SensibleJournal: A Mobile Personal Informatics System for Visualizing Mobility and Social Interactions

Andrea Cuttone

Kongens Lyngby 2013
IMM-M.Sc.-2013-6

Technical University of Denmark
Informatics and Mathematical Modelling
Building 321, DK-2800 Kongens Lyngby, Denmark
Phone +45 45253351, Fax +45 45882673
reception@imm.dtu.dk
www.imm.dtu.dk IMM-M.Sc.-2013-6

Summary (English)

In this thesis I describe SensibleJournal, a mobile *personal informatics* system for visualizing mobility and social interactions. SensibleJournal proposes novel visualization techniques for *quantified-self* data, including a Spiral Timeline view to analyze periodic movement patterns, and a Social Network view to explore the temporal evolution of social connections. The app has been deployed to N=136 university students during a period of six months. An evaluation based on user survey and on app usage data shows that SensibleJournal motivates users to explore *self-tracking* data, and helps them understanding several aspects of their behaviour.

Summary (Danish)

I denne afhandling beskrives SensibleJournal, som er et *personal informatics* system til at visualisere mobilitet og social interaction. I SensibleJournal foreslås en ny visualiseringsteknik til såkaldt *quantified-self* data som omfatter en tidsspiral til at analysere periodiske mønstre i færden, samt en visualisering af sociale netværk til at undersøge ændringer i sociale relationer over tid. Applikationen er blevet brugt af N=136 universitetsstuderende i en periode på 6 måneder. En evaluering baseret på en spørgeskemaundersøgelse og på logning af brug af applikationen viser, at SensibleJournal motiverer brugerne til at udforske deres egne *self-tracking* data, samt hjælper dem til at forstå forskellige aspekter af deres adfærd.

Preface

This thesis was prepared at the department of Informatics and Mathematical Modelling at the Technical University of Denmark in fulfillment of the requirements for acquiring an M.Sc. in Informatics.

The thesis describes SensibleJournal, a mobile personal informatics system for visualizing mobility and social interactions.

Lyngby, 01-March-2013

Andrea Cuttone

Acknowledgments

I would like to thank my supervisors, Jakob Eg Larsen and Sune Lehmann for the invaluable guidance and feedback throughout the thesis.

I would also like to thank Søren Ulrikkeholm, Arkadiusz Stopczynski and Vedran Sekara for their precious assistance on technical issues.

Finally, I would like to thank my mother, father and sister for their continuous support and encouragement.

Contents

Chapter 1

Introduction

It is a common practice for many people to keep track of personal information. Some people track weight and calories, others their finances. Some like to keep a diary of thoughts and mood, others count how many minutes per day they spend working or playing. The main motivation behind the self-tracking process is to learn something about oneself. Are there any patterns in my daily routine? How did my weight change during the holidays? How do my sleep and mood relate? Another significant reason to track one's life is to make changes in order to improve it. Seeing patterns in behaviour can help us recognize what causes our bad habits, or how to improve our routine.

The *quantified-self* movement[1],[2],[3] proposes the adoption of technology to collect and understand personal data. Nowadays there is a multitude of devices that can automatically measure number of steps, location, activity, blood pressure, glucose level, sleep patterns. Smartphones in particular are an excellent source for self-tracking, since they can automatically record and store data from sensors including GPS, Bluetooth, gyroscope, camera and microphones. Such widespread availability of tools for measuring and analyzing personal data has led an increasingly number of people to use technology for tracking their own

[1] http://online.wsj.com/article/SB122852285532784401.html

[2] http://www.forbes.com/forbes/2011/0425/features-health-personal-data-work-out-taking-my-measure.html

[3] http://spectrum.ieee.org/biomedical/devices/how-i-quantified-myself

lives.

Self-reflection is an essential part of quantified-self. Personal informatics systems can facilitate self-reflection by providing feedback, such as statistics and visualizations. There is a multitude of systems that use the sensor data to provide information about the user behaviour, including energy expenditure, physical activity, sleep quality, and location history. However most tools provide information in form of simple charts and numerical statistics, which are often insufficient for analyzing complex data. There is a need for more complete and advanced feedback forms, in order to provide a deeper understanding of user behavior and allow to gain more useful insights.

In this thesis I describe SensibleJournal, a personal informatics system that uses location and Bluetooth sensor data to provide information about user behaviour. SensibleJournal proposes several novel visualizations to provide insights about periodic activity, movement patterns, and social interactions. This thesis describes the process of data mining and the design of the visualizations. The thesis also investigates how the feedback can be optimized for a mobile platform, and evaluates the effectiveness of the proposed visualizations.

The mobile sensor data used by SensibleJournal is provided by the SensibleDTU[4] project. The goal of SensibleDTU is to record and analyze a high resolution network of social interactions. The data is collected using smartphones which are distributed to participants of the project and uploaded to the SensibleDTU server. The collected data includes location, Bluetooth, SMS, phone contacts, WiFi and Facebook friendships.

As part of the work on this thesis I have written two research papers together with my supervisors. One paper has been accepted for the *CHI 2013* workshop[5] and another paper is currently under review for the *Mobile HCI 2013* conference[6]. The two papers are included in Appendix A and B.

The rest of this thesis is structured as follows. Chapter 2 discusses the related work in the fields of personal informatics, data mining and visualization. Chapter 3 provides a specification of the user requirements and an overview of the system architecture. Chapters 4 and 5 focus on how to visualize respectively personal mobility and social interactions. Both chapters describe the format of the data samples, the process of cleanup and data mining, and the design of the visualizations. Chapter 6 describes experimental setup. Chapter 7 discusses the proposed visualizations and evaluates their effectiveness. Chapter 8 summarizes the contribution of this thesis, and proposes some ideas for future work.

[4] http://www.sensible.dtu.dk

[5] http://personalinformatics.org/chi2013/cfp

[6] http://www.mobilehci2013.org

Chapter 2

Related work

In this chapter I will give an overview of the existing work related to this thesis.

SensibleJournal is a personal informatics system for visualizing personal mobility and social interactions. In Section 2.1 I provide some basic concepts of what personal informatics is, and how it can be used for quantifying personal behaviour. I provide an overview of existing personal informatics systems, focusing on those related to location and social networks.

One fundamental step for producing visualizations is to extract semantic information from data. SensibleJournal uses several data mining techniques on location and Bluetooth data to extract knowledge about user movement and social behavior. In Section 2.2 I give a very brief overview of the related work on mining location and Bluetooth data.

SensibleJournal uses the extracted data to produce several visualizations. In Section 2.3 I explain the basic principles of information visualization and the most common techniques for visualizing movement, periodic patterns and social networks.

2.1 Personal informatics

Personal informatics systems “help people collect personally relevant information for the purpose of self-reflection and gaining self-knowledge” [LDF10]. Self-knowledge has many benefits, including self-control and promoting positive behavior [Rog06, IdKM+06], for example increasing physical activity level.

