CONTENTS

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

ETS	École de Technologie Supérieure
GPS	Global Positioning System
RFID	Radio-Frequency Identification
API	Application Program Interface
NSERC	Natural Sciences and Engineering Research Council
RDU	Raleigh-Durham International Airport
LGA	LaGuardia Airport
XML	Extensible Markup language
PLT	AutoCAD Plot

INTRODUCTION

As the use of modern devices capable of stream data is becoming more prevelant in all aspects of life, enormous amounts of digital data are being recorded. For example, everyday use of devices such as smart-phones, GPS receivers and RFID tags has resulted in the collection of massive amounts of Movement or GPS[1] data. Increased availability of such large movement datasets has led to much research in development of analytical and visualization techniques [Rinzivillo *et al.* (2008); Wood *et al.* (2010); Andrienko and Andrienko (2011)]. Analysis of such datasets has the potential to provide insights into the data by discovering patterns in the motion of objects, finding outlier activities or discovering relationships between objects. These patterns can be useful in a number of ways, for example, recommending places to visit based on a user's past and current positions, or police departments monitoring suspicious activity patterns [Zheng *et al.* (2010b); Yan *et al.* (2007)]. Visualization lies at the forefront of exploratory analysis of such datasets.

Although useful, visualizing such data is challenging for a few reasons:

- There are multiple variables involved: *latitude* and *longitude* as a function of *time* and *object id*, where object id identifies a person, vehicle, or other moving entity.
- Movements often cross each other and may repeatedly travel other the same pathways to the same locations, causing occlusion (overplotting) when drawn on a 2D geographic map.
- Movements may occur at widely varying physical scales, e.g., ranging from tens of meters (moving between two buildings) to hundreds of kilometers (traveling between cities) in the case of a single GPS dataset, and zooming out on a 2D map will leave only the largest-scale movements salient.
- The data may cover long spans of time containing thousands of events.

[1] Movement data is defined as a sequence of time-ordered geographical coordinates (latitude and longitude).

Many previous approaches to visualizing movement data have involved the display or analysis of the *shapes* of movement trajectories [Kapler and Wright (2004); Crnovrsanin *et al.* (2009); Hurter *et al.* (2009); Guo *et al.* (2011)], or have proposed ways of dealing with large numbers of moving objects [Andrienko and Andrienko (2011); Zeng *et al.* (2013)], or both [Liu *et al.* (2011); Buchin *et al.* (2013); Krüger *et al.* (2013); Wang *et al.* (2013)], often making use of aggregation [Elmqvist and Fekete (2010)].

In the current work, we are instead interested in understanding a small number (< 20) of moving objects. In such a case, we would like to avoid aggregating groups of objects, so that the history of each entity is visible. At the same time, we are *not* interested in the detailed shape of movement trajectories. Instead, we are interested in the *discrete locations* (e.g., rooms, workstations in a factory, buildings, addresses, cities) visited by the moving entities. We are interested in *where* the people or objects have been (in terms of discrete locations), *when* they were there, *how many times* they visited different locations, and *in what order*. We are also interested in *meetings* that occur between the objects or persons. Such information is difficult to convey in a single 2D geographic map, due to overplotting or the need for animation.

There are several scenarios where we may wish to understand the movements of a small number of objects or persons over a set of discrete locations. These include analyzing meetings and activities of suspected criminals (gang members, terrorists) whose cell phones are tracked; monitoring offenders on probation; understanding movements of a team of workers and their equipment in a factory, to improve workflow processes; analyzing movements of groups of visitors in a museum; understanding movements of health care professionals, patients, and equipment within a hospital, to optimize changes to floorplans or reduce the risk of pathogens spreading; or movements of several customers within a large store, to optimize merchandise displays and layout. Such scenarios may also arise if the user begins with an overview of a large, aggregated dataset, and selects a small subset of moving entities to analyze in more detail.

In these situations with a small number of moving objects and no need to understand the shapes of trajectories, we can discretize space to simplify the visualization. Such discretizations are leveraged in previous work [Andrienko and Andrienko (2011); Thudt *et al.* (2013); Ventura and McGuffin (2014)] to simplify 2D geographic maps, but these approaches either require aggregation of sets of moving entities, or else may still suffer from overplotting. In the current work, we instead *list* the discretized locations as rows in a kind of Gantt chart. In the past, Gantt charts have been used for schedules of activities [Gantt (1919); Plaisant *et al.* (1996); Tory *et al.* (2013); Gigantt (2014); Jo *et al.* (2014)], software execution threads [Trümper *et al.* (2010); Isaacs *et al.* (2014)], and dynamic graphs [Hlawatsch *et al.* (2014)], but Gantt charts have received little attention for visualizing movement data (the only example we know of is Andrienko et al. [Andrienko *et al.* (2008)], who use the vertical axis in a Gantt chart to list different moving objects, but not to list locations). Compared to other visualizations of movement data, Gantt charts have the advantage of offering a single 2D view, free of occlusion, displaying the data without the need to aggregate groups of moving objects, and easily handle movements between different locations at widely different scales. We demonstrate that Gantt charts support several analytical tasks for movement data, and contribute a design study that goes beyond previous literature by identifying multiple ways to display *both* locations *and* moving entities within a Gantt chart, as well as ways to highlight meetings of moving entities. We also apply techniques to make Gantt charts more scalable vertically (by allowing rows to be filtered and sorted) and horizontally (by allowing the user to fold empty spans of time).

The contributions of this thesis are:

- The identification of tasks relevant for understanding the movements and meetings of small groups of entities (Section 1.2.1).
- A taxonomy of ways of visualizing movement data (Section 1.2.2).
- A discussion of design choices around Gantt charts for movement data, analyzing ways to show two variables (person and location) as a function of time, and ways to show meetings (Section 2.2).

- A prototype, called MovmentSlicer, with several features for visualizing movement data (Chapter 3).
- Case studies to illustrate the use of the prototype (Section 3.4).

The prototype uses color, and an explicitly drawn link, and animation to link together multiple coordinated views[2]. It also allows for filtering through an adjacency matrix of meetings between people; has a Gantt chart that can show people within places or places within people as a function of time; allows sorting, filtering, and annotation of rows in the Gantt chart; enables multi-focal "folding" to allow the user to see details of meetings and events with intervening empty time periods compressed; and uses smoothly animated transitions when folding or unfolding time. Thus our prototype enables users to understand the movement patterns of a small number of objects and the interaction that occur between them.

This thesis is divided into 3 chapters. Chapter 1 presents a taxonomy of tasks for movement data, a taxonomy of meeting-oriented tasks, visualization of movement data, and literature relevant to this. Chapter 2 explains the research objectives of the thesis, the methodology used (Section 2.1) and provides an overview of the design decisions (Section 2.2), while the last chapter 3 presents our prototype and case studies.

[2] Coordinated views are simultaneously displayed viewports or windows that show the same data in different ways, and that display connections or relationships across views using some other form of coordination, such as simultaneous highlighting in response to hover (Roberts (2007)).

CHAPTER 1

LITERATURE REVIEW

In this chapter, we introduce the importance of visualization in movement data analysis, followed by identification of tasks related to movement data with a focus on interaction and group behaviors. Following this we present a novel taxonomy for categorization of movement data visualizations. Finally, we discuss research related to visualization of geo-spatial data.

1.1 Introduction

Devices capable of recording data are entering into every aspect of life. Modern devices can record data ranging from your location history (e.g., GPS) to your sleep cycles (e.g., FitBit), respectively. This data can be leveraged to find patterns and meaningful relations or trends. Such analysis is helpful to both the user and the industry in making informed decisions. For example, insights from movement data can be useful in making better recommendations for travelling (Zheng *et al.* (2010b)) or suspicious activity detection (Yan *et al.* (2007); Guo *et al.* (2011)).

Analyzing such immense amounts of movement data has its own problems. Raw movement data consists of long sequences of time-stamped coordinates. Limitations of human working memory make it extremely difficult to find significant patterns in such large datasets (Baddeley (1992)). Computers are capable of remembering data indefinitely and thus, are efficient at finding trends in the dataset. However, just using algorithms to discover patterns or outliers can only give a superficial amount of information about the dataset. To perform insightful decisions based on patterns, human reasoning is required. Heer and Shneiderman (2012) refers to this logic-based reasoning as domain-specific knowledge. Hollands *et al.* (2004) states that information visualization[1] helps improve human perception, allowing it to work with larger datasets and improving reasoning capacity. Visualization also offers the advantage of allowing

[1]Information visualization is the presentation of data using information graphics so as to improve human cognition about data.

presentation of data from different perspectives and interactivity that further increases humans' ability to understand the data. For example, if you consider geo-spatial data, plotting the data on a map may help infer regional relations where as animating on a timeline allows for discovering temporal trends.

Visualization can be designed in two ways. Either the user starts with a task or a hypothesis in mind and designs a visualization that assists him or her in supporting the specific tasks using the data, or, the user starts with a way to visualize the data and finds patterns that help him or her formulate hypothesis. In both cases, whichever one precedes the other, the two parts of task formation and visualization design are equally important for providing meaningful insights into the data. In this thesis, we follow the former path. We suggest a handful of tasks that we aim to support and then design visualizations to facilitate performing those tasks.

The rest of this literature review is organized as follows: in Section 1.2, we survey existing taxonomies of tasks and present a simple but novel taxonomy of visualizations relating to geo-spatial data. Section 1.3 gives a brief review of the existing work relating to visualization of movement data.

1.2 Taxonomy

1.2.1 Taxonomy of tasks

Tasks are fundamental for effective visual analysis. As mentioned earlier for this thesis, we first define tasks which we then use to guide the process of visualization design. Defining tasks for movement data is complex due to: involvement of multiple variables such as space, time and object id's; varying physical scale, i.e. data can be specific to one city or may span multiple countries; and the granularity of time may also vary, people might want to understand data over a month or many years.

A taxonomy helps alleviate the problem. Dodge *et al.* (2008) states several advantages of defining a taxonomy: accurate task definitions and categorization help in designing focused visualizations, and it aid designers in generalizing visualizations across tasks. Also accurate definitions help in evaluating the visualization during user studies.

Bertin (1967) uses task types and reading levels to define a framework for task typology. He defines task types according to the variables in the dataset. For example, in case of GPS data the variables are time, location and object id. Examples of possible tasks are: Given a particular time, what is the location of an object?; Given an object and it's location when was it there?. For each task type, Bertin further defines three levels of reading: elementary (single object), intermediate (multiple objects) and overall (all objects). While Bertin talks about data in general, Peuquet (1994) describes task types specifically for spatio-temporal data. She states that there are three basic kinds of tasks you can perform on a given movement dataset:

- when + where → what : gives information about which object or group of objects were present at a given location or set of locations at any given time or set of times.
- when + what → where : gives information about the location or set of locations occupied by an object or a group of objects at a given time or set of times.
- where + what → when : gives information the time or set of times at which a given object or group of objects occupied a given location or set of locations.

Andrienko *et al.* (2003) uses a mixture of Bertin's and Peuquet's theories with slight modifications. They merge the intermediate and overall levels by Bertin into a single level and call it the set level or the general level. They divide the type of tasks into two categories instead of three by Peuquet :

- when → what + where : gives information about an object or a set of objects and their location or set of locations at any given time or set of times.
- where + what → when : gives information about time or set of times given an object or a set of objects and their location or set of locations.

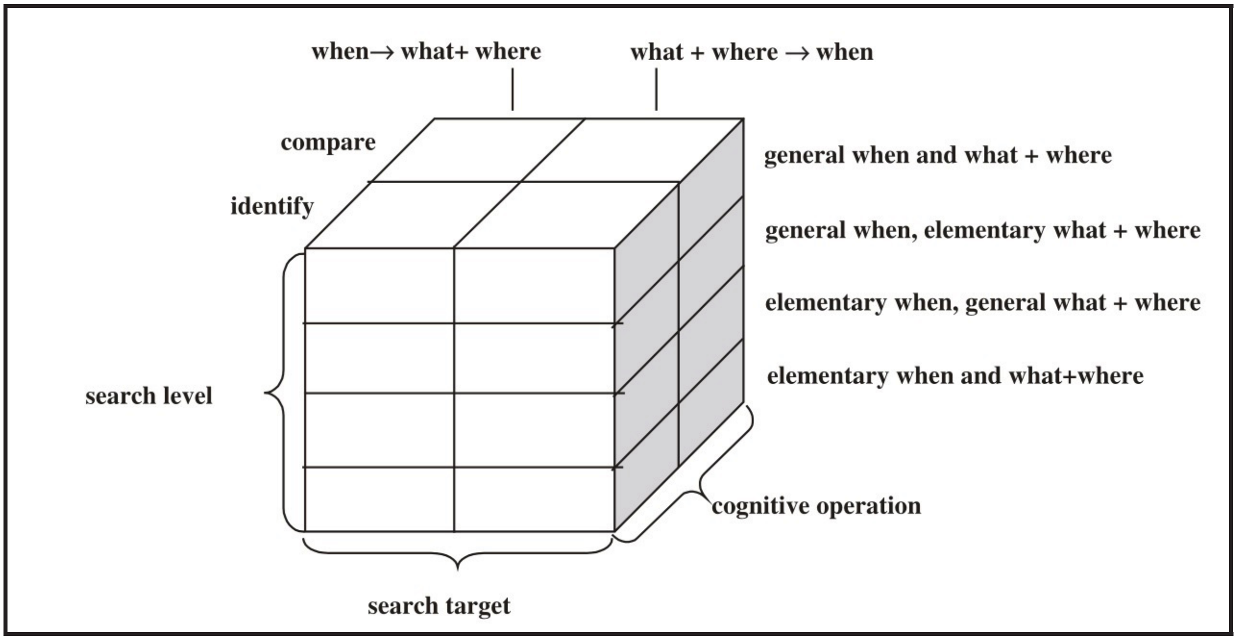

Figure 1.1 Task taxonomy.
Taken from Andrienko *et al.* (2003).

This simplifies the taxonomy without loosing the capability of accurately categorizing movement tasks. They rename the reading levels used by Bertin to search levels. Andrienko *et al.* (2003) also add a new dimension, "identification-comparison", to allow comparison and identification of relations between two or more elements. The final typology they suggest is as follows:

- elementary "when" and elementary "what + where": gives information about an object and it's location at any given time. Example question: Where was Alice at 6:00 pm on 1st January, 2015?
- elementary "when" and general "what + where": gives information about an object or a set of objects and their location or set of locations given a particular time. Example question: At 5:00 pm on 2nd February, 2015 where were Alice and Bob?
- general "when" and elementary "what + where": describes about the attribute of an object over a period of time. Example question: For how long did Alice visit Toronto?

- general “when” and general “what + where”: describes about the location or set of locations of an object or a set of objects over a time period. Example question : List all cities Alice and Bob have visited over the last year?

Elementary (with respect to time) tasks refer to individual time moments. They further provide a classification scheme, where these tasks can be differentiated on the basis of the search target (time or locations of objects), cognitive operation involved (identification or comparison of time or objects or regions) and search level with respect to space and objects as shown in Figure 1.1.

We present here an even simpler, but still useful, set of categories of tasks that are relevant and more focused for understanding the activities of small groups of moving entities.

Person-centric: These are questions about a particular person:

- “Where was Alice at a particular time?”
- “What are the most commonly visited places by Bob, and in what order are they usually visited?”
- “What are some rarely visited places by Carol?”

Location-centric: These are questions about a particular place:

- “Who was at the store on a particular day?”
- “Who most often visits the office building?”
- “Who rarely visits the back alley?”

Meeting-centric: Meetings, unlike people, have a limited duration in addition to a location. Questions about them include:

- “Who are the groups of people who meet together?”
- “How many times, when, and where did they meet?”
- “How many times, and when, did a group meet at a particular place?”
- “Who is usually earliest, or latest, for a group meeting?”

Time-cyclic: Some questions involve cycles of time, such as daily or weekly cycles. For example, if Alice goes to the office most weekdays, we might ask “What time does she usually arrive at the office?”, or “What days did she arrive unusually early or late?” These examples imply that time is broken up into two components: a time-of-day asked about in the first question, and a calendar day asked about in the second question. More generally, for some period P of time (where P could be 24 hours, or 7 days, etc.), we could break time into (time modulo P) and $\lfloor \text{time} / P \rfloor$, and ask questions about each of these components.

