are sometimes location-based advertising bots, location-based game virtual characters, or humans using GPS-capable devices such as smartphones. Understanding location-based behavior can lead to better models of people and cities and help improve decision-making in domains from transportation networks to advertising. In this paper, we focus on geotagged data generated by Twitter-users, and apply data mining and visualization techniques to uncover both behavior patterns as well as the underlying network structure that supports the agent movements in London, Los Angeles and Tokyo. Understanding location-based behavior can be used to build more accurate models of human movement, which could then be deployed to any number of applications ranging from transportation modeling to personalized and predictive location-aware agent services to assessing "patterns of life" in foreign cities and towns.

We first introduce the notion of a *trace*, which is simply a user's trajectory extracted by connecting his tweet locations through the course of a day. Traces are broken down to fragments that correspond to periods where a user is tweeting

L. Cao et al.: ADMI 2012, LNAI 7607, pp. 139–149, 2013.

frequently. These fragments will also include updates of the user's location, which yields relatively accurate knowledge of the user's location during such a fragment. These fragments are used to construct a visualization we call a Trace-Based Heatmap. We then demonstrate an algorithm that can infer an undirected graph depicting the routes in the city where tweeting is most active. We also apply clustering techniques to this spatio-temporal data, and show that users can be roughly described by their geographic area and temporal description of their Twitter use. These results show initial promise towards agent models that can be learned from publicly available geo-tagged data sources.

2 Related Work

With the growing prevalence of social media such as Facebook and Twitter, researchers in social and network science have shown great interest in the datasets generated. Recently GPS-tagged information and "check-ins" have become quite widespread, giving rise to a new field of location data analysis [3]. In the past, the majority of research used human location data procured through mobile phone networks. These studies range from behavioral predictions [4], development of human movement [5], detecting anomalies [6], identification of points of interest from trajectories [7], discovery of the most popular routes [8], trajectory clustering [9], identification of movement flocks [10], and inference of transportation routines [11]. Such extensive research is justified considering the broadness of the potential applications, running the gamut from urban planners on the search for discovering daily routines [12], to biologists modeling the worldwide spread of pandemic influenza [13]. Similar techniques have also been used in ecology to track animal movements [14].

Visualization is a crucial tool in this human mobility research. Visualizations enhance tangibility of data mining outcomes and guide computational methods, providing a compensation for the computer's inability to incorporate humans' tacit knowledge [15]. Mobility data visualization has come a long way from the elementary idea of drawing arrows on an image [16] simply indicating direction of movement. Time-Geography study introduced the "space-time cube" technique; approaches to managing large-scale data have suggested data aggregation techniques such as the temporal histogram, traffic density surface, and accessibility surface; data filtering according to user-specified queries has been an alternative approach to handling large amounts of data [17]; and in a more recent endeavor a multidisciplinary approach was applied to develop a framework for the analysis of massive movement data taking advantage of a synergy of computational, database, and visual techniques [15].

Among these visualization techniques, the following approaches are most relevant to our work: The first approach is based on spatial, temporal or attribute proximity [18] (the space, time or attribute space are divided into compartments, into which the trajectories (viewed as a set of discrete movement events, i.e. geographic locations with respective time stamps) are projected). The second approach is also trajectory-based where trajectories are aggregated in their

entirety based on their similarity in geographic, temporal or attribute space (or a combination thereof) [19]. The "route-based" aggregation [18] is often performed by clustering in the data space or in an abstract projection thereof [20]. The third approach aggregates movement data based on their origin and destination and ignores the route between these two spatial locations, so that the movement is seen as a vector between the two locations, not as a set of recorded positions on a trajectory [18].

The novelty of our approach is that the data we use comes from "geotagged" tweets where we create trajectories via notions of "traces" and "fragments". We then develop various algorithms to turn these into heatmaps, route graphs, flows, behavior characterizations and temporal signatures for cities.