There are five stages in the self-tracking process [LDF10]:

1. preparation: determine which information to collect, which tools to use and which questions to investigate
2. collection: collect the data either manually or with a tool and store it on paper or digitally
3. integration: process the data to extract relevant trends, patterns and episodes
4. reflection: analyze the integrated data to obtain insights about self
5. action: suggest eventual changes in behavior

Stages are iterative, so it is normal to go back and forth through them multiple times. Each stage has barriers which are difficulties that may arise. Some of the biggest barriers in self-tracking are:

- not knowing what to track or what to expect to gain
- choosing the wrong tools
- having to perform the collection manually. This means that often people forget, get tired, or are inaccurate in their readings
- lack of expertise in how to integrate, process, and reflect on the data

Users of personal informatics system alternate between two modes: discovery and maintenance [LDF11]. When in discovery, users try to make sense of the data and look for insights about their behaviour. During maintenance, users track specific aspects of their behaviour in order to change it.

Most personal informatics system rely on traditional charts for visualizations. Alternative visualizations have been proposed, ranging from abstract art [FFD12, RSH00] to virtual objects [LML+06].

The *quantifiedself.com* website[1] provides an extensive list of more than 600 personal informatics apps, websites and tools. I will give some examples to illustrate the broad variety of aspects that personal informatics can cover.

Nike+[2], *FitBit*[3], *Jawbone UP*[4] and *Basis*[5] produce wearable tracking devices and provide companion apps that show information about calories expenditure and activity level. *Zeo*[6] provides feedback about the quality of sleep. *Mint*[7] is a personal financial management service that provides statistics and visualizations about expenses and earnings. *DailyBurn*[8], *MercuryApp*[9], *42goals*[10], *Quantter*[11] are variations of a personal logging system that allow to keep track of accomplishments, food, activity level, or work. *Mappiness*[12] and *HappyFactor*[13] help users to record their mood.

Several personal informatics tools for location are available. *Moves*[14] is an iPhone app that uses mobile location data to provide information about the user movement patterns and exercise levels. *LifeMap*[15] and *PlacemeApp*[16] provide a map of visited locations, and a summary of time spent at locations. *Google Location History*[17] provides detailed information about movement patterns based on the GPS data collected by Android phones. The app automatically detects locations such as home and work. The user can see a summary of time spent at different locations, or the most frequently visited places. An animation shows movements on the map by tracing the position of the user.

[1] http://quantifiedself.com/guide/tools

[2] http://nikeplus.nike.com/plus/

[3] https://www.fitbit.com/home

[4] https://jawbone.com/up

[5] http://www.mybasis.com/

[6] http://www.myzeo.com/sleep/

[7] https://www.mint.com/

[8] http://dailyburn.com/

[9] https://www.mercuryapp.com

[10] http://42goals.com

[11] http://www.quantter.com

[12] http://www.mappiness.org.uk

[13] http://howhappy.dreamhosters.com/

[14] https://itunes.apple.com/us/app/moves/id509204969?mt=8

[15] https://play.google.com/store/apps/details?id=com.mobed.lifemap

[16] https://www.placemeapp.com/placeme/

[17] https://maps.google.com/locationhistory/b/0/dashboard?hl=en

2.2 Data mining

The process of information visualization involves several steps [Fry07, Ch. 1]:

1. acquire the data and store it in a structured fashion
2. remove wrong samples and reconstruct invalid samples
3. filter out data which is not needed
4. apply data mining and machine learning techniques to extract knowledge
5. represent knowledge visually

Data mining techniques allow to discover knowledge about large amount of data. Two data mining categories are relevant for the considered SensibleDTU dataset: GPS data mining and Bluetooth data mining.

GPS data mining aims at extracting trajectories, stop locations, modes of movement and points of interest from GPS logs. [JGO06] provides a comparison between Least-Squares Spline Approximation, Gaussian kernel and Kalman Filter for data cleanup. [SA09] and [YCP+12] use movement speed to determine stop locations. Clustering is the most used technique to extract points of interest from stop locations. [ZZXY10] uses a grid-based clustering. [PBKA08, BZA12, CCJ10] use modified versions of DBSCAN clustering for determining points of interest. [ZXM10] uses hierarchical clustering. [YCP+12] and [ABK+07] use semantic information to extract points of interest.

In the context of social interactions, Bluetooth mining tries to extract social contacts from Bluetooth scans. Bluetooth detections as an approximation of face-to-face interactions have been employed in many studies. Bluetooth mining has been used for studying social interactions and movement patterns in university campuses [EP06] and conferences [ISB+11]. It has also been used for studying network effects in decision making [API+11], political influence [MFGPP11], epidemiological behavior change [MCLP10] and individual identification [PTH+06]. Community detection from Bluetooth data has been proposed for extracting social subgroups among students [BMP+12, LPDR06] and weak-ties communities from acquaintances [HSL].

2.3 Visualization

In this section I introduce some basic concept about visualization and human-computer interaction. Then I give an overview of existing work related to the visualizations proposed by SensibleJournal: movement visualization, periodic events visualization and social networks visualization.

Information visualization is the process of using text, shapes and colors to answer questions about a complex dataset. When designing a visualization, the starting point is to determine which questions need to be answered [Yau11, Ch. 1]. Typical questions include: what differs from the normal pattern? are there any trends? is there a correlation or causation between variables?
An *insight* is an answer provided by visualizations that leads to a deeper understanding of the dataset being visualized [Nor06]. For example, looking at a line chart gives a simple insight of the increasing or decreasing trend over time. Looking at a flow chart, the user can get the insight of the flow of objects over a geographical space. Insights can be divided into four categories [YKSJ08]:

- overview: get the big picture of the data, that is what the overall data looks like, what is present or missing, what high level characteristic it has
- filtering: identify only the information that matches some specific criteria
- pattern: highlight a trend, groups of objects, or outliers
- mental model: represent the data in a way that matches common human mental models such as a map, a picture or geometric forms

Computer-generated visualizations allow the user to interact and investigate specific aspects of a dataset. A common interaction pattern is “overview first, zoom and filter, then details on demand” [Shn96]. The visualization should first give the overall picture of the data, in order to have a general idea of the content. This is usually done by showing all the data with very little details. Then it should be possible to focus on a subset of the data. This can be done by selecting what is interesting or by zooming on a section of the visualization. When zooming, it is very important to maintain reference to the rest of the data. A common principle is “Focus+Context”, that is to focus on one detail while maintaining references to its context. It should be possible to filter out what is not interesting, or to select what is interesting. This can be done by specifying a more or less complex set of criteria that can range from a few options to complex queries. Finally, detailed information for a specific item can be inspected. This can be realized by showing popups or overlays. Animation is a visualization technique where views of the data at different points in time are shown in rapid sequence. Animation allows to highlight change in data over time [RFF$^+$08].

It is often said that visualization provides "answers to questions that you did not know you had". In some cases, especially for very large data sets with multiple variables, it is hard to generate questions or hypothesis about the data. The fields of Visual Mining [K$^+$02, ZLTX05] and Explorative Data Analysis [Shn01, Beh97] utilize visualization to explore the dataset and spot interesting patterns.