The above categories are not mutually exclusive. For example, the question “When was Dave at home?” is both person-centric and location-centric.

In considering design options, we have considered how well the above categories of questions are supported.

1.2.2 Taxonomy of visualization

In addition to providing a simple taxonomy on tasks relevant to movement of small group of objects, we also suggest a novel taxonomy of visualizations for movement data. From a visual perspective, without considering data aggregation, we note that many visualizations of movements fall into the 5 categories of Figure 1.2.

Each category has its own advantages and disadvantages.

The small multiples in Figure 1.2 (upper left) require more space (or must be shrunk down in size, thus reducing the spatial resolution of each snap shot).

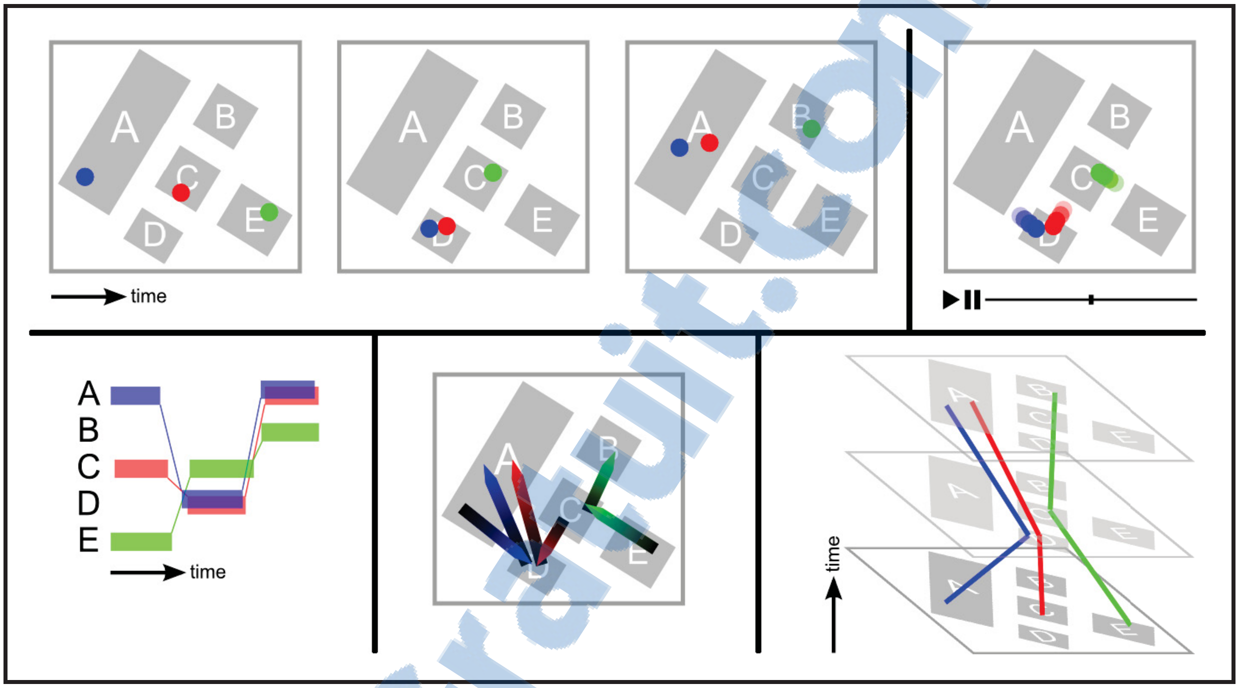

Figure 1.2 A taxonomy of visual representations of the same movement data. *Top left:* small multiples. *Top right:* animation with playback controls. *Bottom left:* Gantt chart. *Bottom middle:* static 2D geographic map with trajectories. *Bottom right:* 3D "space-time cube".

The 3D view (Figure 1.2, lower right) is sometimes called a "space-time cube" (Kraak (2003); Bach *et al.* (2014); Amini *et al.* (2015)) and has been used in a successful commercial product (Kapler and Wright (2004)). 3D views have the added advantage of third dimension. However, such views require 3D camera controls to rotate the visualization to different points of view.

Animated playback have shown to be more natural for understanding changes in movement pattern over time, but they require longer time to watch and understand. Static maps are better at showing the geographical regions involved but they suffer from problems of scaling and over-plotting. Also they are not multi-dimensional, i.e., they can only show the location aspect of movement data. Time has to be a derived dimension either by using 3D like Kapler and Wright (2004) or using glyphs such as arrows (Figure 1.2).

Gantt charts make it simpler to organize and compare time sequenced events. It's easier to see events and information like start time, end time and duration of stay. However, details about shape and the geographical context of the trajectories are lost. Despite this, the simplicity of the Gantt chart allows many questions related to time and space to be answered, as we will see later in Section 2.2.2.

1.3 Visualizations of movement data

One of the simplest techniques for visualization of movement data involves plotting all the points on a geographic map and connecting them using a simple polyline. However this method has several disadvantages (Figure 1.3):

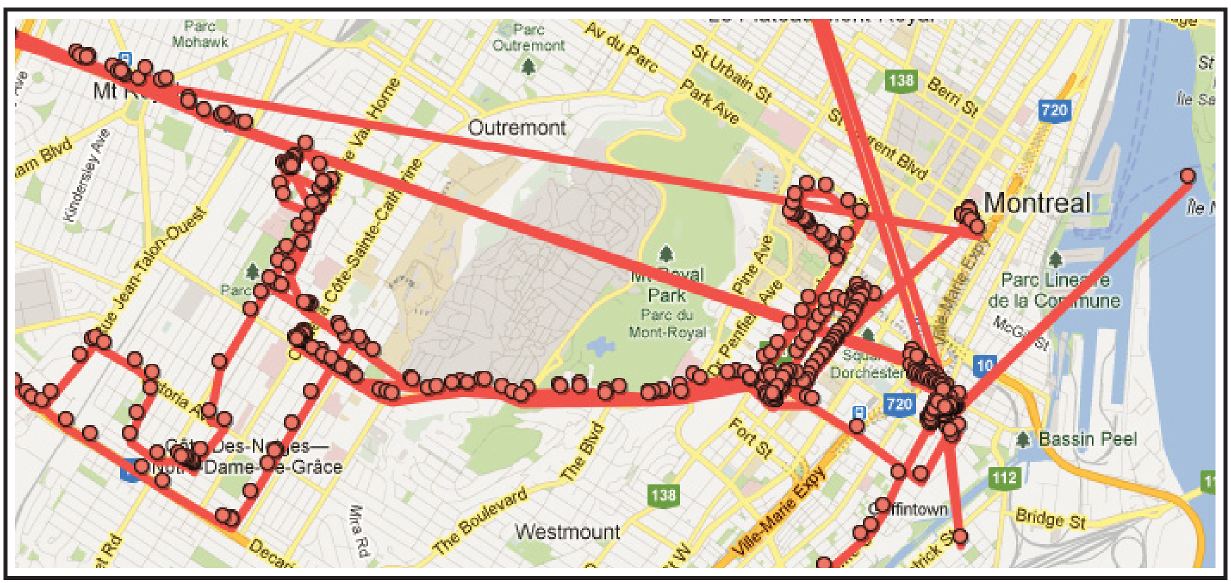

Figure 1.3 Movements of a smartphone over 7 days, as captured by the Backitude application for Android, sampling at 15 second intervals, and subsequently visualized with Google Latitude. Gaps and discontinuous jumps in the data are caused by traveling through tunnels and by occasional spurious errors. We clearly see even with this small period of 7 days there are multiple crossovers and occlusion.

- Only one dimension of a multi-dimensional dataset is visualized. Other dimensions need to be deduced by the user either using 3D, animation, glyphs or small multiples.

- Plotting large dataset causes occlusion and crossovers.
- Simultaneously plotting of a large number of points on a geographic map is a computationally costly operation, especially for web-based visualizations.
- The scale of the data in spatial terms can vary from being within city to between countries. Thus, zooming out on a map could leave only the cross country trajectories visible.

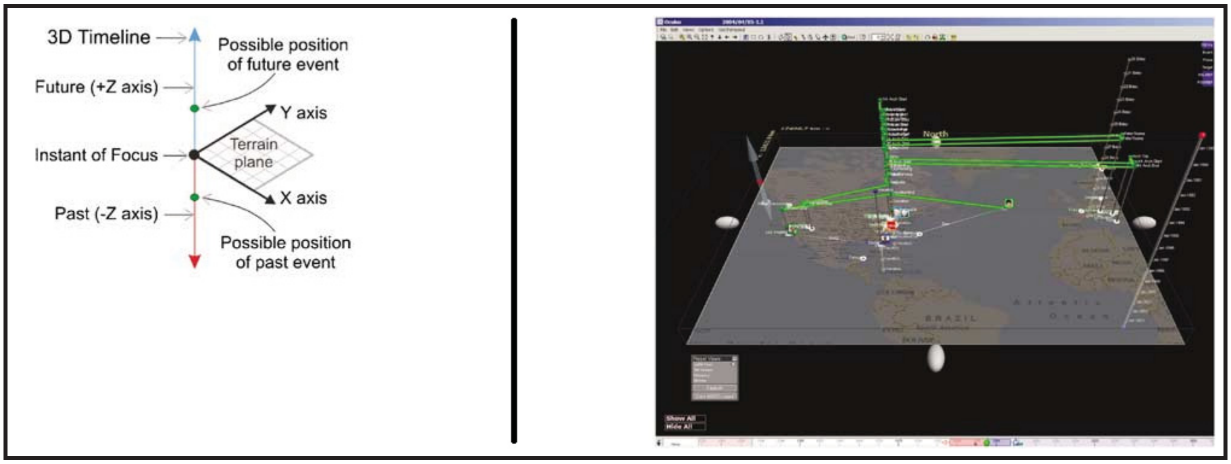

Figure 1.4 *Left:* shows the three dimensions, x-axis and y-axis represent geographic area, while the z-axis shows time. *Right:* the interface shows trajectory of a person plotted in 3D. Plotting time in the third dimension prevents crossover (Figure 1.3) of trajectories over time which would happened if trajectories are plotted on a 2D map. Taken from Kapler and Wright (2004).

One way of overcoming the dimensionality problem is using small multiples. The geographic map in the snapshots show the spatial dimension while the snapshots at different instances of time show the temporal dimension. But having multiple images of the same area is expensive in terms of screen space. Kapler and Wright (2004) propose an alternative way. They plot the geographic map as a 2D plane and plot time as a third dimension in 3D. The time increases upwards out of the plane while it decreases downwards into the plane (Figure 1.4 left). Thus all the future events occur above the geographic plane and all the past events occur below the geographic plane. This provides more detail and context when presenting movement data.

Events like meetings can also be seen by changing perspective. However, occlusion still exists if the dataset is large. In addition, the learning curve of using this 3D mapping is high.

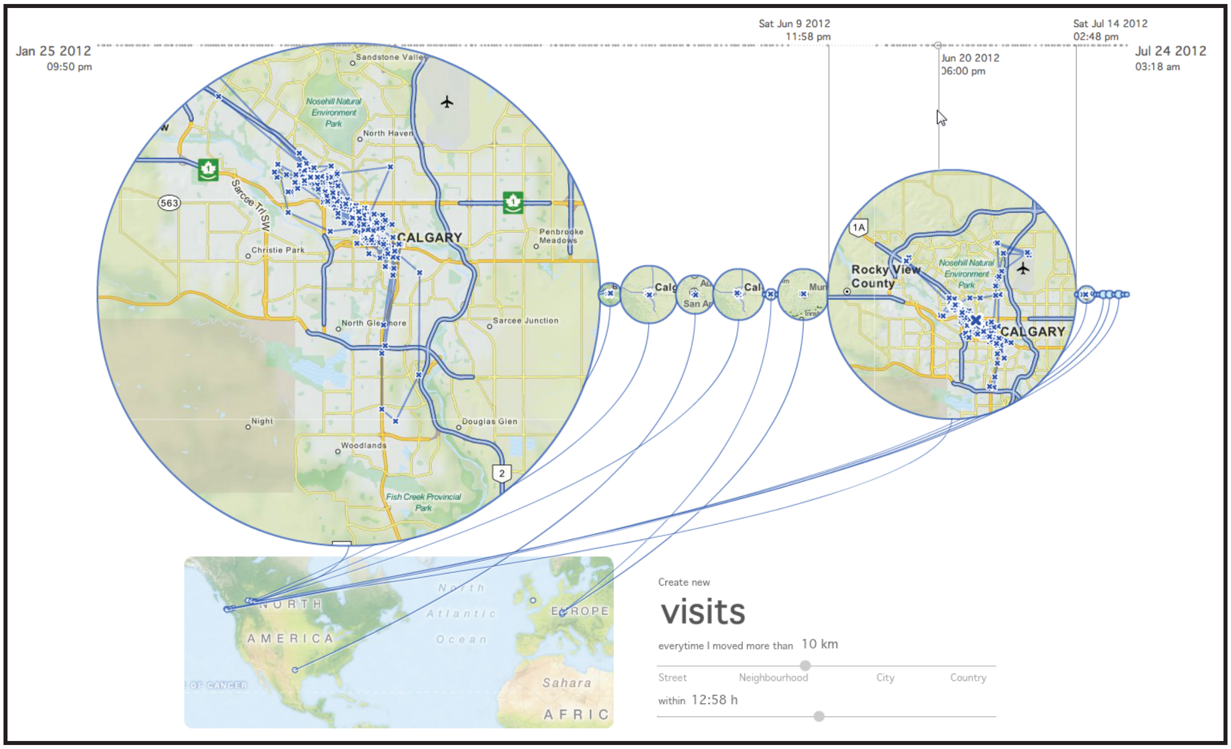

Figure 1.5 Map-timeline showing location history of six months. Circles represented places and the horizontal axis represents time.
Taken from Thudt *et al.* (2013).

Thudt *et al.* (2013) try to address the same dimensionality problem in a different way. They display location histories using a map-timeline (see Figure 1.5). They divide the movement data based on two parameters: places (which are independent of time and can vary in geographical scale from “a hotel" to “a country") and stays (which consists of a duration of time spent). So a stay can consists of multiple places. They arrange stays as horizontally arranged circles. This hybrid approach tries to include two dimensions simultaneously into a single dimension. The locations are represented as circles (size of the circles is proportional to the duration of the stay), and the arrangement shows the time dimension which increases along the axis. We use dimensional stacking, LeBlanc *et al.* (1990), in combination with dimensional clustering

(Section 3.1) to address the issue of dimensionality reduction instead of using length. This enables us to reduce the dimensionality further enabling us to focus on a group of people rather than just on a single individual.