3 Data Visualization

Heatmap Constructions. Two types of heatmaps were generated for each area of study, a "Point-Based HeatMap" and a "Trace-Based HeatMap", the description of which will follow. For the Point-Based HeatMap, for every single tweet occurring at a particular gridpoint, an intensity increment of 3 units was applied to the cell, 2 units of intensity to the surrounding 8 cells, and one unit of intensity to the 16 cells encompassing the previous 9. For the Trace-Based HeatMap, for each fragment in each trace of each user the following was done: Each line in a fragment, was mapped to a discretized line on the grid using "Bresenham's Line Algorithm" [21]. Each line on the grid, contributed to two units of intensity incrementation on each point residing on it; also for the two parallel adjacent discretized lines to the previous, one additional unit of heat was introduced on each point residing on them. We name this method "Radial Line Heat Application".

Route Graph Extraction. Given the trace-based heat map as input, we introduce an algorithm to extract the underlying transportation network upon which the Twitter users are moving. The algorithm proceeds in a greedy manner, identifying potential edges which contain the highest local intensity of traversal by Tweet traces. Intuitively, this corresponds to the lines of red in Figure 1 on page 146. These identified edges are initially short, and through an iterative procedure, they are extended along directions with high Tweet traversal. A few additional tricks are needed to prevent an excess of edges being identified in regions where there is intense Tweeting spread out over a wide area, such as preventing the discovery of new additional edges that are nearly identical to previously identified edges. The pseudo-code is provided in Algorithm 1.

The algorithm keeps track of areas that it has already searched by setting map cells as being "engaged" once an edge has been found nearby. Initially, all cells are set as "disengaged". The algorithm then follows an iterative procedure in which, for every iteration, the following procedure is executed: A grid cell is chosen which has the highest amount of heat among all the "disengaged cells", and a new vertex defined on that location is added to the graph. Every disengaged grid cell within a radius of $searchRadius = 15$ cells is considered as a candidate

for the next vertex to add to the graph, along with the edge that connects the two. Each of these candidates are scored by summing the amount of Tweet traversal intensity along the edge. The grid cell candidate with the highest score determines the location of next vertex and is added to the graph. The edge connecting the previous two vertices is also added to the graph.

Next, the algorithm attempts to extend the new edge in both directions. On each direction, similar to before, all disengaged grid cells within a radius of $searchRadius = 15$ cells are considered as candidates for the next vertex to add to the graph, but this time, vertices that would result in an edge extension with an angular deviation of more than 15 degrees are disregarded. This restriction is intended to ensure that the path being formed corresponds to a single road on the map. If the highest scored vertex has an average heat of more than $thresholdRatio = 0.8$ times the average Tweet traversal intensity of the edge to be extended, it will be added to the graph along with its corresponding vertex, otherwise extension in this direction will reach cessation. During each attempt of extension, if an existing vertex is found within the search radius, and this vertex has an edge which forms an angle of less than 15 degrees with the edge to be extended, the two edges are connected and the process stops. This also prevents having redundant edges denoting essentially the same path. After path extension in both directions is complete, all grid cells within a radius of $engagingRadius = 10$ from the new edge is flagged as "engaged".

Patterns of Life. In the previous section, we use the aggregated data of all the users' activity traces to infer the underlying transportation network which guided the trajectories of the users. Here, we demonstrate a simple clustering technique to understand the different classes of user behavior. First, we apply K-means clustering on the dataset containing all the coordinates of each Tweet in our dataset. This results in clusters representing broad geographic areas where Tweeting activity occurs. We use Dunn indexing to choose an appropriate K. Given these geographic regions, we then create an activity vector v for each user:

$$v = [v_1^1 v_1^2 .. v_1^K v_2^1 v_2^2 .. v_2^K .. v_7^1 v_7^2 v_7^K],$$

where v_j^i is the Tweeting activity level for this user on the jth day of the week in geographic region i. This activity vector is normalized so that Tweeting activity sums to one for each day of the week. We then apply a second K-means clustering to this new set of vectors.

4 Data

Our dataset is extracted from Twitter, a popular micro-blogging service. In the Twitter terminology, the microblog messages or "tweets", are equipped with the option of containing what is referred to as "geotags". Geotags are labels that indicate where a Twitter user was, when the tweet was posted. On a client's side, a geotag can be applied by activating the geotag functionality in the settings of the twitter application being utilized.