Casual visualization [PSM07] is the application of visualization aimed at a general public with no expertise in data analysis. Casual visualization uses data which is personal to the final user, such as his movements log or his financial records, and provides insights about habits and life patterns. News articles, websites and personal informatics systems are increasingly employing casual visualization for providing data analysis.

Much work exist on the visualization of location data. The most common visualizations are based on geographical maps. Flow maps allow to represent movement of objects or information, for example goods on roads or data on a network. [PXYH05] describe an algorithm for generation of flow maps, a layout for avoiding overlapping edges, and a clustering to group symbols in complex flow maps. [Rae09] uses flow maps for a case study of the population migration from the 2001 UK census.
When visualizing spatio-temporal data, the three variables to be considered are: where, when and what [AAG03]. Using combinations of these variables, several questions can be asked:

- which objects exist given a location and a time
- where are objects at a given time
- at what time objects are located at specific locations

To represent location changes over time, four techniques are common:

- static representation
- snapshots in time
- movement history from the start to the current instant
- a time window showing part of the route in an interval of specified length, represented as chain of locations

Spirals as visualization for periodic time-series data are proposed in [WAM01]. It is shown how a spiral can much better highlight periodic patterns. Color, texture, style, thickness, symbols are proposed for mapping data on the spiral. Multiple data sources can be represented at the same time for comparison. Several interactions are suggested, including brushing a section of the spiral, and allowing to vary the period. [SSOG08] proposes a framework for visually analyzing business events. Events are represented using glyphs on concentric circles. Different placement strategies, glyph grouping and coding are proposed. The clustering of glyphs is proposed for visualizations containing large number of events. Several procedures of visual analysis are suggested to infer causal relations between events and to detect anomalies. [DH02] uses a spiral as clock metaphor for representing periodic time events. Some example applications are given: a bus schedule and an events calendar. [HHI99] combines a spirals with a map to visualize periodic patterns in GPS data.

There are two main ways to visualize social networks: as graph and as matrices [GFC05]. A social network can be imagined as a graph, where nodes represent people, and links represent their relations. Visualizing very large networks is hard, since links and nodes tend to overlap. In a matrix representation, each person is assigned a column and a row in a NxN matrix. If there is a relation between two people A and B, then the cells at the intersection between row A and column B and row B and column A are marked. *SocialAction* [PS06] visualizes social network using graphs, lists, and scatter plots. Nodes can be filtered according to various attributes. Social groups are extracted using a community detection algorithm and nodes are grouped accordingly. *Live Social Semantics* [ASC+09] represents face-to-face interactions data using bubbles and lines. *Vizster* [HB05] represent the social network as a graph with force-directed layout. The graph is dynamically expanded as nodes are clicked. It is possible to filter by different parameter. Communities of various granularity can be automatically detected and displayed using colored areas. *NodeTrix* [HFM07] integrates the graph and the matrix into one visualization. Subnetworks are shown as matrices, which are connected by graphs.
The popular professional network website *LinkedIn* recently released a tool for visualizing professional networks as a graph with distinct communities[18]. *WolframAlpha* provides a similar visualization tool for data extracted from *Facebook*[19].

[18] http://inmaps.linkedinlabs.com/

[19] http://blog.wolframalpha.com/2013/01/23/introducing-expanded-personal-analytics-for-facebook/

Chapter 3

Design

This chapter covers the general architecture of the system. Section 3.1 defines the requirements of the system, and motivates the visualizations I proposed. Section 3.2 describes the general design choices for the UI. Section 3.3 describes the communication with the SensibleDTU server. Section 3.4 describes a tool used for collecting usage data. Finally, Section 3.5 addresses the possible privacy concerns.

3.1 Requirements

The goal of SensibleJournal is to provide insights about the user behaviour, based on the sensor data. Several different sensors could have been used for this purpose. The current work focuses on the two sensors which are most interesting in terms of understanding user behaviour: location and Bluetooth. Location data describes the user moving patterns, while Bluetooth can be used to infer social relations. Many different kinds of information can be extracted from these two data sets.

The choice of what kind of information to present is determined by the end-users requirements. The users of SensibleJournal are students between 20 and

25 years old, with little or no training in data analysis tasks. Most users do not have a specific data analysis goal, but they want to explore their data in order to find interesting insights about their own behaviour. For these reasons, visualizations should be simple enough to be understandable by a non-expert user. Moreover, visualizations should provide a broad overview of the data, give a few factual insights and support exploratory data analysis.

I identified a set of possible questions, and a set of visualizations that try to answer these questions.

For location data, questions include:

- how much time did I spend stationary, walking, or on a vehicle?
- what are my most commonly visited locations?
- are there periodic patterns in my movement?

I proposed the following visualizations:

- a Stats view that shows statistics on the time spent stationary, walking, or on a vehicle
- a Movement view that displays movement patterns and top locations
- a Spiral Timeline view that displays periodic movement patterns on a spiral

For Bluetooth data, questions include:

- how do the contacts with other participants evolve over time?
- are there any social subgroups in my social network?
- who have I spent most time with?

I proposed the following visualizations:

- a Social Contacts view that shows the list of most common social contacts
- a Social Network view that displays the evolution of user social network over time

3.2 UI design

The general UI structure of the app is a tabbed interface, where each tab contains a different visualization. This layout gives an easy access to all visualizations and makes it easy to spot at a glance which visualizations are available.

The usage of mobile apps differs from the usage of standard software applications or websites in several ways [Tid10, Ch. 10]:

- usage sessions are shorter, and users expect to get access to the needed information very quickly
- screen space is very limited
- interaction is performed using the touch screen instead of keyboard and mouse

I took into consideration some principles when designing the UI and the visualizations:

- quick information loading. I made a particular effort in making sure that the performance of the visualizations is acceptable. Moreover, data is loaded progressively and in the background, to start giving the user some feedback right away
- optimal screen usage. All visualizations try to present an overview of the information, then allow to zoom into a subset of the data, and show details on demand. The presence of UI widgets is kept to minimum, in order to reduce clutter
- touch interface interaction. The visualizations support the common touch interface interaction gestures (pinch, scroll, tap). I attempted to minimize the number of steps for interaction

I added some simple tutorials in order to help the users understand how the app works and how to use the visualizations. The first time the user opens a visualization, one or more example screenshots with text overlays explain the basic functionalities and interaction patterns. Moreover, I provided a user guide on the SensibleDTU website[1].

[1] http://www.sensible.dtu.dk/?page_id=805

3.3 Server communication

SensibleJournal needs to access the collected data on the SensibleDTU server. An access token is needed in order to retrieve the data. For requesting an access token, the user must log-in with username and password. Once an access token is obtained, the developer API can be used to retrieve the data from the user associated with the access token. All results are provided in JSON format.