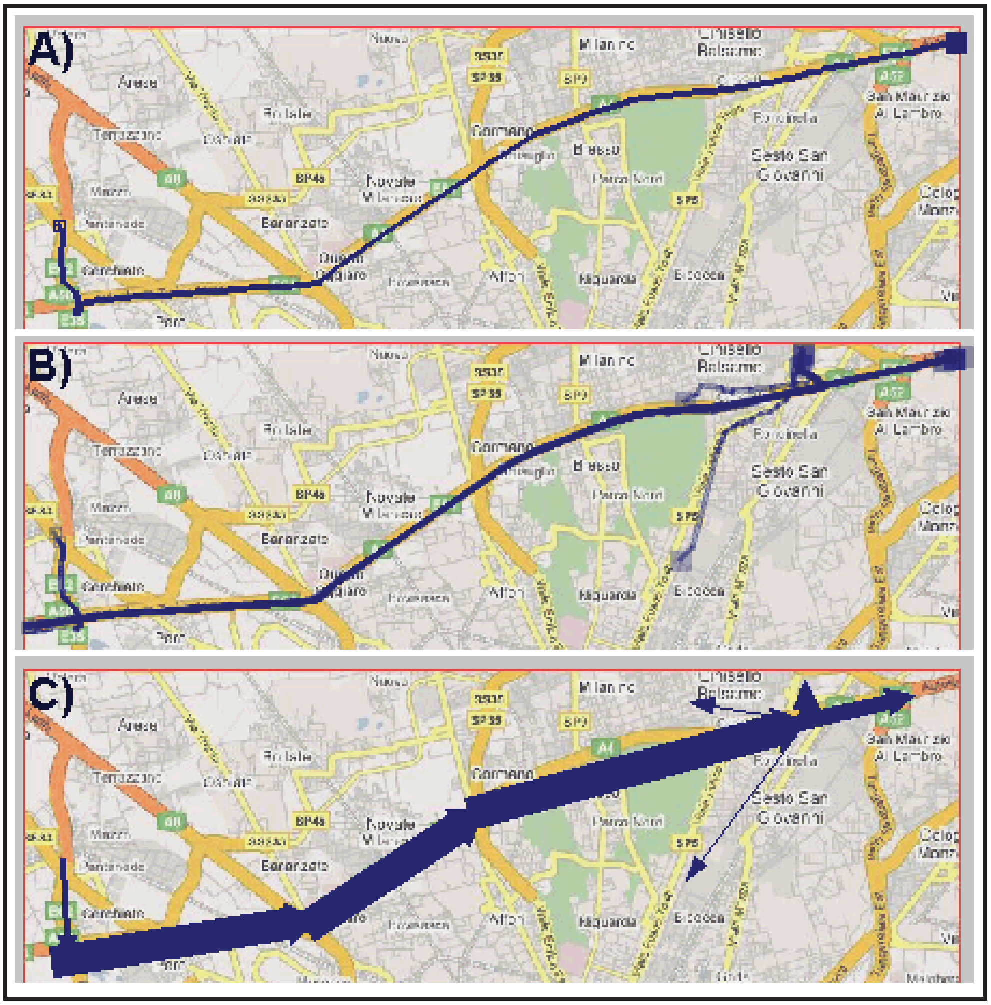

Figure 1.6 A) shows a single trajectory. B) shows similar trajectories clustered together. C) shows a summarized representation of cluster using flow arrows. It's easy to see that clustering helps prevent occlusion by grouping together similar trajectories. Taken from Rinzivillo *et al.* (2008).

Although the above methods help in addressing multivariate issues, problems of occlusion and overplotting still exists. Aggregation (or clustering) help alleviate this issue, however just clustering trajectories is not enough to make decisions. According to Rinzivillo *et al.* (2008), an essential phase for effective use of clustering is interpretation of the obtained cluster by humans who can reason and validate the usefulness of the cluster. Thus researchers extensively use clustering and visualization to enable such interpretation.

Many algorithms exist that try to summarize the dataset, providing an overview of the dataset using clustering and visualization. Trajectories or movement paths are usually clustered together to provide an overview, which enables users to identify areas or regions of interest. Rinzivillo *et al.* (2008) define the process of progressive clustering for this purpose. They designed simple functions which could be applied to entities as a group or whole to cluster them together. The functions can also be reapplied to a previously obtained cluster for further aggregation. Rinzivillo *et al.* (2008) visualize trajectories on a map by drawing polygonal lines connecting positions in time sequence. The starting position is marked by a small hollow square and the ending position by a larger filled square. However, when multiple trajectories are visualized in this way, the map becomes cluttered and unreadable. He uses progressive clustering to represent each group in a summarized way without drawing the individual trajectories.

Similarly, Adrienko and Adrienko (2011) define an aggregate model that clusters trajectories into movement flows. Their algorithm automatically detects points of interest from a given set of trajectories and partitions them based upon directional flows. The algorithm uses spatial coordinates to extract characteristic points from trajectories which in turn are used to generate a Voronoi diagram[2]. These cells are then used as a grid to divide the geographical areas into territories. Trajectories are then plotted as flows between these territories.

Wood *et al.* (2010) use treemaps to display visual trajectory summaries. He proposes mapping OD (origin and destination) vectors as cells rather than lines, comparable with the process of constructing OD matrices. But OD matrices generally has the problem that it loses spatial layout. To overcome this issue and preserve the spatial layout of all origin and destination

[2]Voronoi diagram is a partitioning of the plane into n regions as a function of n seed points. Each region is divided such that all points belonging to one region are closer to the seed point of that region than to the seed point of other regions.

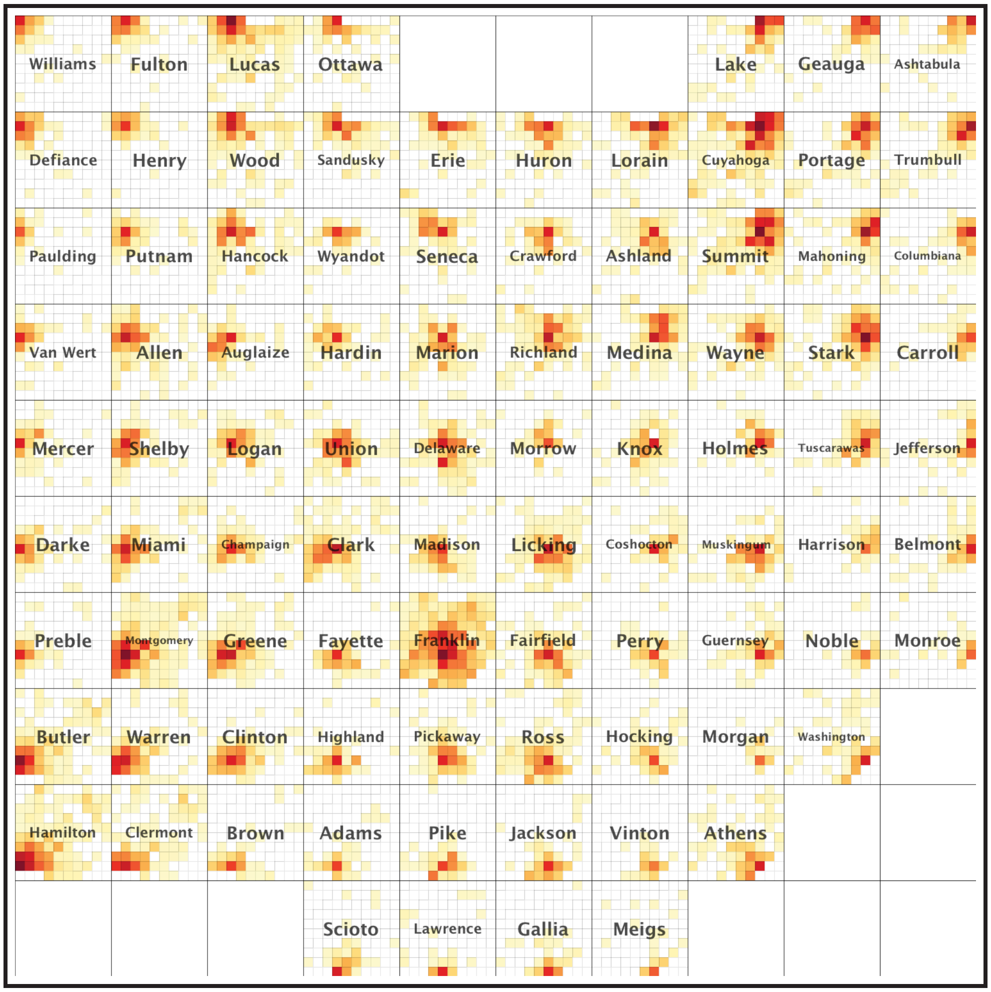

Figure 1.7 State of Ohio is divided in 10×10 grid. Each cell of the larger grid is a city and is considered as the origin city. Destination grid is a nested small multiple of the larger origin grid having the same spatial layout. Thus forming a two level treemap. Intensity of color in each cell denotes the density of trajectories from a particular larger origin cell to a smaller destination cell nested inside it.
Taken from Wood *et al.* (2010).

locations, Wood et al. construct a gridded two-level spatial treemap. The result is a set of spatially ordered small multiples upon which any arbitrary geographic data may be projected. Each trajectory can therefore be referenced by two grid cell locations : the grid cell in which the trajectory origin lies, and the grid cell in which its destination lies. By nesting the destination grid cells within the origin grid cells, they preserve the spatial relationships between origin cells and the relationships between destination cells in a single matrix.

Guo *et al.* (2011) interpret trajectories as multivariate data (e.g., computing each trajectory's maximum speed, average speed, angular range, etc.) and display them as tuples within a parallel coordinates plot that is also coordinated with other views. Other recent work (Andrienko and Andrienko (2011); Liu *et al.* (2011); Zeng *et al.* (2013); Krüger *et al.* (2013)) have proposed several more ways of aggregating movement data to show overviews. Unlike them, our work is not geared toward overviews of large numbers of objects, nor towards visualizing trajectories, but rather using aggregation to provide overview and details about the behaviors of an individual or a small groups of moving people/objects. Thus we use clustering to discretize trajectories into smaller places of interest called locations and then summarize the motion of people or objects based upon these locations. Furthermore, since we are not interested in mapping the shape of the trajectories, we chose Gantt charts that can enable us to answer a greater variety of movement data questions easily (Section 1.2.2).

The Gantt chart transforms position (latitude and longitude) into a single (vertical) axis, thereby preventing the user from judging distances between locations, and failing to show the shape of trajectories taken between locations. Gantt charts show at least one categorical (or ordinal) variable crossed with time, and were originally proposed for visualizing work schedules (Gantt (1919)). They are still used for schedules in recent research (Tory *et al.* (2013); Jo *et al.* (2014)), and for showing intervals over time in other contexts (Plaisant *et al.* (1996); Andrienko *et al.* (2008)). The Gantt chart that we propose, unlike previous work on movement data, maps *two* variables to the vertical axis (location and person), and we discuss the trade-offs of nesting location inside person, or vice versa, during this mapping.

Crnovrsanin *et al.* (2009) propose an approach somewhat related to the Gantt chart, that maps a continuous *distance* to the vertical axis, which varies with time shown on the horizontal axis. The distance shown is computed with respect to some fixed (or moving) point of reference. This way, objects that move toward or away from each other geographically also usually move toward or away from each other in the distance $\times$ time chart. This allows more details of trajectories to be seen. In our work, we choose a Gantt chart so as to emphasize the locations where people are stationary, and to avoid the problems with widely-different scales of motion

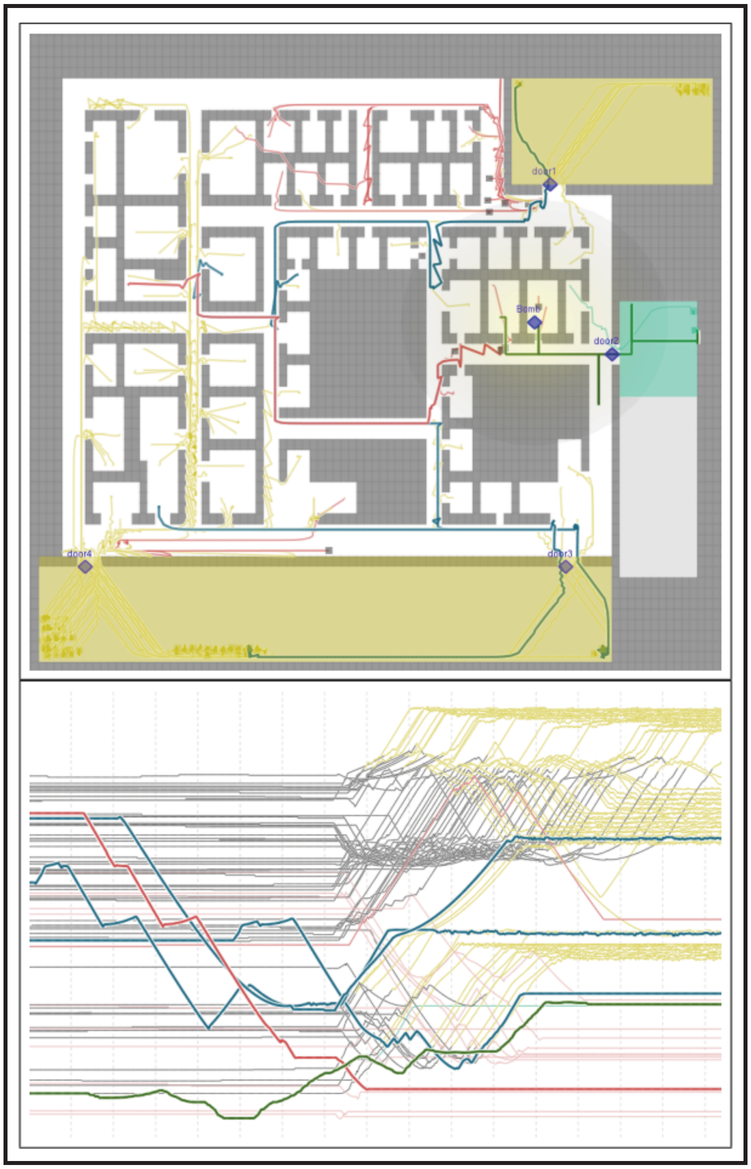

Figure 1.8 A Gantt chart like visualization with distance to a point plotted along y-axis and time plotted along x-axis.
Taken from Crnovrsanin *et al.* (2009).

mentioned in the introduction. We also use multiple coordinated view, which includes a map view, that enable us to add more geographical context to the Gantt chart.

In the next chapter, we describe our research objectives and explain the methodology used to validate our study. We then provide a detailed discussion of our various design decisions while designing the Gantt chart.

CHAPTER 2

OBJECTIVES, METHODOLOGY AND DESIGN DECISIONS

This chapter includes a description of specific design objectives and methods for validating our visualization design. Further, we justify our choice of Gantt charts and various intra-design possibilities while adapting Gantt charts for visualization of movement data.

2.1 Objectives and methods

The main objectives of this thesis are to facilitate exploratory analysis of a movement dataset, to find patterns in the motion of the various objects/persons and to allow for detection of interactions between them. The focus is specially on movement of a small number of people/objects and their interactions.

The objectives of this thesis are :

- Define a set of tasks for movement dataset. We concentrate on a fixed set of tasks for a small number of people (Section 1.2.1).
- Design an appropriate visualization that help facilitate exploratory analysis enabling users to find answers to the tasks defined.
- Design a visualization that helps resolve or mitigate problems related to movement dataset visualization (Section 1.3). Design subgoals :
  - Should enable displaying meetings between objects.
  - Should facilitate the discovery of movement patterns of an individual specifically support the task discussed in Section 1.2.1.
  - Should scale well allowing to plot data collected over a large period of time.
  - Should prevent overplotting and crossover.

- Evaluate the designed visualization.

Munzner (2009) defines a nested model for visualization design and validation. She divides the nested model into four categories, namely task characterisation and problem definitions, data abstraction design, encoding visualization design, and algorithm design. Each category is interconnected such that the output from one becomes the input to the other. Task characterisation consists of understanding the problem domain and obtaining relevant tasks that are needed to be supported based on the data at hand. Such characterisation is usually carried out by interviewing users and understanding the problems they are trying to solve. A valid method to check if correct tasks regarding the data are gathered is usually seen by the rate of adoption. Once the task is defined the next step is processing the raw data into data types and operations that can be mapped using a visual encoding scheme, i.e., data abstraction design. She further discusses that the best way to validate proper mapping of data and operations can be achieved by performing tests on target users. Encoding the data abstraction into a new visual encoding leads to a useful visualization tool. She regards case studies and informal user studies as effective way to analyze that the visual encoding is effective at communicating the proper data abstraction to the user. Finally, once the encoding and interaction scheme is designed lies the algorithm to implement it. A valid way to test if the algorithm is working fast enough to allow for live interaction is by analyzing computational complexity and measure the system time and memory usage.

Munzner states that a visualization paper may contribute to either one category or to more than one category. She provides different evaluations techniques to allow for validation according to the area the paper contributes too. Our paper, can be best described as a encoding visualization design or design study. A good design study provides users with enough background about the problem area so that they can pass judgement about the visual design. This includes stating design requirements by analyzing task followed by proper reasoning behind each design choices made to achieve those requirements. Validation of the design in a design study is equally important as the final visualization itself, so that the readers have a complete grasp of the limitations and usefulness of the visualization. Thus, we provide comprehensive analysis of our design choices while designing the Gantt chart (Section 2.2) and have evaluated our design by performing three case studies (Section 3.4) using the following datasets:

a. Case study 3.4.1: We collected data of one person over a period of 6 months sample at varied rate. Total number of datapoints 21,530.

b. Case study 3.4.2: We also collected data of 6 people over a period of one month. The data ensure there were at least 3 places where everyone meet. The dataset consists of 135,507 data points collected at a frequency ranging from 15 seconds - 5 minutes per point.

c. Case study 3.4.3: GeoLife dataset, Zheng and Fu (2011), collected by Microsoft Research Asia. The data consists of GPS coordinates of 178 users collected over a total duration of 48,203 hours sampled at every 1-5 seconds or every 5-10 meters per point.