Algorithm 1. Graph Route Extraction given *heatmapGrid*

{*heatmapGrid* is assumed to store heat values assigned to each grid cell}
{Graph G is initially empty and *gridCellFlags* is a grid of booleans initially set to false .}
{The distance between two gridpoints is the number of cells along the line connecting them with Bresenham's algorithm, which is equal to their Chebyshev distance}
$searchRadius \leftarrow 15$
$engagingRadius \leftarrow 10$
$thresholdRatio \leftarrow 0.8$
for $i = 1 \rightarrow numberOfIterations$ **do**
  $gridPoint_0 \leftarrow$ gridpoint with most heat value among disengaged gridpoints
  Add a new vertex v_0 to G with location defined as $gridPoint_0$'s location
  for every $gridPoint$ within a radius of $searchRadius$ from $gridPoint_0$ **do**
    $sum \leftarrow 0$
    for every $midGridPoint$ that appears along the line connecting $gridPoint$ and $gridPoint_0$ **do**
      {The line connecting two grid points is determined with Bresenham's Line Algorithm}
      $sum \leftarrow sum + heatmapGrid$[XOf($midGridPoint$)][YOf($midGridPoint$)]
    Assign sum as the score for $gridPoint$
  $pathEdges \leftarrow \{\}$
  $gridPoint_1 \leftarrow gridPoint$ with the highest score
  Add a new vertex v_1 to G with location defined as $gridPoint_1$'s location
  Add a new edge $e = \{v_0, v_1\}$ to G
  $addetopathEdges$
  $directions \leftarrow \{(gridPoint_0, gridPoint_1), (gridPoint_1, gridPoint_0)\}$
  while !isEmptydirections$direction$ **do**
    $direction = /textremoveFirstElement(directions)$
    $extendInDirection(direction)$
  for every edge e in $pathEdges$ **do**
    for every $midGridPoint$ that appears along e **do**
      for every $gridPoint$ within a radius of $engagingRadius$ from $midGridPoint$ **do**
        $gridCellFlags$[XOf($gridPoint$)][YOf($gridPoint$)] $\leftarrow true$

A tweet logged would contain exact latitude and longitude coordinates if and only if the tweet was sent through a smartphone (or any hand-held GPS-equipped device with the geotagging functionality switched on); Otherwise a tweet will include more general geotags like "Santa Monica" or "Marina Del Rey"; or perhaps lack any type of geotag whatsoever. One of the services Twitter's Streaming API provides, is live-streaming all tweets originating from a predetermined coordinate range (known as the "filter" service). This implies all such returned tweets will have non-empty geotag fields. Among retrieved tweets, those without latitude and longitude coordinates (roughly half of them) were discarded. In this paper, we focus on London, Los Angeles and Tokyo. From September 18, 2011 to February 9, 2012, we collected 22,496,299 tweets geotagged with latitude and longitude information.

Algorithm 2. Path Extension given *direction*

```
head = directionHead(direction)
tail = directionTail(direction)
threshold = thresholdRatio × averagePathHeat(head, tail)
for every gridPoint within a radius of searchRadius from head do
  angleDeviation = LineAngleDifference(line(head, tail), line(head, gridPoint))
  if angleDeviation/leqπ/12 then
    connectToVertex ← false
    verticesFound ← {}
    if hasVertex(gridPoint) then
      for every edge link originating from gridPoint do
        angleDeviation = LineAngleDifference(line(head, tail), link)
        if angleDeviation/leqπ/12 then
          connectToVertex ← false
          add gridPoint to verticesFound
    if isEmpty(verticesFound) then
      sum ← 0
      cellCount ← 0
      for every midGridPoint that appears along the line connecting gridPoint
      and head do
        sum ← sum + heatmapGrid[XOf(midGridPoint)][YOf(midGridPoint)]
        cellCount ← cellCount + 1
      Assign sum/cellCount as the average score for gridPoint
if !isEmpty(verticesFound) then
  v_con ← nearest vertex to head in verticesFound
  Add a new edge e = {v_con, head} to G
  addetopathEdges
else
  gridPoint_con ← gridPoint with the highest score
  if score(gridPoint_con) ≥ threshold then
    Add a new vertex v_con to G with location defined as gridPoint_con's location
    Add a new edge e = {v_con, head} to G
    addetopathEdges
    add direction {head, v_con} to directions
```