Many of the proposed data mining and visualization techniques require intensive computations. Performance-wise it would be much more efficient to perform these computations offline on an external server, and make SensibleJournal request only the computation results. This would have several advantages:

- data processing could be done offline, and then the only time needed would be for retrieval of data
- data processing would be much faster on an external server with more powerful hardware than a smartphone
- requesting only the computation results would reduce the size of the data that needs to be transferred. This would reduce network delays and bandwidth consumption

However, the use of an external server would have conflicted with the privacy principles of SensibleDTU. One of the core principles is that each user should have access only to his own data, unless he would grant special permissions to other users. Having an external server would require to have access to all data of all users, and store this sensible data. This would have conflicted with the privacy principles described above. For this reason, SensibleJournal performs all the computations at client-side (directly on the phone). This guarantees that the sensible data is not stored in any other systems.

There are three aspects I considered regarding the client-server communication: request time, network usage and offline access.
Request time represents the time needed for the server to reply with the requested data. It includes a fixed cost and a variable cost. The fixed cost is due to network latency, and the time needed for opening and closing the connection. The variable cost is proportional to the size of records to be returned: a bigger set of records takes more time to be transferred. For the user, it is important that the request time is minimized. This is especially true for mobile users, which have shorter interaction time with applications compared to desktop users.

The network usage is the amount of data transferred over the network. For mobile users, this amount should be minimized for two reasons. First, many users may have limited bandwidth, so a large amount of data could take a lot of time to be downloaded. Second, many users have a cap to the amount of data usage for mobile networks, so apps that require a lot of data are not well received.
Offline access is the capability to access to the data when the user has no network connectivity, or when server cannot be reached.

To optimize these three aspects, I implemented a caching mechanism. Once SensibleJournal fetches the data from the server, it caches it into a SQLite database on the phone. This avoids the need to access the server every time, and dramatically speeds up the data access since the database access time is almost instantaneous. Moreover, data can be accessed while offline. In order to minimize the number of requests, SensibleJournal requests data in chunks of size of one week. The data collector may upload new samples at unknown times, so the server must be checked periodically for new data. For this reason, SensibleJournal maintains a table with week number and last time of checking, and periodically refreshes the data from the server every 24 hours.

3.4 Usage analysis collector

One of the goals of SensibleJournal is to understand how effective the provided visualizations are. A way to empirically measure the effectiveness of a visualization is to count the amount of time that the user spends on that visualization. We can assume that users tend to spend more time on visualizations that they found more interesting.

For this purpose, I implemented a simple event logging system. Each time the user switches to a different part of the application, for example to the Movement view, the logging system creates an entry in an event log database with an anonymized user id, the name of the event and a timestamp. The logging system creates an event also when the user sends the app to background or locks the screen. This custom logging mechanism has several advantages:

- it is active only for SensibleJournal, and does not consume resources while the app is in background
- it allows to have an arbitrary granularity for event collection. For example, it can log events when a user select a visualization, when a widget is selected, or when a button is clicked

The logging system temporarily stores the events in a SQLite database on the phone, and periodically uploads them to a central server. The uploads are incremental, since the app sends only the records after the latest uploaded one. All usage data is anonymized, so no privacy concerns exist.

I provide an analysis of the collected usage data in Section 7.3.

3.5 Privacy issues

Important privacy issues arise when treating personal data. SensibleJournal makes sure that data privacy is respected:

- each user can access his own data only
- the app does not store any credentials on the phone
- the app communicates with the SensibleDTU server over a secure HTTPS connection
- the app periodically checks the access token for validity, and logs out the user when the access token expires

Chapter 4

Visualizing personal mobility

This chapter describes the process of visualization of location data. Section 4.1 describes the data format, and which errors may affect the location samples. Section 4.2 and 4.3 describe several strategies for removing invalid samples. Sections 4.4, 4.5 and 4.6 introduce some data mining techniques for extracting mode of movement and points of interest. Finally, in Sections 4.7, 4.8 and 4.9 I describe the implementation of three visualizations: Movement, Stats and Spiral Timeline.

4.1 Location samples

On Android phones, the location data can come from two providers: network and GPS[1]. The network provider uses WiFi and cell tower data to determine the location. The network provider can determine a location quite rapidly and has low battery consumption, but it is less accurate than GPS. GPS is a positional system that uses satellites orbiting around Earth. GPS position data is subject to several errors, including satellite orbit errors, clock errors and atmospheric

[1] https://developer.android.com/guide/topics/location/strategies.html

```
{
    timestamp: 1351609870,
    type: "person",
    location: {
        latitude: 55.7893137,
        longitude: 12.5312681,
        accuracy: 80
    }
}
```

Figure 4.1: Format of a location record

conditions [OGB+02]. GPS suffers from the *cold-start* problem, which is the time delay needed to synchronize with satellites at the beginning of the detection. Moreover, the receiver must be in line of sight of satellites, which means that GPS does not work indoor. The GPS provider is much more accurate than the network provider, but it requires more time before the location can be determined.

Figure 4.1 shows the format of a location record provided by the developer API. `timestamp` refers to the instant when the record was created. `latitude` and `longitude` are the geographical coordinates from the GPS sensor. `accuracy` is the GPS positioning accuracy in meters at the time of collection.

4.2 Data cleanup and mining

The SensibleDTU collector records one location sample approximately every 10-15 minutes. Due to the difficulties in obtaining a precise location, some data collected by the SensibleDTU collector may be inaccurate. In order to improve the quality of the location data, some cleanup needs to be done. Before going into details about the data processing, it is useful to define some basic concepts.

I calculate the distance between two locations samples using the haversine formula[2]. Location samples contain information about geographical position but also about time. If we consider two consecutive locations samples l_1 and l_2, we can imagine that the user has travelled from l_1`.location` to l_2`.location` in a time interval l_2`.timestamp - `l_1`.timestamp`. The speed between two location samples is the distance between the samples divided by difference in timestamp of the two samples.

[2] http://www.movable-type.co.uk/scripts/latlong.html

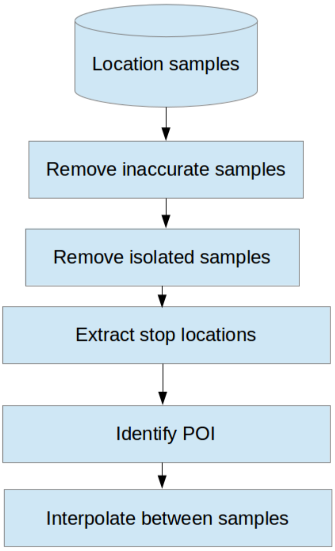


Figure 4.2: The complete process of location data mining

Figure 4.2 summarizes the complete process of location data mining.

4.3 Invalid samples removal

A first step in the cleanup process is to discard all samples with accuracy greater than 100 meters, since they would be too inaccurate to represent a meaningful location.