For case study 3.4.1 data was collected in part using Google Latitude and Backitude (1st December, 2012 - 31st May, 2013). The second datasets for case study 3.4.2 were collected starting from 1st May, 2013 - 31st May, 2013. We made sure that there were at least three instances where all of the 6 people involved in the data collection (note: during the period of data collection users were allowed to switch off Backitude at instances for privacy purposes). The data was collected using Backitude with the settings mentioned in Annex I:

2.2 Design: Gantt Charts

2.2.1 Why Gantt Charts?

Gantt charts eliminate the overplotting that would occur on 2D geographic maps, and also solve the problem of widely-varying scales of distance between locations of interest. This eliminates two of the four problems mentioned in the second paragraph the introduction. Gantt charts also make it possible to see three variables at once (time, location, and object id), and can be made to scale better along time via folding (discussed later), which at least mitigates the other problems mentioned in our second paragraph of the introduction.

Gantt charts make several behaviors visible (Figure 2.1), allowing us to see frequently and infrequently visited places, repetitions and outliers, and changes in patterns.

Notice that, in the charts of Figure 2.1, every location has a dedicated row, whereas the identity of people is shown by color. Use of color may not scale well when there are 10 or more people. Also, where there are many people and/or many locations, a single person may "jump" vertically by arbitrary distances. This could make it difficult to answer person-centric questions. The next section shows that there are multiple ways to design Gantt charts, some of which address these shortcomings.

2.2.2 Design Choices

As mentioned in the introduction, we can model movement data with 4 variables: *latitude* and *longitude*, that are functions of *time* and *object id*. For convenience in our discussion, we will use *person* as a synonym of *object id*.

Because we are interested primarily in the discrete locations that a person visits (such as home, work, store, café, ...), we can replace *latitude* and *longitude* with a single *location* variable. We can then map *time* to a horizontal axis, and use dimensional stacking (LeBlanc *et al.* (1990); Mihalisin *et al.* (1991)) to map both *location* and *person* to the vertical axis. Depending on the order of nesting of variables, we end up with a chart resembling Figure 2.2 A or B.

Figures 2.2 A and C are person-centric, whereas B, D, and E are location-centric. Each of these is more suited to either the person-centric or location-centric questions presented in Section 1.2.1. For example, to answer the location-centric question "Who most often visits the store?", in a location-centric chart, the user simply needs to look across the row for the store,

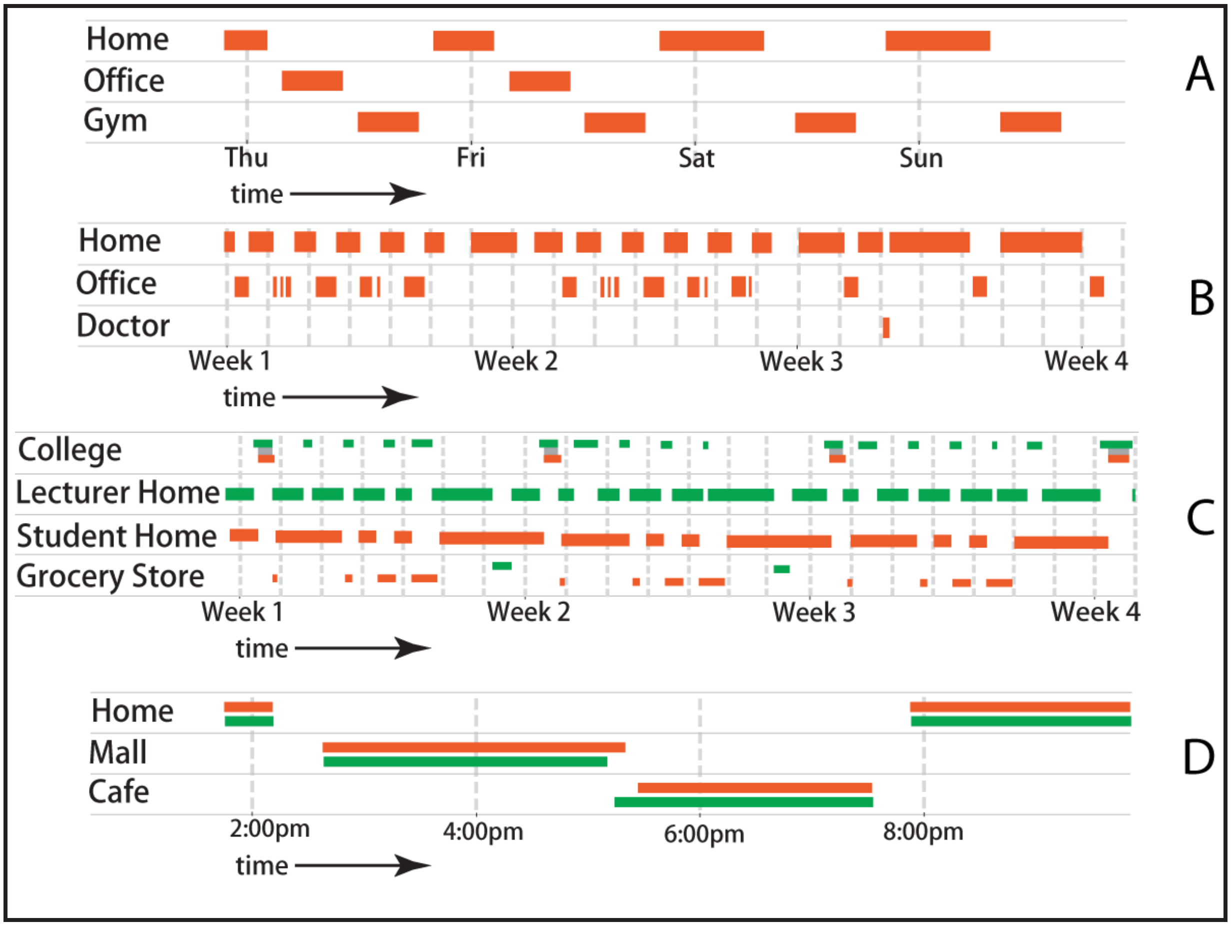

Figure 2.1 Patterns, outliers, meetings, and parallel movements. *A* shows someone going to work and the gym each weekday, and then only to the gym on the weekend. *B* shows a change in pattern: after going to work for 2 weeks, this person visits a doctor, then stays home several days (presumably due to sickness). *C* illustrates possible meetings: a student and lecturer sometime coincide on the campus of a college, while also visiting a grocery store at different times. *D* shows movement together: two people leave home to visit the mall. Around 5:30pm, one goes to a café before being joined by the other a bit later, then both return home.

and examine the contents of that row. Answering the same question in a person-centric chart would require looking over many or all rows.

Thus, there is good reason to support both person-centric and location-centric visualizations, to facilitate both kinds of questions.

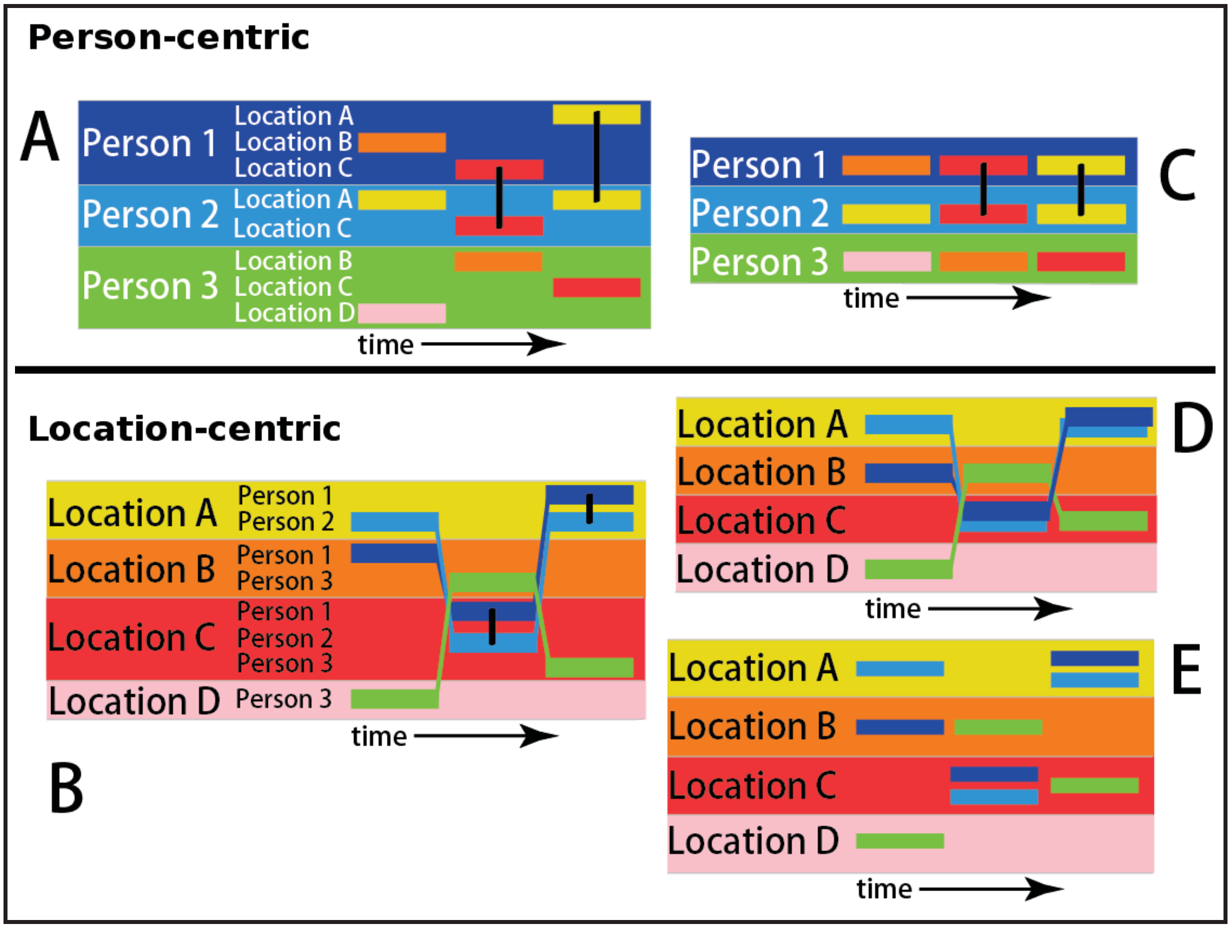

Figure 2.2 All variants of these Gantt charts show the same fictitious data, and all use the same warm colors for locations and cold colors for persons. *A* is person-centric, and *B* is location-centric, each of them nesting one variable within another along the vertical axis. C, D, and E show ways of making rows more narrow. C does this by exploiting the fact that a person can only be in one place at a time. D and E either overlap or stack people, respectively, when multiple people are in the same location. In all charts, the background colors of rows are superfluous and only for illustration. Furthermore, the foreground colors used in A and B are not strictly necessary, though they may help with visual search tasks. Vertical black line segments are used to highlight meetings between people.

Of the two person-centric charts sketched in Figure 2.2 (A and C), C has the advantage of saving space vertically, but requires that locations be color-coded. The datasets we have worked with contain many more locations than people, and therefore color coding of locations would not scale well. Figure 2.2 A, however, does not *require* any color coding. Therefore, **we choose A** as one of the charts we implemented.

Of the three location-centric charts (B, D, E), **we choose E** because it saves space vertically over B, and does not depend on overplotting as in D. E does require that people be color-coded, but so long as we are only visualizing groups of less than 20 people, this does not seem a major problem.

2.2.2.1 Depiction of Meetings

Meetings are clearly illustrated in location-centric views. We suspect that location-centric views will be better for visualizing meetings of people, to show the activity of the whole group within one localized region of the chart.

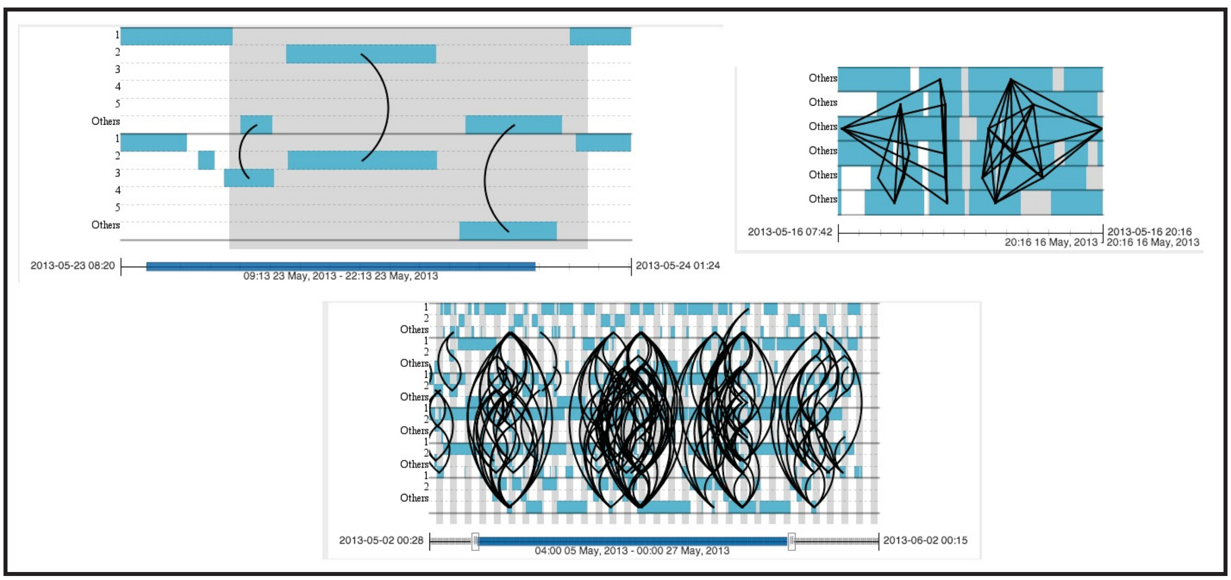

Figure 2.3 Early, unsatisfactory attempts to indicate meetings in person-centric views, drawing either curves or straight line segments between the centers of intervals of all participating people.

In a person-centric view, it was not initially obvious how to best highlight meetings. We first tried drawing connections between the centers of all pairs of relevant intervals, and tried both curved and straight connections (Figure 2.3). This yielded excessively complex results when a meeting involved several people. An alternative approach is to draw a single vertical line segment for each meeting, connecting all relevant intervals, however this may create ambiguities if the line segment happens to pass over intervals of people *not* participating in the meeting. Our final solution was to add a round dot to indicate which people were truly participating in the meeting. For example, Figure 3.7 shows two meetings at overlapping times, in different locations, involving people whose rows are interlaced, but avoids any ambiguity due to the use of dots.

2.2.2.2 World Lines in Location-Centric Views

In location-centric views, another design issue is whether to show "world lines", i.e., polygonal line segments connecting each person across the locations they visit. This is shown in Figure 2.2 B and D, but not in E. We recommend making these lines optional. Displaying them can make each person's path more salient, but can also create visual noise if the user is zoomed out in time and there are many nearly vertical line segments connecting the locations of a person. As shown in the later figures, we sometimes have these turned on (Figure 3.10 and Figure 3.5) in our prototype, and sometimes not (Figure 3.3, 3.7 and 3.11).

CHAPTER 3

MOVEMENTSLICER

The purpose of this chapter is to present our prototype and its features. We provide details about the algorithms used for discretization of movement data into locations, followed by presenting the various features of our prototype.

3.1 Clustering

As mentioned earlier, we are interested only in the movements of people/objects between discrete locations. Zheng *et al.* (2009) suggest a framework that allows for extraction of prominent locations from raw GPS data. Their technique uses a two level clustering approach (Figure 3.1 and Figure 3.2).

- First level: *staypoints* are clustered based on distance and time (Figure 3.1). If two GPS points are in the vicinity of each other i.e. $\Delta D < D_{threshold}$ ($D_{threshold} = 200$ meters in our case) and the time elapsed between them is less than a threshold i.e. $\Delta T < T_{threshold}$ ($T_{threshold} = 20$ minutes in our case), then they are grouped into a *staypoint*. This transforms GPS sequence into *staypoint* sequence.