Data Processing. Among the many users tweeting via their phones, the most useful are those who tweet frequently throughout the day. By having the tweets for a such user, one can attempt to extrapolate the person's location throughout the day. We define a *trace* to be all tweets for one user starting at 4:00 AM of a day and ending at 3:59 AM of the following day. Empirically, it was the time of lowest activity in all the cities we were analyzing. In order to gauge the value of a particular trace of a user's daily activity, we use a "Pulsating Heuristic". Each time a person tweets it "pulsates", i.e., a radius of 400 meters (which is roughly the GPS error in our tweeting data which was calculated empirically) is "affected" by a pulse sent out from the tweet location. The lifespan of this pulse effect is set to half an hour. Once a person tweets again, the effect of previous pulses will be nullified. Now for evaluating a tweet and assigning a value to its

utility, starting from a score of zero, each time a person tweets from a non-effected location, the score is incremented. This heuristic avoids giving weight to users that frequently tweet but always at the same location. Instead it gives more weight to users that frequently tweet from different locations, even if the number of distinct locations is small.

Bot Filtering. Many geotagged tweets in our dataset were sent by bots sending location-specific news or advertisements. Although one might assume bots to be stationary (resulting in all tweets originating from a fixed location), certain services have multiple stations which all send information through the same alias. Traffic and incident reporting services and dining-venue-advertising functions would best exemplify such users. We use two heuristics to filter out most of these messages: (1) if a user is tweeting URLs (identified by finding the substring "http" in the tweet) more than 70% of the time, it is likely to be a bot, and (2) if a user's movement speed is more than 120 km/h, it is likely to be a bot (speed being calculated as the straight line distance between two consecutive tweeting events).

Trace Fragmentation. We next process the data by segmenting each day's trace into one or more fragments that represent a period of continuous user movement. For every trace, after displaying a point on the map representing the location of a tweet occurrence, the assigned "trajectory" to each trace was defined as the jagged line created by connecting each tweet's corresponding point, to the point of its next tweet. The idea was to extrapolate one's daily trajectory. One issue is that drawing a direct line between two points is too poor of an estimation, unless the two occurrences are fairly close in time. Therefore given a trace, we define a notion of a "fragment" as a maximal sub-path in a trajectory which for every two consecutive tweet occurrences, the following two conditions hold: (1) the time between the two is less than 30 minutes, and (2) the location distance is more than 400 meters apart. The former condition is enforced to reduce extrapolation error, and the latter to avoid scenarios in which the observed displacement is merely a result of GPS error. The notion of trajectory fragments is the basis of the work in this paper.

Location and Area Discretization. In order to facilitate the process of HeatMap construction, the area of study was sliced into a grid where each cell covers a 0.0005 degrees of latitude by 0.0005 degrees of longitude surface (being roughly equivalent to a 50 by 50 meter coverage). After executing necessary heat application calculations (stated earlier), for visualization purposes, this grid was in turn "flattened" using the Mercator method.

5 Results

Transport Network Inference. When comparing cities it seems that people in Tokyo make much more use of the public transportation, and this is characterized by seeing heat on particular routes in the Trace HeatMap that have blobs of red in the Point HeatMap for Tokyo. Such paths were actually bus or subway

routes and the blobs of red corresponded to the location of stops or stations; and example of which is shown on Figure 3.

By comparing the visualization results of our "Point-Based HeatMap" and the "Trace-Based HeatMap" displayed in Figure 1, clearly the underlying transportation routes are significantly more visible in the latter. Heat is sometimes seen to be relatively higher on portions of routes in the Point heatmap due to the fact that naturally, the more time people spend in an area, the more likely it is to have tweet activity; of course time spent in routes for each individual may not be high, but in a large scale, routes are densely populated.