Another category of errors in the location data is the presence of single isolated data points far from all the others. I have not determined yet why these errors are present. Presumably, they are due to temporary inaccuracies of the Android location service, which would not find a GPS position and then would use a very approximate location from a cell tower. They could be also due to the reflection of GPS signal in urban areas making the detected position jump to distant location [SA09]. Nevertheless, an approach for filtering out these isolated locations was needed, otherwise the movement visualization could be negatively affected. I considered several approaches. Figure 4.3 provides a side by side comparison of the results of the different approaches. The leftmost panel represents all locations as blue dots. The dot in the top-right corner near Klampenborg is the

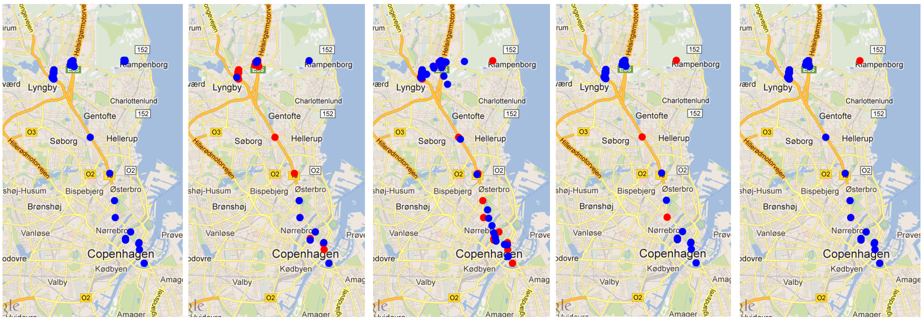


Figure 4.3: Comparison of different cleanup techniques. In the leftmost panel, the blue dots represent all the samples, while the dot on the top right near Klampenborg represents an invalid sample. In the other panels, the blue dots represent the samples considered valid and the red dots the discarded samples. The panels from II to V show the result of the following techniques: Douglas-Pecker, Gaussian smoothing, DBSCAN and a speed-based heuristic

erroneous sample to be eliminated.

The use of a smoothing algorithm such as Gaussian smoothing or Douglas-Pecker [DP73] is not appropriate, since it would modify the other locations to fit the existing data points. Figure 4.3 shows that for Douglas-Peucker (the second panel) the wrong samples are kept, and other valid points are eliminated (the red dots represent discarded samples). Panel III in Figure 4.3 shows that when a Gaussian smoothing is applied, the algorithm modifies smoothed locations (the blue dots) in comparison to the original locations (the red dots).

A clustering algorithm such as DBSCAN [EKSX96] could be used to extract and discard isolated clusters. Panel IV shows the result of DBSCAN, with red dots as discarded samples. DBSCAN successfully identified the erroneous location as isolated cluster. Unfortunately, DBSCAN also marked as an isolated cluster some locations which were quite distant from each other, but were actually valid data. Due to the low frequency of sampling, if the user is in rapid movement (for example on a car), several locations can be quite distant from each other. In these cases, DBSCAN failed to understand that they were valid data points.

The approach that I finally adopted was a simple heuristic based on the observed data. I noticed that, for some test datasets, each time an invalid location B was recorded, it would be preceded by a valid location A, and followed by another valid location C. If we imagine a user moving according to this sequence of locations, we would see him starting at A, running far away to B, and then

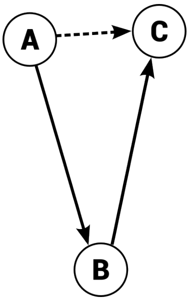


Figure 4.4: Schema for the invalid samples removal heuristic. An invalid location B is detected because the speed of movement from A and to C is very high, while the speed of movement between A and C is normal.

running back to C and close to A. Thus the defining characteristic of the invalid location B is that movement speed between A and B is very high, and the movement speed between A and C is much lower. Figure 4.4 represents this situation. Panel V in Figure 4.3 shows that the heuristic discards only the invalid sample (marked in red). I tested this heuristic with many datasets and I have observed that it filters most of the invalid locations while keeping almost all valid ones.

4.4 Detection of movement mode

As described before, it is possible to calculate the speed between two locations samples by dividing the distance between the locations by the timestamps difference. I use this speed to estimate the movement mode of the user in that time frame:

- `0` ≤ `speed` ≤ `walking speed`: user is stationary
- `walking speed` ≤ `speed` ≤ `vehicle speed`: user is moving on foot
- `speed` ≥ `vehicle speed`: user is on a vehicle e.g. train, bicycle, car

where walking speed is 0.5 m/s and vehicle speed is 3 m/s. I chose these values by tweaking the average human walking speed[3] to optimize the accuracy of categorization on several test datasets.

[3] https://en.wikipedia.org/wiki/Preferred_walking_speed

I assume that the user has spent the time between the two location in that movement mode. If this classification is repeated for all pairs of consecutive locations, a breakdown of time spent in the three movement modes is obtained.

4.5 Stop locations and Points Of Interest

The stationary movement classification is of particular interest. When the user is walking or on a vehicle, it means that he is commuting, that is, moving between different locations. When the user is stationary, he is spending time at a specific location, which in most of the cases is a *point of interest* (POI) for the user.

A POI is a location of relevance for the user, for example his home or his workplace. Defining what a POI is depends on the chosen granularity. A POI could be a street, a building, or an entire area such as a park or a square. For the purposes of SensibleJournal, a POI is defined as an address, such as “Anker Engelunds Vej 1”.

The time spent at POI is a very good indicator of the importance of a location. For many people, it is common to spend 7 or more hours during the evening and through the night at home, and spend 5-8 hours during the day at the workplace, at the university or at school. Moreover, it is typical to spend significant amount of time in locations such as friends’ and relatives’ apartments, or public places (e.g. gyms, bars, parks, libraries).

I use the term *stop location* to indicate a location where the user had a stationary movement mode. A stop location is described by latitude and longitude that describe the position, and a start timestamp and a duration that describe the period of time when the user was stationary at that location. I extract stop locations from all locations samples by finding the locations classified as stationary.

When the user is at a POI, the SensibleDTU collector records location samples all around that POI. This is first because of the inaccuracies of the location data as discussed before. Second because the user may slightly move around that POI: image a user walking around a building or moving inside. All stop locations around a single POI must be logically grouped together. The problem of grouping together similar samples is known as clustering. For locations, samples similarity is given by the distance, and clusters represent POI. I used a DBSCAN implementation to assign locations to clusters. Once stop locations are assigned to a cluster, the app determines GPS coordinates for each POI, calculated as

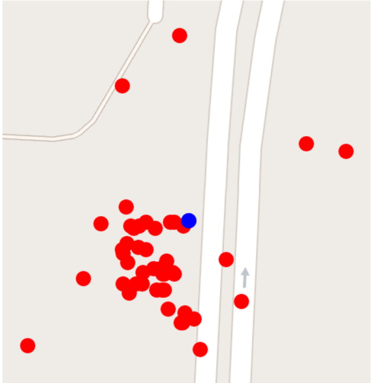


Figure 4.5: Result of clustering. The red dots are the original samples, and the blue dot is the resulting cluster center.

the average of all other GPS coordinates in that cluster. Figure 4.5 shows an example of the result of clustering: the red dots are the samples in a cluster, and the blue dot represents the clustered sample. The app reverse-geocodes the POI GPS coordinates to obtain a street address. The app calculates the total time spent at a POI by summing all durations of stop locations related to that POI.