- Second level: *stayregions* are formed based on a grid-clustering algorithm. For example, figure 3.2 shows *staypoints*, circles, collected over three days. The grid-based clustering algorithm works by dividing the enitre earth into a grid of using the Mercator projection each of a certain size. The *staypoints* are overlayed onto this grid. The cell with the most number of staypoints is selected and then all the *staypoints* inside a $n \times n$ cells (in our case 3×3) are grouped into a *stayregion*. This transforms *staypoints* sequence into larger *stayregion* sequence. Each *stayregion* represents a location.

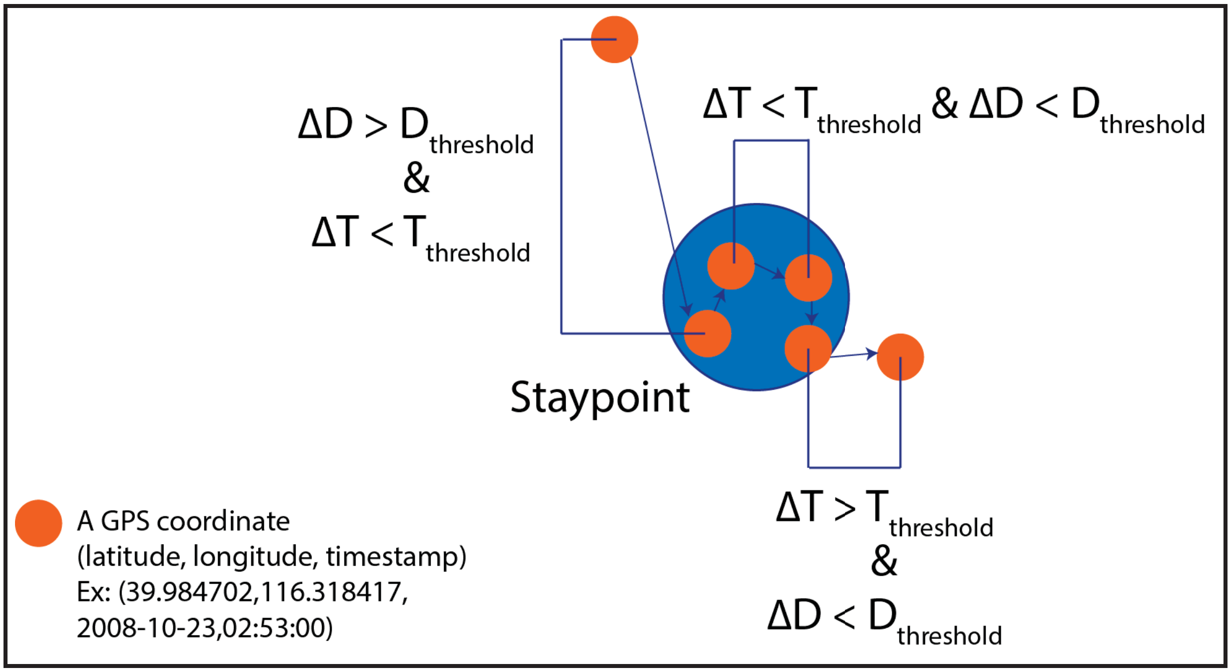

Figure 3.1 Orange circles represent a GPS coordinate. Length of the arrows between two orange circles represents the distance between them. Time increases in the direction of the arrow. If the distance between two points is less than $D_{threshold}$, and the time elapsed between two consecutive coordinates is less than $T_{threshold}$ then those GPS points are grouped into the same *staypoint* shown as the larger blue circle.

This discretizes the raw GPS data into location histories. This history is a summary of all the places a person has visited or an object has been. We visualize these summaries using Gantt charts coordinated with other views.

3.2 Description

Figure 3.3 shows the user interface of our prototype. A session begins with the user choosing movement data files to read in, which are processed using the clustering algorithm to identify locations. Meetings are identified automatically based on a distance threshold. In our case if two *staypoints* are within a distance of less than 50 meters we consider that a meeting has occurred between the people/objects involved with those *staypoints*.

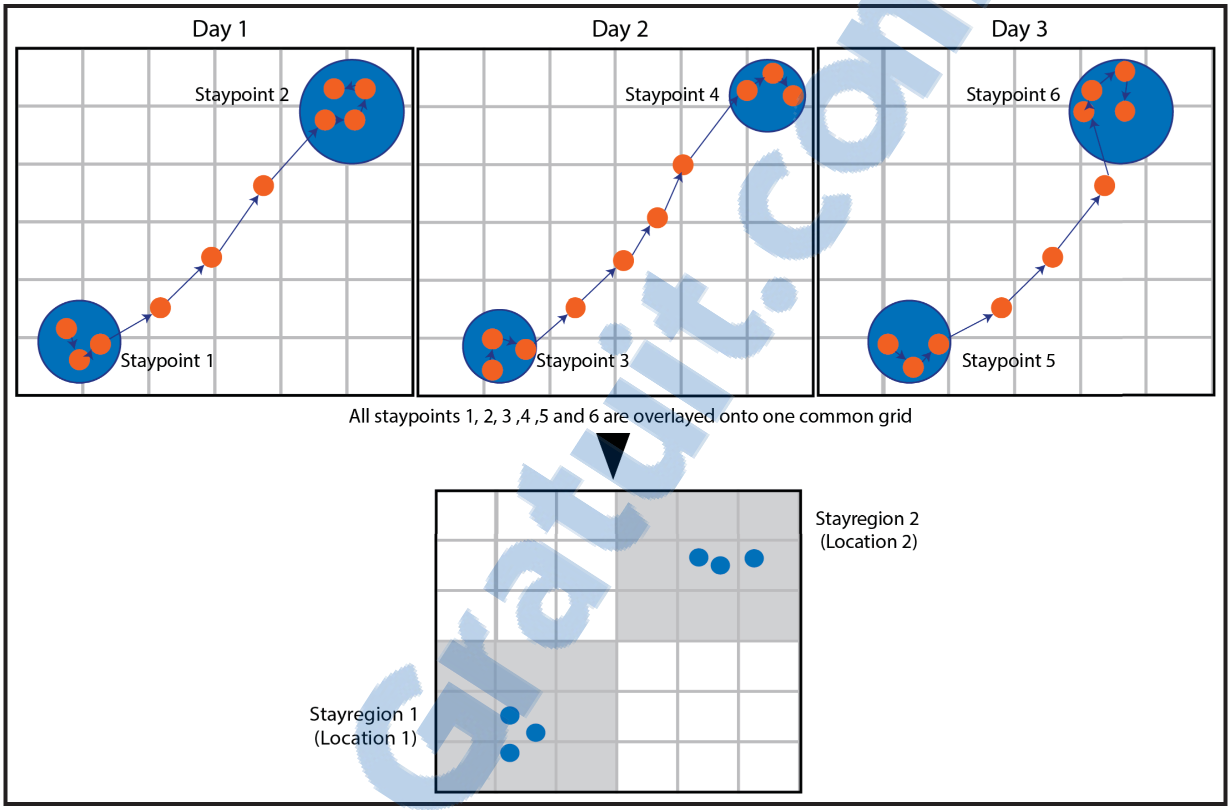

Figure 3.2 Orange circles represent a GPS coordinate. Blue circles represent a *staypoint*. Figure shows a person travelling from home to work for three days. *Staypoints* for each day are overlayed onto one common grid and all the *staypoints* lying within a 3×3 cells are grouped into a single *stayregion* (shaded in grey).

3.2.1 Matrix view

An adjacency matrix, with each person assigned to a row and column, shows the number of meetings between each pair of people. The matrix can be ordered with or without using the barycenter heuristic (Sugiyama *et al.* (1981); Mäkinen and Siirtola (2005)). Figure 3.4 shows that the barycenter heuristic can cluster subsets of people who meet with each other close together.

The matrix also serves as a filtering mechanism. The user may select one or more people in the matrix, causing only data for those people to appear in the other views. The user may also select one or more cells in the matrix, causing only data related to *meetings* of people to appear

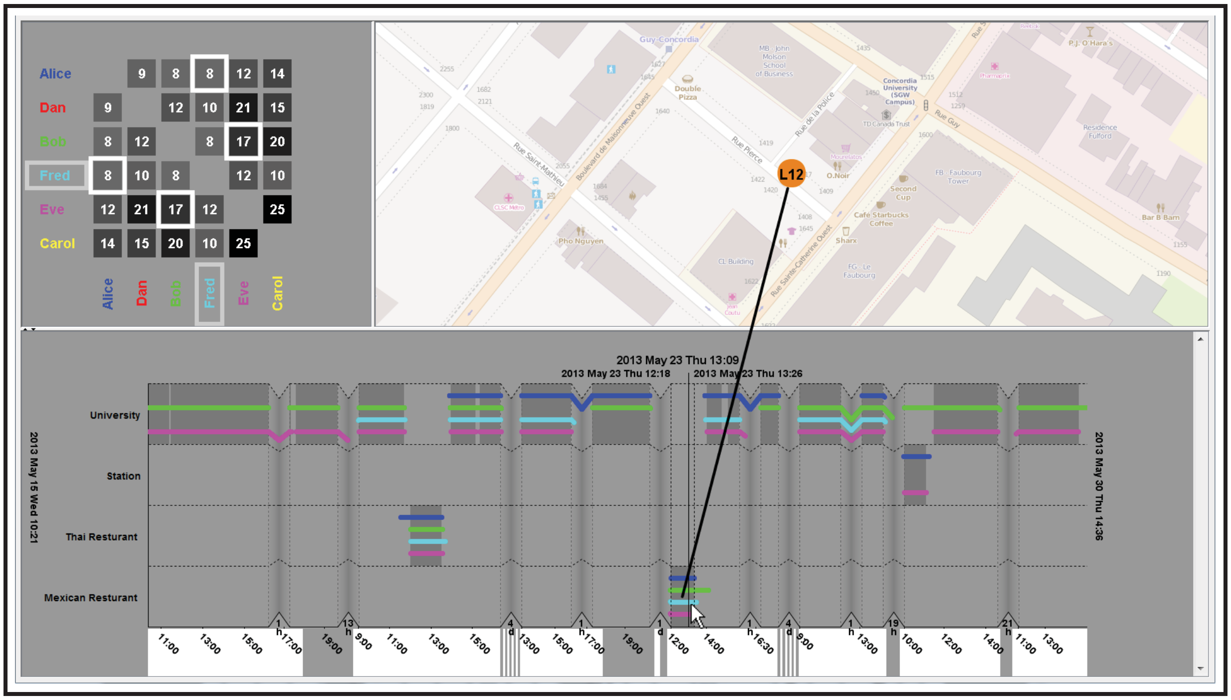

Figure 3.3 The main window. Upper right: 2D geographic map. Upper left: adjacency matrix showing meetings between people. Bottom: Gantt chart, in location-centric mode, with time folding activated to hide empty regions of time. Note the small labels below each fold indicates how much time has been compressed ("1 h" for 1 hour, "4 d" for 4 days, etc.).

in the other views. Thus, although 10 initial people might result in a large number of rows in the Gantt chart, selecting only the cells (meetings) of these people in the matrix typically greatly reduces the number of rows in the Gantt chart.

3.2.2 Map view

The map view is used to display the geographical aspect of movement data. Whenever a subset of meetings between the people or a subset of people is selected in the matrix view, the map view updates to show the regions where the meetings took place. Locations are shown using orange circles with the location label at the center (see Figure 3.3). The map can be scrolled or zoomed using the mouse to allow for more detailed view of regions.

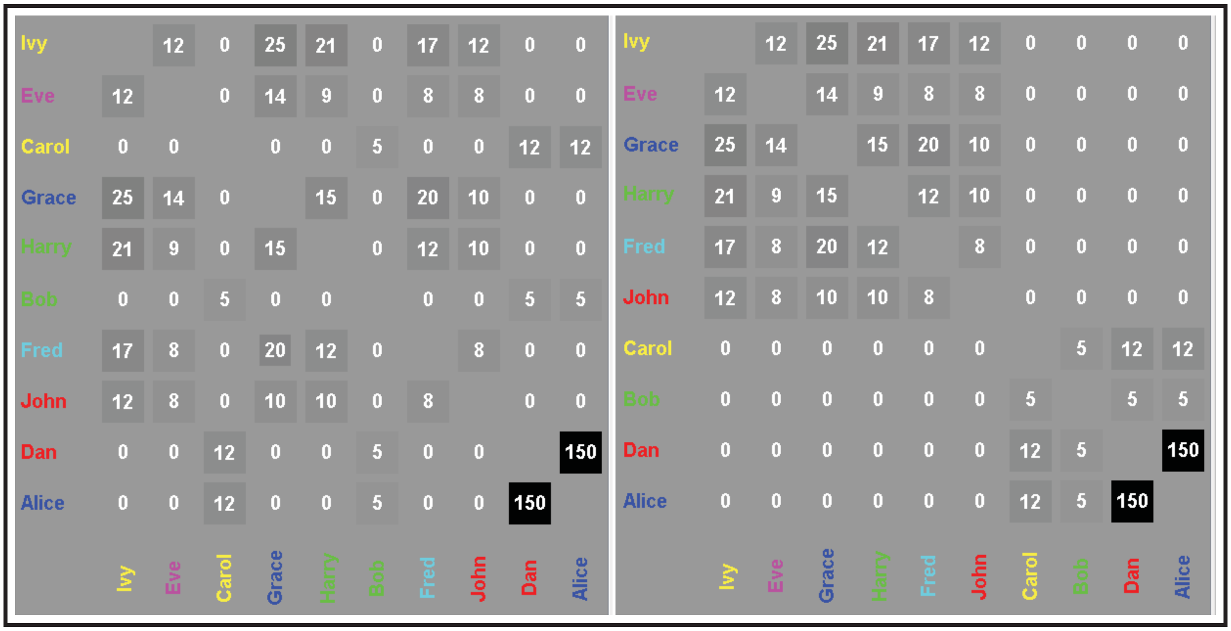

Figure 3.4 The matrix shows meetings between 10 people. *Left:* matrix view ordered without randomly. It's harder to read the matrix as the data is spread all over the matrix. *Right:* the same matrix view reordered using barycentric heuristics which clusters data close together. This increases the readability of the matrix.

3.2.3 Gantt chart

The Gantt chart in Figure 3.3 is location-centric. The user has selected cells in the matrix to view meetings between Fred, Alice and Eve represented with cyan, dark blue and pink color respectively. Meetings are shown in the Gantt chart as darker grey bands, and the horizontal colored strokes allow the user to see when each person arrival, departure and duration of stay. It is straightforward to see who arrived first or last at a meeting, or who left first or last, because the temporal folds increase the horizontal resolution available for each meeting, and the people in the same meeting appear vertically very close.

The temporal folding supported by the Gantt chart is a focus-in-context technique Cockburn *et al.* (2008), comparable to other work where folds are performed along one dimension Mackinlay *et al.* (1991); Elmqvist *et al.* (2008); Trümper *et al.* (2010). We found the darker shading and converging lines in our folds were important to convey a metaphor of receding planar surfaces.

Figure 3.5 shows the same data, over the same time span, but now without folds. Notice that the empty space between meetings has greatly reduced the horizontal resolution available for meetings, making it difficult to see details such as who arrived first at a given meeting.

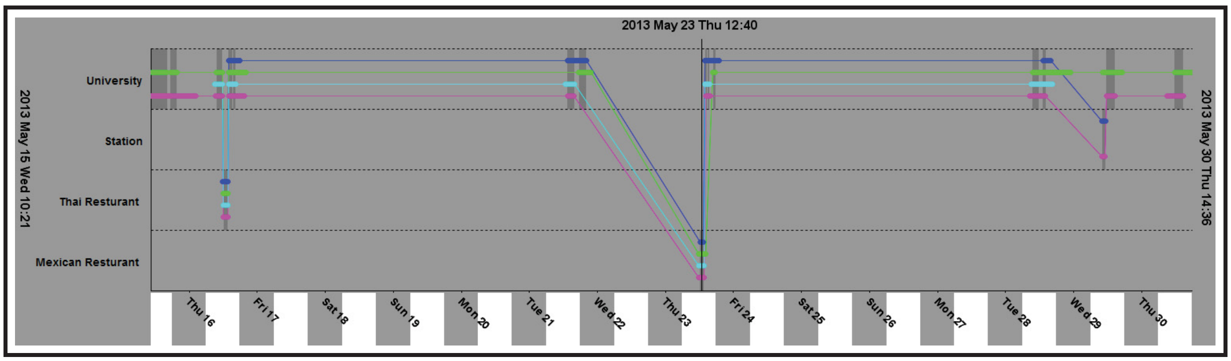

Figure 3.5 The same time span as in Figure 3.3, but now with folds removed, to show how much empty space there is between meetings. World lines are also displayed.