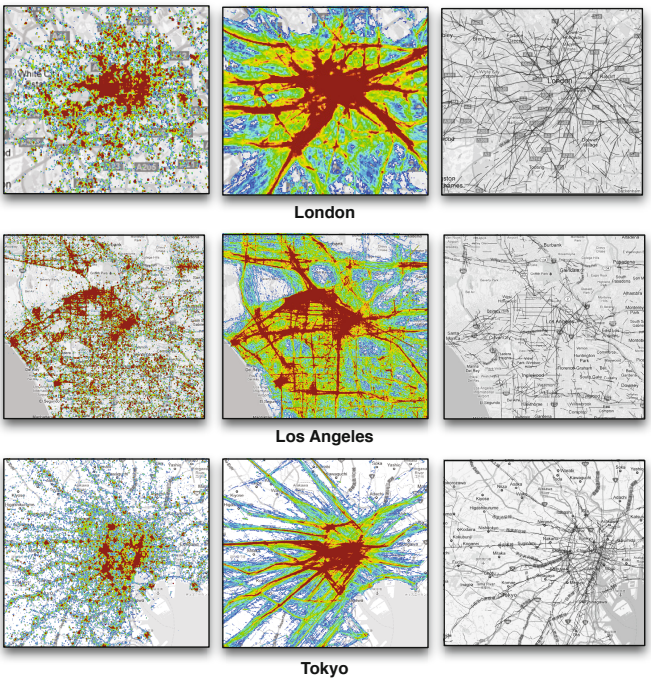

Fig. 1. Point-Based HeatMap, Trace-Based HeatMap and Route Graph of London, Los Angeles and Tokyo

We also managed to identify area-specific phenomena through studying the visualizations. Many of the main routes are nicely highlighted, as shown in the Figure 2, the 405 freeway is virtually inactive; which shows that tweeting on the 405 is not a common practice.

For the graph construction algorithm, an example of its functionality can be seen in Figure 4 which parts of Sunset Boulevard and West Hollywood in Los Angeles are identified.

Patterns of Weekly Activity. No surprises were uncovered in our analysis of weekly activity patterns. Our algorithm reported nine broad geographic areas of

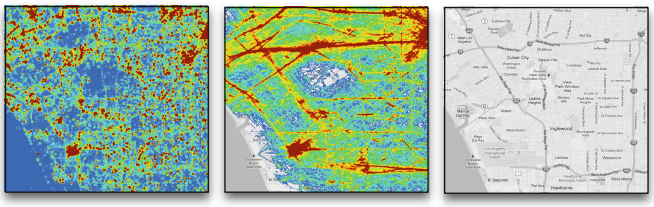

Fig. 2. Point-Based HeatMap, Trace-Based HeatMap and Map of Los Angeles Showing the 405 Freeway Not Properly Delineated in Visualization

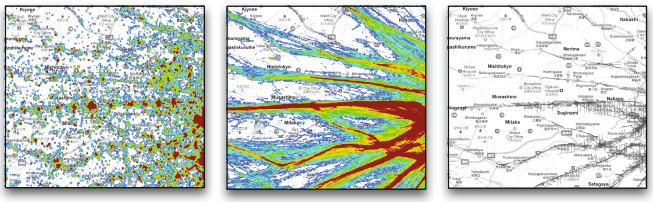

Fig. 3. Point-Based HeatMap, Trace-Based HeatMap and Route Graph of Tokyo, Uncovering the Existence of a Public Transportation Route

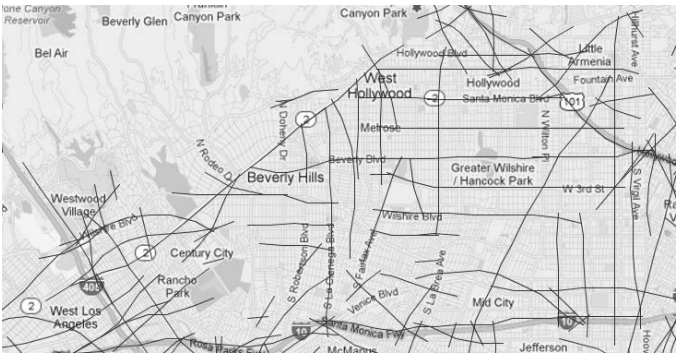

Fig. 4. An example of the Graph Route Construction Algorithm's Efficacy

tweeting activity within the Los Angeles dataset. User behavior was clustered into eight classes, with each of these classes roughly corresponding to users whose activity was mainly centered in one of the broad geographic areas. The results we got by applying our Algorithm (to find pareiodic patterns in movement of people based on K reference points) to our Los Angeles Data set were not surprising. There was some differentiation between activity on weekdays and weekends, with users exhibited slightly higher entropy in terms of their distribution of activity over the nine geographic regions. This makes intuitive sense, since people tend to travel more for leisure on weekends.