4.6 Interpolation

The location samples represent snapshots of the user position over time. These snapshots are distant 10-15 minutes in time. In order to show a detailed history of locations on the map, a higher granularity of data is needed, so that two consecutive points are not too distant. This is especially true during the time frames when the user is moving rapidly, for example on a train or on a bus. In these cases, representing the points on a map would be perceived as “jumps” between locations. We can make the assumption that the movement between two locations at close distance and recorded in a short interval of time ($<$ 10-15 minutes) is typically a straight line at constant speed. Under this assumption, I estimated the intermediate positions during that time frame using linear interpolation.

The interpolated locations are generated at time intervals of 120 seconds. After having tested this amount of locations, I observed that it is enough for representing a detailed movement path, and not too much to slow down the processing. Figure 4.6 shows an example of the result of interpolation. The original loca-

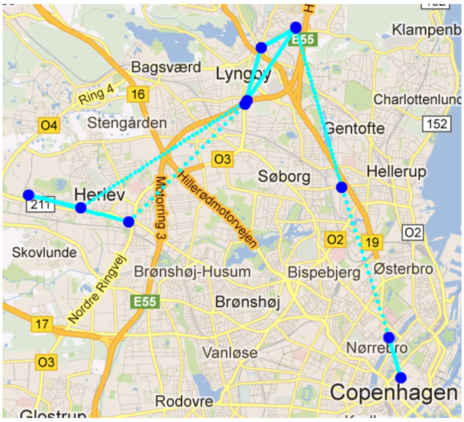


Figure 4.6: Result of interpolation. The big blue dots are the original samples, and the smaller light-blue dots are the interpolated samples.

tions are represented by blue dots, and the interpolated locations by light-blue dots.

4.7 Movement view

The Movement view (Figure 4.7) shows the user mobility during the day. The goal of this visualization is to provide the following insights:

- are there any patterns in my movements?
- how does my location vary over time?
- in which locations do I spend the most of my time?

The visualization is a geographical map with overlays to describe the movement. The app displays the data as arrows that connect the locations of the path, in order to give the overview of the movements for the day.

When clicking the play button, the app displays an animation to show the change in position of the user over time. The animation shows the user location as a red dot, and the current time of the day on the top right part of the screen. As time progresses, the dot moves to the location on the map corresponding to the location of the user. The dot leaves a trail behind it, in order to make

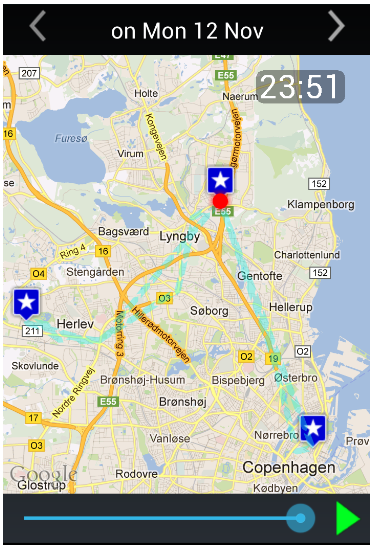


Figure 4.7: The Movement view

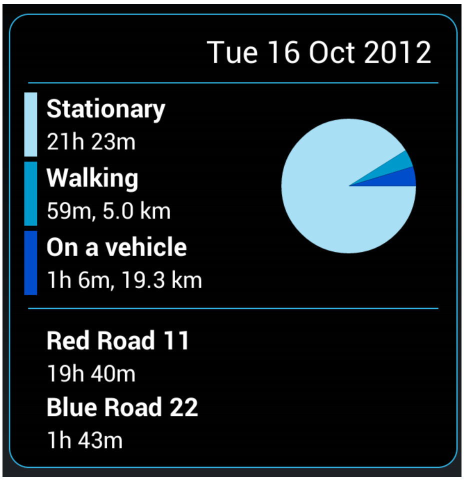
it easier to identify the movement path. A progress bar at the bottom part of the map shows the progress of the animation. Changing the progress bar value allows to navigate the animation, similarly to how videos are navigated on video players.

The map has icon overlays for the POI. Clicking on a POI icon displays a pop-up balloon with the address of the location and the total time spent. The map supports the standard interaction patterns of map apps such as *Google Maps*[4]. The pinch gesture is used to zoom in and out. The scroll gesture is used to pan the map. When navigating the map, all the overlays are moved and scaled accordingly. At the top of the map, a date selection widget allows to pick the date to inspect. Scrolling left or right decrements or increments the date, and double-tapping opens a date selection dialog.

4.8 Stats view

While the Movement view aims to give qualitative insights about the movement behavior of the user, the Stats visualization (Figure 4.8) aims to give more quantitative insights.

[4] https://play.google.com/store/apps/details?id=com.google.android.apps.maps



Tue 16 Oct 2012

Stationary
21h 23m

Walking
59m, 5.0 km

On a vehicle
1h 6m, 19.3 km

Red Road 11
19h 40m

Blue Road 22
1h 43m

Figure 4.8: The Stats view

The visualization is a list of days in reverse chronological order, from today towards the past. When scrolling the list, more days are loaded and displayed.

For each day, the Stats visualization gives some statistics. It displays the time spent in the three movement modes (stationary, walking, and on a vehicle), and visualizes the percentages using a pie chart. The visualization also shows the distance covered by walking and on a vehicle, and the top three POI, with address and estimated time spent at the location.

4.9 Spiral Timeline view

Human movement behavior is surprisingly regular and predictable [SQBB10, JLJ+10]. Most people spend the majority of their time at a few common locations, and they move between these locations with regularity.

Linear timelines represent events in a sequence and are not able to show the periodic nature of movement behaviour. To overcome this limitation I propose a Spiral Timeline view, which visualizes the stop locations on a spiral, in order to highlight periodic movement patterns (Figure 4.9). The goal of this visualization is to provide the following insights:

- are there any recurring patterns in my movements?

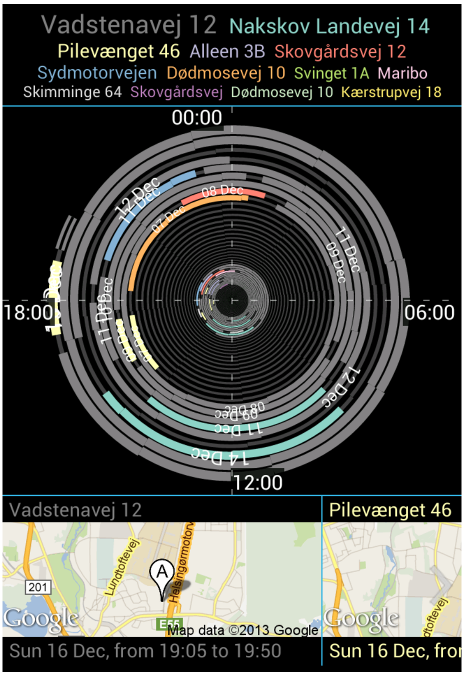

Vadstenavej 12 Nakskov Landevej 14
Pilevænget 46 Alleen 3B Skovgårdsvej 12
Sydmotorvejen Dødmosevej 10 Svinget 1A Maribo
Skimminge 64 Skovgårdsvej Dødmosevej 10 Kærstrupvej 18
00:00
06:00
12:00
18:00
12 Dec
08 Dec
07 Dec
11 Dec
09 Dec
12 Dec
11 Dec
14 Dec
Vadstenavej 12
Pilevænget 46
201
Lundtoftevej
Helsingørmotorv
A
Google
E55
Map data ©2013 Google
Sun 16 Dec, from 19:05 to 19:50
Sun 16 Dec, fro

Figure 4.9: The Spiral Timeline view

- do I tend to visit the same location in particular times of the day, or days of the week?