In both Figures 3.3 and 3.5, a black-and-grey strip pattern below the Gantt chart shows day and night (white for day, grey for night). Also, numbers below folds in Figure 3.3 indicate the time hidden by each fold: “1 h” for 1 hour, or “4 d” for 4 days.

The Gantt chart can be panned and zoomed by the user. Optionally a tooltip or excentric labels can be enabled shown in Figure 3.6. The tooltip makes it easy to see which person in involved in the meeting. When the mouse rolls over a meeting, i.e. the darker grey bars, the tooltip displays the person represented by horizontal lines nearest to the mouse. Similarly, excentric labels (Fekete and Plaisant (1999)) can be used. They provided an added advantage of displaying all attendees of the meeting while at the same time showing the person under the mouse by using highlighting. However, excentric labeling is more occluding and blocks other meetings besides when displayed. Thus our prototype provide the options to toggle between three modes: (1) using no tooltip and no excentric labeling, no occlusion, in this mode

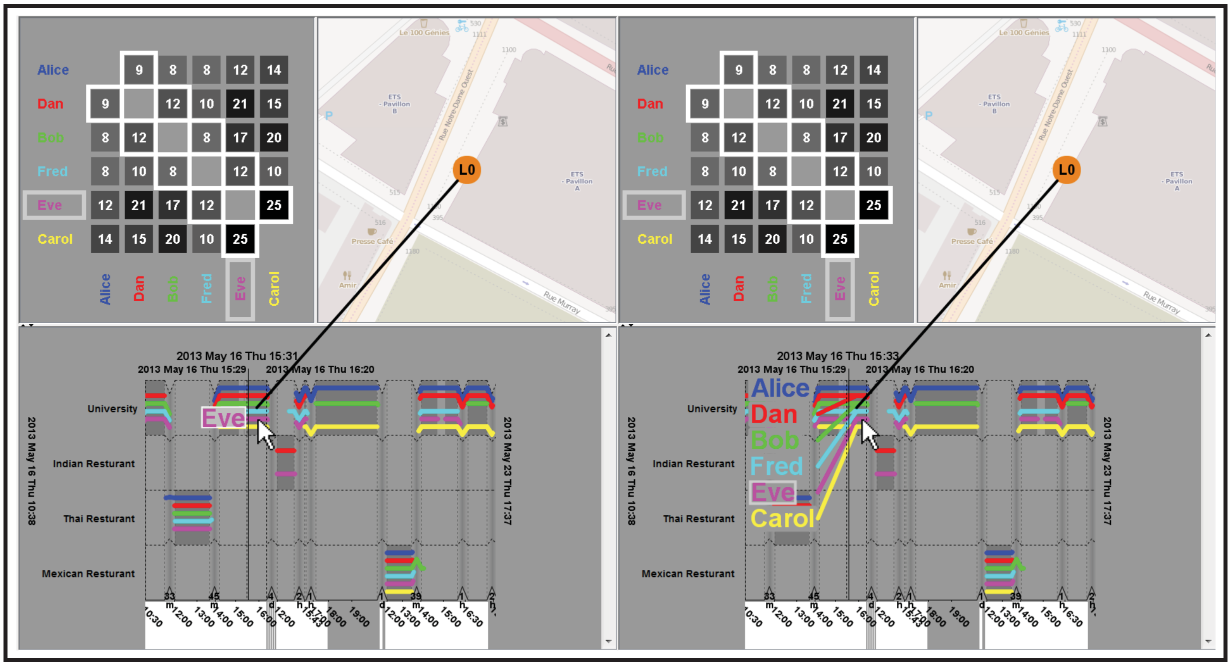

Figure 3.6 Left: Tooltip displays only the person, Eve, under the cursor but is less occluding. Right: Excentric labels display all the people present for the meeting, the highlighting shows the person under the cursor, i.e. Eve. However, excentric labels are more occluding since they take up more space. Both views are coordinated to the matrix view using coordinated highlighting, Eve is highlighted in the matrix view.

coordination with the matrix view is required to see the person under the cursor ; (2) tooltip, some occlusion, using a tooltip in which the person under the cursor is highlighted but all the persons/objects involved in the meeting are not displayed; (3) excentric mode, occlusion, but all the people/objects involved in the meeting are displayed as well as the person under cursor using highlighting.

Hotkeys are used to toggle folding in the Gantt chart, as well as to toggle the drawing of world lines. During the toggling of folding, a smoothly animated transition shows the empty regions folding away, while the unfolded regions grow bigger, all the while maintaining the same total duration within the Gantt chart. Perspective drawing and color shading are used for drawing the folds to enhance the metaphor that “time is being shrunk” in the space available, i.e., per pixel the amount of time passed is greater in the folded regions than the unfolded regions.

Names of locations in the left margin of the Gantt chart can be clicked and edited by the user, to annotate rows with meaningful location names. Thus, during an analytical process, as the user identifies locations on a map, they can record a meaningful name for each location, such as “Building 1”, “University”, “Bob’s Home”, etc.

The Gantt chart supports both the location-centric view shown in Figures 3.3, 3.5, and the person-centric view shown in Figure 3.7. The user may freely switch between these views according to the kinds of tasks or questions they have.

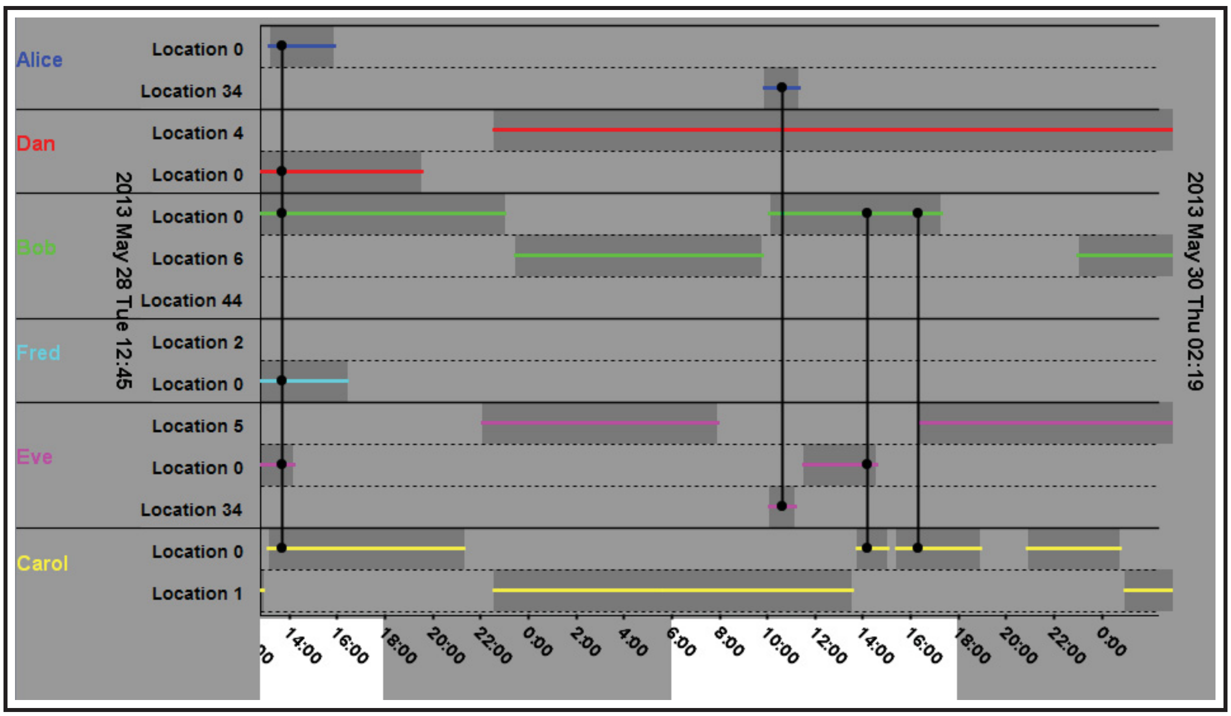

Figure 3.7 Person-centric mode, showing three meetings, two of which overlap in time. Colors identify people, and black vertical line segments show meetings. Black dots disambiguate the participants in the meetings: “blue Alice”, “red Dan”, “green Bob”, “cyan Fred ” , “pink Eve” and “yellow Carol” meet at Location 0. While “blue Alice” and “pink Eve” meet later again at Location 34.

When using the location-centric view, rows within the Gantt chart are sorted by default in descending order of activity with the most visited locations at the top. The user may also reverse this sorting order. This makes it easy for the user to identify the most-often visited locations, or the most rarely visited locations, for a given set of people selected in the adjacency matrix. The user may also interactively adjust the height of rows, and scroll vertically through the Gantt chart if the number of rows exceeds the available space.

To summarize, although the number of rows that are visible in the Gantt chart is limited by screen dimensions, we increase the scalability of our system in multiple ways:

- The adjacency matrix allows the user to identify interesting meetings and filter on people.
- Sorting of rows makes the most interesting (frequent or rare) locations appear at the top of the Gantt chart.
- Folding increases scalability along time and reduces the need to pan and zoom.
- The user may also remove individual locations by right-clicking on rows to concentrate on the important rows and removing irrelevant ones.

3.2.4 Coordinating views

The views in the main window are coordinated North and Shneiderman (2000); Wang Baldonado *et al.* (2000); Roberts (2007) in various ways. First, hovering over a location in the Gantt chart causes the geographic map to animate to that location (coordination through animation), and the relationship is further emphasized through an explicitly drawn black line segment drawn from the cursor in the Gantt chart to the location in the map (Figure 3.3, such explicitly drawn links have also been discussed Buja *et al.* (1991) and used Collins and Carpendale (2007); Steinberger *et al.* (2011); Thudt *et al.* (2013) in previous work). Second, the Gantt chart is coordinated to the matrix view using highlighting. When you roll over the mouse over a meeting in the Gantt chart the corresponding person in the meeting closest to the mouse is highlighted in the matrix view using a light grey color as shown in Figure 3.5 and Figure 3.6 Lastly, the colors allotted to the people and their relative position the same in the matrix view and the Gantt chart. This helps improve coordination further. Coloring is optional and could be toggled on or off.

3.3 Implementation

MovementSlicer follows the model-view-controller architecture. Java and the Java Swing API were used for implementing the front and the back-end. Jxmapviewer (Marinacci (2007)) was used to draw the map view. Our implementation, running on an Intel core i7 processor with 8GB of RAM at 1633MHz, was able to read in data containing 135,507 raw points, containing gaps and noise, and process it to find visited locations and meetings in under 5 seconds.

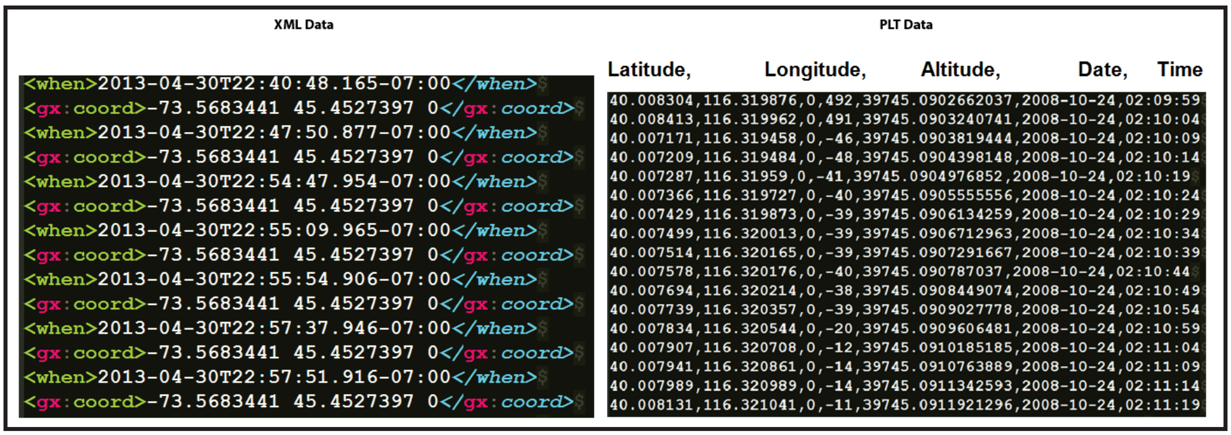

Figure 3.8 Left: XML format of data, Right: PLT format for data.

The data was stored in XML or PLT files (figure 3.8) and loaded upon executing the application. The latitude, longitude and timestamps for each person are loaded into the person class. This data is then processed using the clustering algorithm (section 3.1) to derive the staypoints for each person. The staypoints from all the persons are then used together for clustering to obtain stayregions which are used as locations in the Gantt chart. The clustering algorithm for the staypoints has a complexity of $\mathcal{O}(n^2)$ for n raw data points while the algorithm for calculating the stayregions has a complexity of $\mathcal{O}(m^3)$ for m staypoints. Note that m is typically much smaller than n (in our case from 135,507 GPS points reduces to 433 staypoints). Thus, there are many fewer staypoints than raw GPS points. So even though the complexity of calculating stayregions is cubic, the time taken to calculate the stayregions (locations) is much less (84 stayregions from 433 staypoints) than the time to calculate staypoints.

Interval trees are used for calculate meetings between people. Interval trees are data structures used to store and retrieve time intervals. They are specifically suited to find overlaps between time intervals. We construct an interval tree for all the stayregions once. Meetings can then be displayed by traversing the interval tree for the selected people. The naive algorithm for finding meetings among the people has a complexity of $\mathscr{O}(m^4)$ where m is the number of staypoints. However, using interval trees greatly speeds up the process as it reduces query time. Constructing an interval tree has complexity of $\mathscr{O}(m \log m)$ which is a one-time operation. Once constructed, querying the tree for meetings is a $\mathscr{O}(\log m)$ operation. Construction of interval trees is progressive: new staypoints and stayregions are added on the go without the need to rebuild the entire tree making, it useful for realtime systems. Also, using an ArcGIS database would help improve the performance of the application (Note: for our prototype we are not using an ArcGIS database). ArcGIS databases allow for faster loads as they are optimized for movement databases. They have features which allow for filtering of the dataset based on area and time which could further improve speed when calculating for a specified time interval or region.

Thus, our prototype is scalable for datasets spanning large time spans. However, visually it is scalable up to 20 people when used in location-centric mode, as the location-centric mode uses color to identify people. Whereas for person-centric mode, since color is not important and only used for visual searches, the prototype scales well for more people.

3.4 Case Studies

Munzner (2009) states that a good visual design paper not only mention reasons of choosing a visual scheme but also use case studies to validate and display the usefulness of design. This allows the user to have a better understanding of the capabilities and limitations of the system. We now illustrate the uses of our prototype with real-world datasets.

3.4.1 Case Study 1: One individual over 6 months

This data set was produced by tracking one person's smartphone for 6 months, yielding a total of 21,030 data points. In Figure 3.9, we see 3 months of data, all within the same city. Although there are initially 41 locations identified in the data, and 41 rows displayed, the rows are automatically ordered in descending order of activity, allowing the user to easily see the most frequently visited locations without scrolling downward. The user manually inspected each row to see the corresponding location on the map, and annotated the rows in the Gantt chart to make the visualization more suitable for interpretation and presentations to others.