6 Conclusions and Future Work

This paper describes the visualization, analysis, and mining of location-based data generated by mobile agents. Here, our analysis focused on geo-tagged Tweets. However, in the future, we hope to apply these techniques to other agent-generated trajectory data, such as agents in location-based games, UAVs, or indeed Twitter bots. Discovery of spatio- temporal patterns in such data is an important and challenging problem, and here we have only presented an initial step in this direction.

References

1. Cao, L., Gorodetsky, V., Mitkas, P.: Agent mining: The synergy of agents and data mining. IEEE Intelligent Systems 24(3), 64–72 (2009)
2. Cao, L.: Data mining and multi-agent integration. Springer, Dordrecht (2009)
3. Noulas, A., Scellato, S., Mascolo, C., Pontil, M.: An empirical study of geographic user activity patterns in foursquare. In: Proc. of the 5th Int'l AAAI Conference on Weblogs and Social Media, pp. 570–573 (2011)
4. Song, C., Qu, Z., Blumm, N., Barabási, A.L.: Limits of predictability in human mobility. Science 327(5968), 1018–1021 (2010)
5. Azevedo, T.S., Bezerra, R.L., Campos, C.A.V., de Moraes, L.F.M.: An analysis of human mobility using real traces. In: Proceedings of the 2009 IEEE Conference on Wireless Communications & Networking Conference, WCNC 2009, pp. 2390–2395. IEEE Press, Piscataway (2009)
6. Candia, J., Gonzalez, M.C., Wang, P., Schoenharl, T., Madey, G., Barabasi, A.L.: Uncovering individual and collective human dynamics from mobile phone records. Math. Theor. 41, 224015 (2008)
7. Palma, A.T., Bogorny, V., Kuijpers, B., Alvares, L.O.: A clustering-based approach for discovering interesting places in trajectories. In: Proceedings of the, ACM Symposium on Applied Computing, SAC 2008, 863–868. ACM, New York (2008)
8. Chen, Z., Shen, H.T., Zhou, X.: Discovering popular routes from trajectories. In: Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, ICDE 2011, pp. 900–911. IEEE Computer Society, Washington, DC (2011)
9. Masciari, E.: A Framework for Trajectory Clustering. In: Trigoni, N., Markham, A., Nawaz, S. (eds.) GSN 2009. LNCS, vol. 5659, pp. 102–111. Springer, Heidelberg (2009)
10. Vieira, M.R., Bakalov, P., Tsotras, V.J.: On-line discovery of flock patterns in spatio-temporal data. In: Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS 2009, pp. 286–295. ACM, New York (2009)
11. Liao, L., Patterson, D.J., Fox, D., Kautz, H.: Learning and inferring transportation routines. Artif. Intell. 171(5-6), 311–331 (2007)
12. Sevtsuk, A., Ratti, C.: Does urban mobility have a daily routine? learning from the aggregate data of mobile networks. Journal of Urban Technology 17(1), 41–60 (2010)
13. Colizza, V., Barrat, A., Barthelemy, M., Valleron, A.J., Vespignani, A.: Modeling the worldwide spread of pandemic influenza: Baseline case and containment interventions. PLOS Med. 4, e13 (2007)

14. Li, Z., Ji, M., Lee, J.G., Tang, L.A., Yu, Y., Han, J., Kays, R.: Movemine: mining moving object databases (2010)
15. Andrienko, G., Andrienko, N., Wrobel, S.: Visual analytics tools for analysis of movement data. SIGKDD Explor. Newsl. 9(2), 38–46 (2007)
16. Vasiliev, I.R.: Mapping Time. Cartographica 34(2) (1997)
17. Kapler, T., Wright, W.: Geo time information visualization. Information Visualization 4(2), 136–146 (2005)
18. Gennady, A., Natalia, A.: A general framework for using aggregation in visual exploration of movement data. Cartographic Journal 47(1), 22–40 (2010)
19. Laube, P., Imfeld, S., Weibel, R.: Discovering relative motion patterns in groups of moving point objects. International Journal of Geographical Information Science 19, 639–668 (2005)
20. Skupin, A., Hagelman, R.: Visualizing demographic trajectories with self-organizing maps. Geoinformatica 9(2), 159–179 (2005)
21. Bresenham, J.E.: Algorithm for computer control of a digital plotter, pp. 1–6. ACM, New York (1998)