A spiral is a two-dimensional geometrical curve that revolves around a central point, continuously increasing its distance from it. A logarithmic spiral is defined by the following equations:

$$x(t) = ae^{bt}\cos(t)$$

$$y(t) = ae^{bt}\sin(t)$$

a and b are real numbers > 0. a determines how large is the spiral, and b determines how quickly the distance between arcs grows.

The spiral is a timeline that begins at the present moment and goes backwards in time. The present corresponds to the outermost arc of the spiral. The spiral then goes counterclockwise, and inner arcs represent the past. A full arc (2π) corresponds either to 24 hours or 7 days, depending on the chosen period. The spiral arcs represent stop locations on the spiral. The start and end time of each stop location correspond to angles on the spiral. The following formula converts a timestamp t to an angle in radians:

angle = start_angle - $2 * \pi *$(now - t) / period

now is the timestamp of the current instant, period is the number of seconds in 24 hours or 7 days (depending on the chosen period), start_angle is a constant corresponding to the angle on the spiral for t = now.

Since the outermost arc represents the present and the spiral is read counterclockwise, the timestamp t = now corresponds to start_angle, and timestamps farther in time correspond to lower angles. A difference of period seconds corresponds to a full circle. Using this process for representing the stop locations on the spiral, arcs corresponding to locations with the same periodicity are drawn in the same sections of the spiral. This allows to visually spot locations which have the same period.

The visualization assigns each different POI to a different color which is used when drawing arcs representing time spent at that POI. A simple algorithm for color generation is just to use random values for the RGB components of colors. However, the use of random colors corresponds to a rainbow palette, which is not optimal for visualizations. In fact the human eye perceives some colors with more strength than others, thus a rainbow palette distorts perception of the visualization [BT07]. To solve this problem, I adopted a qualitative color scheme suggested in [HB03].

The arcs on the spiral are decorated with the date of the corresponding stop location. The size of the text is also scaled to be proportional to the thickness of the arcs. When the text size would become too small to be read, the text is not displayed at all. This allows to have a scalable amount of details depending on the current level of zoom. An additional overlay with hour of the day or day of the week is shown on top of the spiral.

The spiral supports touch gestures. The pinch gesture changes the zoom factor, so that inner arcs become thicker and more details about the past events are revealed. The scroll gesture pans the spiral, allowing the user to view all parts of the shape when it does not completely fit the screen. The double-tap gesture changes the period between 24 hours and 7 days. The combination of the gestures allows the user to focus into the details of a specific period, or to have a bigger overview of the data.

One significant challenge for this visualization is to give an overview of the periodic patterns, while still being able to focus on a particular time period. Using a logarithmic spiral, the distance between the arcs progressively increases with the angle in geometric progression. The outer arcs are drawn thicker, and become progressively thinner as they approach the center. This has the effect that the periods on the outer arcs (that is, the ones closer to the present) are bigger in size and can be drawn with more details than periods in the inner arcs (that is, the ones farther from the present).

This view also gives two complementary pieces of information: a tag cloud and a linear timeline of events.
The top part of the screen contains a tag cloud of the POI. A tag cloud is a set of text labels, where the size of each label is proportional to a value associate with the label. In this case, the size of each label is proportional to the total time spent at that POI. The tag cloud uses a logarithmic scale, since the time at the top POI is usually much bigger than the other times. The color of labels is the same color used in the spiral for the POI.
The bottom part of the screen shows a timeline of events in reverse chronological order. Each element of the timeline describes a stop location, with the start and end time, an address and a static map. The color of the text and of the map marker are the same color used in the spiral for the corresponding POI. The scroll gesture moves between items of the timeline. Tapping on a timeline item marks it as selected on the spiral by filling it with a black bar.

Chapter 5

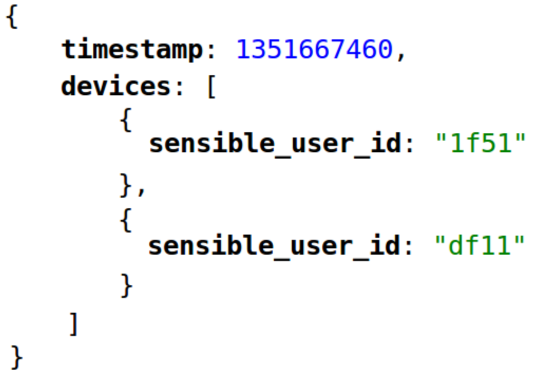
Visualizing social interactions

This chapter describes the process for visualizing social interactions, based on Bluetooth data. Section 5.1 describes the format of the Bluetooth records. Section 5.2 describes how to infer social interactions from Bluetooth logs. Section 5.3 describes how social groups can be extracted from the social interactions. Sections 5.4 and 5.5 provide the details for the implementation of two visualizations: Social Contacts and Social Network.

5.1 Bluetooth data

Bluetooth is a protocol for short distance wireless communication. Bluetooth devices can perform a discovery, which detects all discoverable Bluetooth devices in the range of 5-10 meters.

Since each device corresponds to a specific user, a Bluetooth device and his owner can be used interchangeably. I use the term *current user* to indicate the user that is detecting the other Bluetooth devices. I use the term *detected user* to indicate a user that has been detected by the current user.



```
{
    timestamp: 1351667460,
    devices: [
        {
          sensible_user_id: "1f51"
        },
        {
          sensible_user_id: "df11"
        }
    ]
}
```

Figure 5.1: Example of a Bluetooth record

The SensibleDTU Bluetooth probe periodically scans for Bluetooth devices. Each scan produces one record, including a timestamp and a list of all devices detected. Figure 5.1 shows an example Bluetooth record provided by the developer API.

`timestamp` refers to the instant when the scan was performed. `devices` is a list of Bluetooth devices detected by the current user. Each `sensible_user_id` is the id of a detected user.