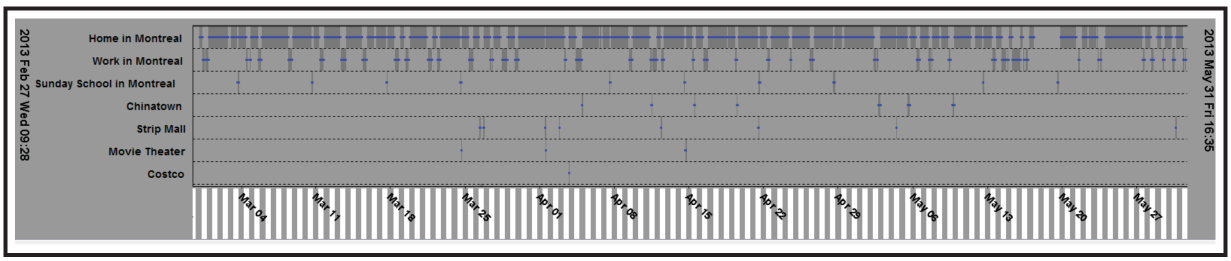

Figure 3.9 Case study 1: three months of activity within Montreal.

In Figure 3.10, we see a trip from Montreal, Canada to Durham, NC, within the same dataset. The data covers a large range of geographic distances: flight distance from Montreal to Durham covers over 1000 km, but travelling between “Hotel in Durham” and “Meeting in Durham” covers less than 3 kilometers. Zooming out on a geographic map to see all places visited, we would only see the 3 cities covered: Montreal, Durham, and New York City. The details of movements within any city are far too small to see in such a geographic overview. In contrast, with the Gantt chart, all locations where the person visited are allocated a row, making the sequencing and timing salient. This demonstrates an important advantage of the Gantt chart over the other approaches in Figure 1.2.

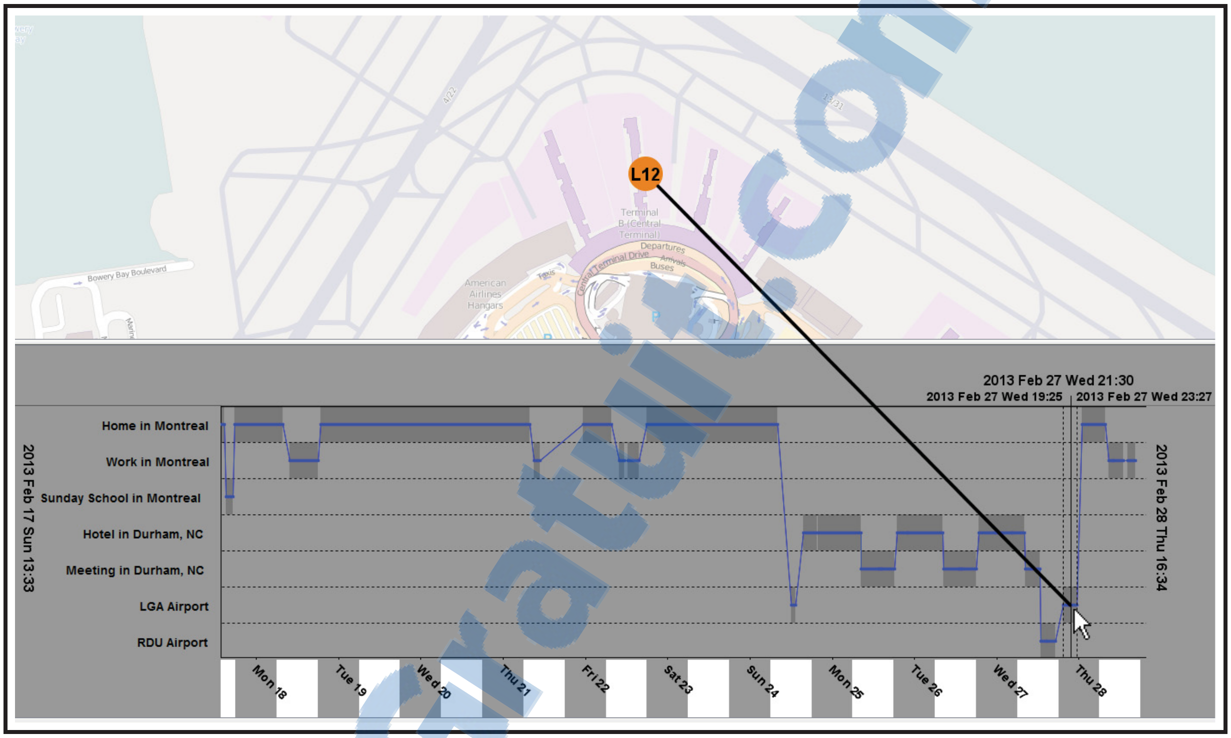

Figure 3.10 Case Study 1: one week of activity in Montreal followed by a round-trip flight to Durham (airport code: RDU) via New York (airport code: LGA). GPS tracking failed during the LGA-RDU leg of the trip, which shows up in the Gantt chart as a sudden jump from LGA to a hotel in Durham.

3.4.2 Case Study 2: Six people over 1 month

Six people in our research team tracked their movements over a 1 month period, while traveling to and from work, and occasionally meeting in places outside work such as restaurants. Our implementation is able to read in the data for all six people (10,000 raw points, or roughly 1 point per person every 2 minutes), containing gaps and noise, and process and cluster it to find visited locations in under 5 seconds on a recent laptop. Our implementation also identifies meetings between multiple people. Visualizing such meetings is important for answering questions such as “Who among these people often meet together?”, “Where do they meet?”, and “Who is early or late for a meeting?”

Initially, 78 different locations are identified in the dataset, and all of these locations can be viewed in the Gantt chart if the user selects all 6 people in the adjacency matrix. However, if the user selects only the cells in the matrix, the filtering results in only 4 locations being shown in the Gantt chart, where meetings of at least two people occurred (Figure 3.11).

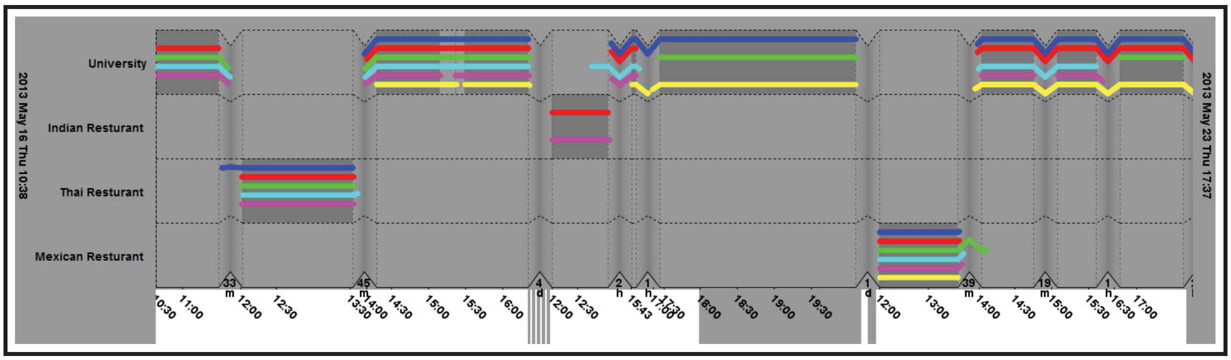

Figure 3.11   Case study 2: meetings between 6 members of our lab.

3.4.3 Case Study 3: GeoLife data

We examined subsets of the GeoLife (Zheng *et al.* (2010b)) dataset, containing 178 individuals tracked over several months. Figure 3.12 shows meetings for a subset of 12 people that were loaded. We immediately notice that 3 of these people have met an unusually high number of times. The matrix allows the user to easily filter down to any pair of these 3 people. Examination of the “red Dan” and “pink Eve” people indicate that they may be close friends, as they are often together (Figure 3.13). (In the GeoLife dataset, these two people are identified as 003 and 004.)

However, examining the activity of “red Dan” and “blue Alice” reveals that they seem to have *identical* activity patterns over large spans of time (Figure 3.14). (In the GeoLife dataset, these two people are identified as 003 and 000.) After discovering this, we checked the raw data, and found that 129 of the original files for these individuals seemed identical. This points to an interesting anomaly worthy of further investigation. Were portions of the data accidentally duplicated?

	Carol	Dan	Katy	Larry	Grace	Harry	Eve	Bob	Alice	Ivy	John	Fred
Carol		0	0	0	0	0	0	0	0	0	0	0
Dan	0		0	0	0	0	416	2	344	0	0	0
Katy	0	0		0	0	0	0	0	0	0	0	0
Larry	0	0	0		0	0	0	0	0	0	0	4
Grace	0	0	0	0		0	0	0	0	0	0	1
Harry	0	0	0	0	0		0	0	0	0	0	0
Eve	0	416	0	0	0	0		0	220	0	0	0
Bob	0	2	0	0	0	0	0		2	0	0	0
Alice	0	344	0	0	0	0	220	2		0	0	1
Ivy	0	0	0	0	0	0	0	0	0		8	1
John	0	0	0	0	0	0	0	0	0	8		1
Fred	0	0	0	4	1	0	0	0	1	1	1	

Figure 3.12 Case study 3: an adjacency matrix for several people from the GeoLife dataset.

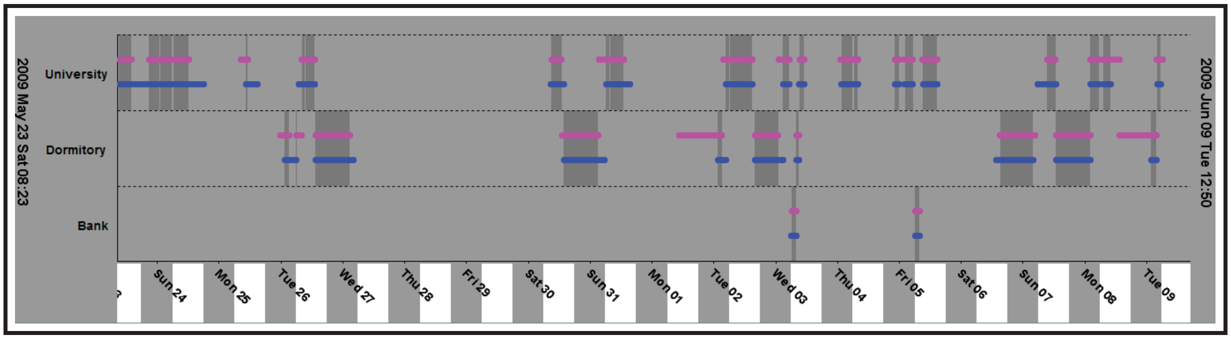

Figure 3.13 Case study 3: two people often together, at a university, at a dormitory, and at a bank.

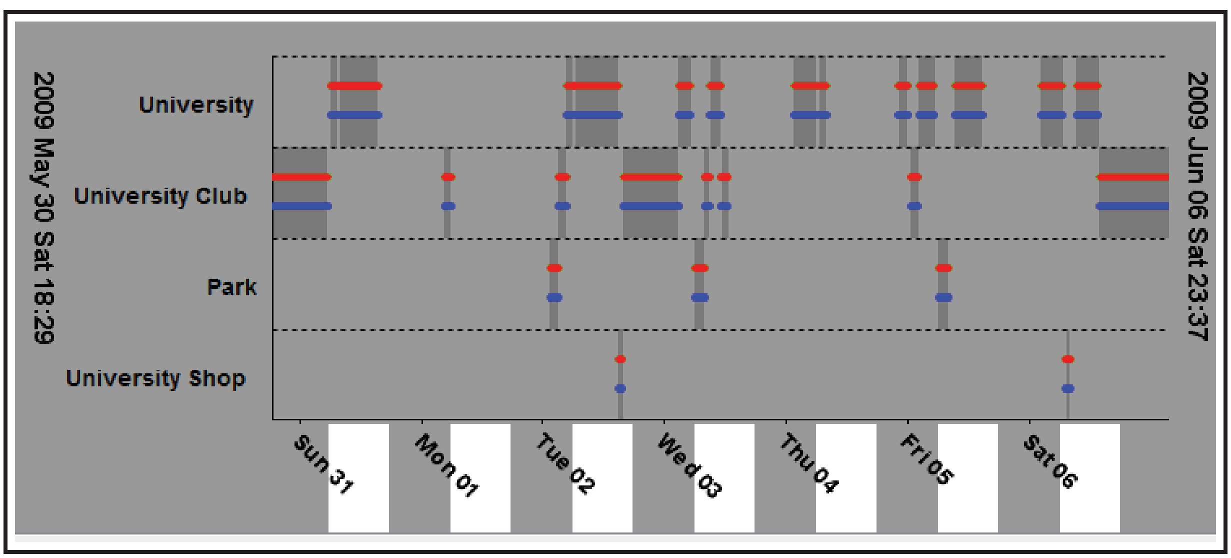

Figure 3.14 Case study 3: two people with identical data.

CONCLUSION AND FUTURE DIRECTIONS

We have presented MovementSlicer, a tool designed for understanding individual behaviors and meetings between small groups of people. Our taxonomy of visualizations (Figure 1.2, proposed categories of tasks (Section 1.2.1), and analysis of design options suggest that two kinds of Gantt charts are particularly suited for such a tool, person-centric and location-centric, both of which show two categorical variables with respect to time. Our prototype supports filtering via an adjacency matrix of meetings, sorting of rows by activity level, and multi-focal temporal folding, all features that help handle more people, locations, and longer time spans, respectively. Our tool's use of coordination through links, smoothly animated transitions, and user-annotation of rows are all useful features to consider implementing in similar systems.

Several enhancements to our prototype would be possible. First, we could implement the display of trajectories on the geographic map, possibly clustering them and displaying median trajectories for each cluster (Buchin *et al.* (2013)). Second, we could implement the Gantt chart variant shown in Figure 2.2, C, as a means of saving space vertically. Third, to enhance coordination between views, we have started experimenting with displaying thumbnails of geographic map regions *within* the Gantt chart, for example, when the mouse cursor rolls over an event. (This is comparable to the use of embedded maps in Thudt *et al.* (2013).) Fourth, to further enrich coordination, we have begun to design a kind of lens that could be passed over the geographic map, that would display the activity under the lens as a distance $\times$ time graph, i.e., like a small version of Crnovrsanin *et al.* (2009)'s graph, where distance could be measured with respect to the center of the map region under the lens. Such a lens could be useful for seeing when and how many times different people visit a given region of the map.

Finally, our prototype could be made more flexible by allowing the user to map the horizontal and vertical axes to different quantities. We have already demonstrated the ability to nest person within location or location within person along the vertical axis. However, other systems (Hurter *et al.* (2009); Mackinlay *et al.* (2007)) allow arbitrary associations between data and axes. In the context of movement data, one useful mapping could be time along the horizontal, and *duration* of events along the vertical (Muelder *et al.* (2009); Qiang *et al.* (2012)), to make longer meetings or events stand out from others. Another would be the ability to map time-of-day to one axis, and date to another axis (as done with the "pixel based road speed views" in Wang *et al.* (2013)), or more generally.

APPENDIX I

SETTINGS OF THE APPLICATION USED FOR DATA COLLECTION

The data was collected using Backitude with the following settings: (Note: During the period of data collection users were allowed to switch off Backitude at instances for privacy purposes.)

Enable Backitude: Checked

Settings > Standard mode settings

- Time Interval: 3 minutes
- Location Polling timeout: 30 seconds

Settings > Update settings

- Time Interval: 3 minutes
- Location Steals : Checked
- Maximum Steals Rate: 15 seconds - 5 minutes
- Minimum Change in Distance: None

Settings > Accuracy Settings

- Minimum GPS Accuracy: 6 metres
- Minimum WiFi Accuracy: 100 metres
- Timeout Fallback Options: Most accurate found, or do not update at all
- Previous Accuracy Fallback: Checked

Settings > Offline Storage Settings

- Offline Storage Enabled: Checked
- Sync Options: WiFi available only

Settings > Status bar icon: Display when application is enabled

BIBLIOGRAPHY

Adrienko, N. and G. Adrienko. 2011. "Spatial Generalization and Aggregation of Massive Movement Data". *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, n° 2, p. 205-219.

Amini, F., S. Rufiange, Z. Hossain, Q. Ventura, P. Irani, and M. J. McGuffin. 2015. "The Impact of Interactivity on Comprehending 2D and 3D Visualizations of Movement Data". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 21, n° 1, p. 122–135.

Andrienko, G., N. Andrienko, P. Bak, D. Keim, S. Kisilevich, and S. Wrobel. 2011. "A conceptual framework and taxonomy of techniques for analyzing movement". *Journal of Visual Languages & Computing*, vol. 22, n° 3, p. 213–232.

Andrienko, N. and G. Andrienko. 2011. "Spatial generalization and aggregation of massive movement data". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 17, n° 2, p. 205–219.

Andrienko, N. and G. Andrienko. 2013. "Visual analytics of movement: An overview of methods, tools and procedures". *Information Visualization*, vol. 12, n° 1, p. 3–24.

Andrienko, N., G. Andrienko, and P. Gatalsky. 2003. "Exploratory spatio-temporal visualization: an analytical review". *Journal of Visual Languages & Computing*, vol. 14, n° 6, p. 503–541.

Andrienko, N., G. Andrienko, M. Wachowicz, and D. Orellana. 2008. "Uncovering interactions between moving objects". In *Proc. International Conference GIScience*. p. 16–26.

Bach, B., P. Dragicevic, D. Archambault, C. Hurter, and S. Carpendale. 2014. "A Review of Temporal Data Visualizations Based on Space-Time Cube Operations". In *Proc. Eurographics Conference on Visualization (EuroVis)*.

Backitude. "Downwind Development: Backitude". <https://play.google.com/store/apps/details?id=gaugler.backitude&hl=en>.

Baddeley, A. 1992. "Working memory". *Science*, vol. 255, n° 5044, p. 556–559.

Bertin, J., 1967. *Sémiologie graphique: Les diagrammes, Les réseaux, Les cartes*. (2nd edition 1973, English translation 1983), Paris : Éditions Gauthier-Villars.

Blaas, J., C. P. Botha, E. Grundy, M. W. Jones, R. S. Laramee, and F. H. Post. 2009. "Smooth graphs for visual exploration of higher-order state transitions". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 15, n° 6, p. 969–976.

Buchin, K., M. Buchin, M. van Kreveld, M. Löffler, R. I. Silveira, C. Wenk, and L. Wiratma. 2013. "Median Trajectories". *Algorithmica*, vol. 66, n° 3, p. 595–614.

Buja, A., J. A. McDonald, J. Michalak, and W. Stuetzle. 1991. "Interactive Data Visualization using Focusing and Linking". In *Proceedings of IEEE Visualization (VIS)*. p. 156–163, 419.

Cockburn, A., A. Karlson, and B. B. Bederson. 2008. "A review of overview+detail, zooming, and focus+context interfaces". *ACM Computing Surveys (CSUR)*, vol. 41, n° 1.

Collins, C. and S. Carpendale. 2007. "VisLink: Revealing relationships amongst visualizations". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 13, n° 6, p. 1192–1199.

Crnovrsanin, T., C. Muelder, C. Correa, and K.-L. Ma. 2009. "Proximity-based Visualization of Movement Trace Data". In *Proc. IEEE VAST*.

Davies, J. "Spherical Voronoi Diagram". <https://www.jasondavies.com/maps/voronoi/>.

Dinkla, K., M. J. van Kreveld, B. Speckmann, and M. A. Westenberg. 2012. "Kelp Diagrams: Point Set Membership Visualization". *Computer Graphics Forum*, vol. 31, n° 3, p. 875–884.

Dodge, S., R. Weibel, and A.-K. Lautenschütz. 2008. "Towards a taxonomy of movement patterns". *Information visualization*, vol. 7, n° 3-4, p. 240–252.

Dragicevic, P., G. Ramos, J. Bibliowitcz, D. Nowrouzezahrai, R. Balakrishnan, and K. Singh. 2008. "Video browsing by direct manipulation". In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*. p. 237–246.

Elmqvist, N. and J.-D. Fekete. 2010. "Hierarchical aggregation for information visualization: Overview, techniques, and design guidelines". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 16, n° 3, p. 439–454.

Elmqvist, N., N. Henry, Y. Riche, and J.-D. Fekete. 2008. "Mélange: space folding for multi-focus interaction". In *Proc. SIGCHI Conference on Human Factors in Computing Systems*. p. 1333–1342. ACM.

Fekete, J.-D. and C. Plaisant. 1999. "Excentric labeling: dynamic neighborhood labeling for data visualization". In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. p. 512–519. ACM.

Gantt, H. L., 1919. *Organizing for work*. New York : Harcourt, Brace & Rowe.

Gigantt. 2014. "http://www.gigantt.com/". http://www.youtube.com/watch?v=XiD4ga2s13w.

Guo, H., Z. Wang, B. Yu, H. Zhao, and X. Yuan. 2011. "TripVista: Triple Perspective Visual Trajectory Analytics and Its Application on Microscopic Traffic Data at a Road Intersection". In *Proc. IEEE PacificVis*.

Haunert, J.-H. and L. Sering. 2011. "Drawing Road Networks with Focus Regions". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 17, n° 12, p. 2555–2562.

Heer, J. and B. Shneiderman. 2012. "Interactive dynamics for visual analysis". *Queue*, vol. 10, n° 2, p. 30.

Henry, N., J.-D. Fekete, and M. J. McGuffin. 2007. "NodeTrix: A Hybrid Visualization of Social Networks". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 13, n° 6, p. 1302–1309.

Henry Riche, N., T. Dwyer, B. Lee, and S. Carpendale. 2012. "Exploring the Design Space of Interactive Link Curvature in Network Diagrams". In *Proceedings of ACM Advanced Visual Interfaces (AVI)*.

Hlawatsch, M., M. Burch, and D. Weiskopf. 2014. "Visual Adjacency Lists for Dynamic Graphs". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 20, n° 11, p. 1590–1603.

Hollands, J. G., R. Korsnes, J. Llinas, G. Ovsttun, and T. Porathe. 2004. *Using visualization to address human capacity limitations, syndicate 3*. Technical report.

Holten, D. 2006. "Hierarchical Edge Bundles: Visualization of Adjacency Relations in Hierarchical Data". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 12, n° 5, p. 741–748.

Holten, D. and J. J. van Wijk. 2009. "Force-Directed Edge Bundling for Graph Visualization". *Computer Graphics Forum (EuroVis 2009)*, vol. 28, n° 3, p. 983–990.

Hurter, C., B. Tissoires, and S. Conversy. 2009. "FromDaDy: Spreading aircraft trajectories across views to support iterative queries". *TVCG*.

Isaacs, K., P.-T. Bremer, I. Jusufi, T. Gamblin, A. Bhatele, M. Schulz, and B. Hamann. 2014. "Combing the Communication Hairball: Visualizing Large-Scale Parallel Execution Traces using Logical Time". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 20, n° 12.

Javed, W. and N. Elmqvist. 2012. "Exploring the design space of composite visualization". In *IEEE Pacific Visualization Symposium (PacificVis)*. p. 1–8.

Jo, J., J. Huh, J. Park, B. Kim, and J. Seo. 2014. "LiveGantt: Interactively Visualizing a Large Manufacturing Schedule". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 20, n° 12.

Kapler, T. and W. Wright. 2004. "GeoTime Information Visualization". In *Proc. IEEE InfoVis*.

Kraak, M.-J. 2003. "The space-time cube revisited from a geovisualization perspective". In *Proc. 21st International Cartographic Conference*. p. 1988–1996.

Kristensson, P. O., N. Dahlbäck, D. Anundi, M. Björnstad, H. Gillberg, J. Haraldsson, I. Mårtensson, M. Nordvall, and J. Ståhl. 2009. "An Evaluation of Space Time Cube Representation of Spatiotemporal Patterns". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 15, n° 4.

Krüger, R., D. Thom, M. Wörner, H. Bosch, and T. Ertl. 2013. "TrajectoryLenses: A Set-based Filtering and Exploration Technique for Long-term Trajectory Data". *Computer Graphics Forum*, vol. 32, n° 3pt4, p. 451–460.

Krzywinski, M., J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra. 2009. "Circos: an information aesthetic for comparative genomics". *Genome research*, vol. 19, n° 9, p. 1639–1645.

LeBlanc, J., M. O. Ward, and N. Wittels. 1990. "Exploring N-Dimensional Databases". In *Proceedings of IEEE Visualization (VIS)*. p. 230–237.

Lee, B., C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. 2006. "Task Taxonomy for Graph Visualization". In *Proceedings of AVI workshop BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*.

Liu, H., Y. Gao, L. Lu, S. Liu, H. Qu, and L. M. Ni. 2011. "Visual analysis of route diversity". In *Proceedings of IEEE Visual Analytics Science and Technology (VAST)*. p. 171–180.

Mackinlay, J. D., P. Hanrahan, and C. Stolte. 2007. "Show Me: Automatic Presentation for Visual Analysis". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 13, n° 6, p. 1137–1144.

Mackinlay, J. D., G. G. Robertson, and S. K. Card. 1991. "The perspective wall: Detail and context smoothly integrated". In *Proc. ACM Conference on Human Factors in Computing Systems (CHI)*. p. 56–63.

Mäkinen, E. and H. Siirtola. 2005. "The Barycenter Heuristic and the Reorderable Matrix". *Informatica*, vol. 29, n° 3, p. 357–363.

Marinacci, J. 2007. "JXMapViewer". <https://today.java.net/pub/a/today/2007/10/30/building-maps-into-swing-app-with-jxmapviewer.html>.

Mihalisin, T., J. Timlin, and J. Schwegler. 1991. "Visualization and Analysis of Multi-variate Data: A Technique for All Fields". In *Proceedings of IEEE Visualization (VIS)*. p. 171–178.

Montoliu, R. and D. Gatica-Perez. 2010. "Discovering human places of interest from multimodal mobile phone data". In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*. p. 12. ACM.

Muelder, C., F. Gygi, and K.-L. Ma. 2009. "Visual Analysis of Inter-Process Communication for Large-Scale Parallel Computing". *IEEE TVCG*, vol. 15, n° 6, p. 1129–1136.

Munzner, T. 2008. "Process and pitfalls in writing information visualization research papers". Chapter (pp. 134–153) in Andreas Kerren, John T. Stasko, Jean-Daniel Fekete, Chris North, editors, Information Visualization: Human-Centered Issues and Perspectives, Springer LNCS Volume 4950.

Munzner, T. 2009. "A nested model for visualization design and validation". *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, n° 6, p. 921–928.

Nelson, T. H. 1999. "Xanalogical structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use". *ACM Computing Surveys*, vol. 31, n° 4, p. 33.

North, C. and B. Shneiderman. 2000. "Snap-Together Visualization: A User Interface for Coordinating Visualizations via Relational Schemata". In *Proceedings of Advanced Visual Interfaces (AVI)*. p. 128–135.

North, C. L. 2000. "A User Interface for Coordinating Visualizations based on Relational Schemata: Snap-Together Visualization". PhD thesis, Department of Computer Science, Department, University of Maryland, College Park, Maryland.

Peuquet, D. J. 1994. "It's About Time: A Conceptual Framework for the Representation of Temporal Dynamics in Geographic Information Systems". *Annals of the Association of American Geographers*, vol. 84, n° 3, p. 441–461.

Plaisant, C., B. Milash, A. Rose, S. Widoff, and B. Shneiderman. 1996. "LifeLines: visualizing personal histories". In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*.

Pupyrev, S., L. Nachmanson, and M. Kaufmann. 2010. Improving layered graph layouts with edge bundling. Brandes, U. and Sabine Cornelsen, editors, *International Symposium on Graph Drawing (GD)*, volume 6502 of *LNCS*, p. 329–340.

Qiang, Y., M. Delafontaine, M. Versichele, P. De Maeyer, and N. Nico Van de Weghe. 2012. "Interactive analysis of time intervals in a two-dimensional space". *Information Visualization*, vol. 11, n° 4, p. 255–272.

Rinzivillo, S., D. Pedreschi, M. Nanni, F. Giannotti, N. Andrienko, and G. Andrienko. 2008. "Visually driven analysis of movement data by progressive clustering". *Information Visualization*, vol. 7, n° 3-4, p. 225–239.

Roberts, J. C. 2007. "State of the Art: Coordinated & Multiple Views in Exploratory Visualization". In *Coordinated and Multiple Views in Exploratory Visualization (CMV)*. p. 61–71.

Rufiange, S., M. J. McGuffin, and C. P. Fuhrman. 2012. "TreeMatrix: A Hybrid Visualization of Compound Graphs". *Computer Graphics Forum*, vol. 31, n° 1, p. 89–101.

Santosa, S., F. Chevalier, R. Balakrishnan, and K. Singh. 2013. "Direct Space-Time Trajectory Control for Visual Media Editing". In *Proceedings of ACM Conference on Human Factors in Computing Systems (CHI)*. p. 1149–1158.

Shneiderman, B. 1996. "The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations". In *Proceedings of IEEE Symposium on Visual Languages (VL)*. p. 336–343.

Steinberger, M., M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. 2011. "Context-Preserving Visual Links". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 17, n° 12, p. 2249–2258.

Stolte, C., D. Tang, and P. Hanrahan. 2002. "Polaris: A System for Query, Analysis, and Visualization of Multidimensional Relational Databases". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 8, n° 1, p. 52–65.

Sugiyama, K., S. Tagawa, and M. Toda. February 1981. "Methods for Visual Understanding of Hierarchical System Structures". *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-11, n° 2, p. 109–125.

Thudt, A., D. Baur, and S. Carpendale. 2013. "Visits: A Spatiotemporal Visualization of Location Histories". In *Proc. Eurographics Conference on Visualization (EuroVis), Short Papers*. p. 79–83.

Tory, M., S. Staub-French, D. Huang, Y.-L. Chang, C. Swindells, and R. Pottinger. 2013. "Comparative visualization of construction schedules". *Automation in Construction*, vol. 29, p. 68–82.

Trümper, J., J. Bohnet, and J. Döllner. 2010. "Understanding Complex Multithreaded Software Systems by Using Trace Visualization". In *Proc. International Symposium on Software Visualization (SoftVis)*. p. 133–142.

Ventura, Q. and M. J. McGuffin. 2014. "Geo-Topo maps: hybrid visualization of movement data over building floor plans and maps". In *Proc. Graphics Interface (GI)*. p. 159–166.

Wang, Y.-S. and M.-T. Chi. 2011. "Focus+context metro maps". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 17, n° 12, p. 2528–2535.

Wang, Z., M. Lu, X. Yuan, J. Zhang, and H. van de Wetering. 2013. "Visual Traffic Jam Analysis Based on Trajectory Data". *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, vol. 19, n° 12, p. 2159–2168.

Wang Baldonado, M. Q., A. Woodruff, and A. Kuchinsky. 2000. "Guidelines for Using Multiple Views in Information Visualization". In *Proceedings of Advanced Visual Interfaces (AVI)*. p. 110–119.

Wattenberg, M. 2002. "Arc Diagrams: Visualizing Structure in Strings". In *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*. p. 110–116.

Wood, J., J. Dykes, and A. Slingsby. 2010. "Visualisation of origins, destinations and flows with OD maps". *The Cartographic Journal*, vol. 47, n° 2, p. 117–129.

Yan, P. F., R. P. Biuk-Aghai, S. Fong, and Y. W. Si. 2007. "Detection of suspicious patterns in secure physical environments". In *Fourth International Conference on Information Technology and Applications (ICITA 2007), Jan. 2007, v. 1, p. 40-45*.

Zeng, W., C.-W. Fu, S. M. Arisona, and H. Qu. 2013. "Visualizing Interchange Patterns in Massive Movement Data". *Computer Graphics Forum*, vol. 32, p. 271–280.

Zhao, S., M. J. McGuffin, and M. H. Chignell. 2005. "Elastic Hierarchies: Combining Treemaps and Node-Link Diagrams". In *Proceedings of IEEE Symposium on Information Visualization (InfoVis)*. p. 57–64.

Zheng, V. W., Y. Zheng, X. Xie, and Q. Yang. 2010a. "Collaborative Location and Activity Recommendations with GPS History Data". In *Proceedings of the 19th International Conference on World Wide Web*. (New York, NY, USA 2010), p. 1029–1038. ACM.

Zheng, Y. and H. Fu. July 2011. *Geolife GPS trajectory dataset - User Guide*. <http://research.microsoft.com/apps/pubs/default.aspx?id=152176>.

Zheng, Y., L. Zhang, X. Xie, and W.-Y. Ma. 2009. "Mining Interesting Locations and Travel Sequences from GPS Trajectories". In *Proceedings of the 18th International Conference on World Wide Web*. (New York, NY, USA 2009), p. 791–800. ACM.

Zheng, Y., X. Xie, and W.-Y. Ma. 2010b. "GeoLife: A Collaborative Social Networking Service among User, Location and Trajectory". *Data Engineering*, p. 32.