5.2 Quantifying social interactions

The Bluetooth data provided by the SensibleDTU API contains Bluetooth scans from the point of view of the current user. This means that all the records refer to Bluetooth detections of other users by the current user. Each record is a snapshot of the nearby users at a given instant of time. If the current user detects another user, it means that this second user is in close proximity. This close proximity can correspond to several cases:

1. the detected user had a face-to-face contact with the current user, which means that the current user was talking or interacting with the detected user

2. the detected and the current user were in the same environment, for example sitting or standing nearby, but not interacting face-to-face

3. the detected user was located nearby but not in direct relation with the

current user. For example, the detected user was sitting in a room adjacent to the current user room

4. the detected user was just walking by, and it happened to be detected at that point by the current user

Case 1 and 2 imply a social interaction between the current and the detected user. Case 3 does not imply a direct social interaction, but still represents some relation between the current and the detected user. Case 4 represents noise in the Bluetooth data, but it has quite low probability to happen compared to the other cases. In conclusion, the Bluetooth proximity can be used as a rough estimation of the social interaction between users.

One way to measure social interactions is to quantify how many times there has been contacts with a specific user. Using Bluetooth proximity, it is reasonable that the more frequently a user is detected, the more social contacts there have been with that user.

The simplest method to quantify this interaction is to count how many times each user is detected. This count gives an estimate of the time spent with the detected user. This method is subject to errors since the number of Bluetooth detections is not always accurate. In some scans, Bluetooth devices may not be detected. This could happen because the user momentarily moves away, or because the Bluetooth signal is blocked for some reason. In some other cases, there could be more samples for the same interval of time. In these cases, the simple counting method would lead to a very imprecise estimation.

A simple improvement to the counting method is to group Bluetooth records into time slots or *bins*. Records are put into bins depending on their timestamp. Each bin contains all records in a specific interval of time. The choice of the interval can vary, and depends on the frequency of the scans. In the current implementation, bins of size 300 seconds are used. Each bin is then processed separately. All users in a bin are considered seen together in a group meeting, and their counter is increased. If a user is present multiple times in a bin, (s)he is counted only once. This prevents that missing Bluetooth scans influence the count of contacts.

This naive binning method is unable to determine meaningfulness of detections. When the current user is in public places, the bins are likely to contain a large number of users. For instance, at the DTU canteen or at the library it is likely to observe a large number of users. All these detections have little or no significance, since the detected users probably had no social interactions with the current user. On the other hand, bins that contain just a few users have much larger significance. When the current user detects a single other user, it is likely that the two are having a face-to-face contact. A simple heuristic to take into account these differences is to assign to social interactions a weight inversely proportional to number of users in the bin. For example, if a bin contains only A, then the social interaction estimation for A is increased by 1. If a bin contains A, B and C, then the social interaction estimation is increased by 1/3.

5.3 Estimating communities

The Bluetooth proximity is an indication of co-presence of the current user and the detected users. This co-presence can be also used to estimate relations between detected users. In fact, if two users are detected in the same Bluetooth scan, then they are likely to be in proximity of each other. Thus detected users in the same bin are likely to have some social interactions between each others as well.

A way to represent this information about social relations is to use a social network graph, where users are nodes and social interactions are links. If users are present in the same bin, a link between all pairs of users in the bin is added. A more precise representation is to use an undirected weighted graph, where the link weight is an indication of the strength of this social interaction. As already discussed, the inverse of the number of users in a bin can be used as metric. For instance, image a bin containing users A, B and C. A graph can then be constructed with A, B and C as nodes, and links AB, AC and BC of weight 1/3.

A variety of data can be extracted from social network graphs, including how people are divided into social communities [Fre04]. A community is a subset of the graph in which nodes are more connected internally than to nodes outside of the community. In social network graphs, communities correspond to social groups. For example, in a Facebook network graph, there may be a community of people from work, one of people from school and so on. Running a community detection algorithm on a social graph allows to assign users to different communities. Several community detection algorithms have been proposed [New04, APF+06, ABL10, LN08]. Among them, I chose the *Louvain* algorithm [BGLL08], since it supports weighted graphs and has very good performance [LF09]. The algorithm is based on the idea of creating partitions by iteratively maximizing modularity. I made an Android version adapted by an existing Python implementation[1]. One significant challenge in the implementation was the optimization of objects allocation, which was significantly slowing down the

[1] https://bitbucket.org/taynaud/python-louvain

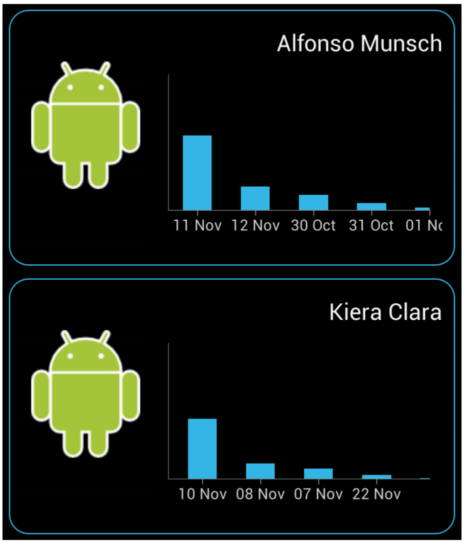


Figure 5.2: The Social Contacts view provides a list of social interactions, with names and days of meeting.

algorithm. To overcome this problem, I applied the *object pool* design pattern using the *Apache Commons Pool* library[2].

I have now extracted two kinds of information from the Bluetooth data: an estimate of social interaction and a community membership for each user. I use this information to provide two visualizations: Social Contacts and Social Network views.

5.4 Social Contacts view

The goal of this visualization is to provide the following insights:

- who do I spent the most time with?
- which are the days where I see determinate people more often?

The view presents a scrollable list of contacts. Each item provides the name of the user, a profile picture and a bar graph with the days of social interaction,

[2] https://commons.apache.org/pool/index.html

sorted by decreasing order. The items on the list are sorted by total frequency of contacts, by decreasing order. This way, the most frequent contact are shown at the top of the list. Figure 5.2 shows an example.

5.5 Social Network view

The goal of this visualization is to provide the following insights:

- who do I spent the most time with?
- how do my social interactions vary over time?
- are there any sub-groups in my social network?

The visualization is shown in Figure 5.3. The visualization is in form of animation composed by a sequence of days, starting from 1 October 2012 (the start of SensibleDTU project) to the present day. The bottom part of the screen shows a progress bar with a scrollable cursor. Each step of the bar corresponds to one day. A secondary progress bar indicates the number of days in the animation which are loaded and ready to be displayed. The progress bar works similarly to streaming video players such as Youtube, where the current and buffered progress is shown. A play/pause button allows to control the animation playback.

Each step in the animation represents the social network as a bubble graph for a specific day, where each bubble represents a different user. The radius of each bubble is proportional to the social interaction estimation for the corresponding user, and the color of each bubble represents the community that it belongs to. The day is displayed on the top left corner.

The animation supports two layout modes: *communities clusters* and *sorted grid*. In communities clusters mode, all bubbles of a community form a separate cluster. Each cluster occupies a different position on the screen. Bubbles positions are calculated so that they orbit around the center of their cluster and do not overlap between each other. This is realized using a circle packing algorithm[3]. In sorted grid mode, bubbles are arranged on a two-dimensional grid and sorted by decreasing social interaction estimation, from left to right, from top to bottom.

[3] http://wiki.mcneel.com/developer/sdksamples/2dcirclepacking

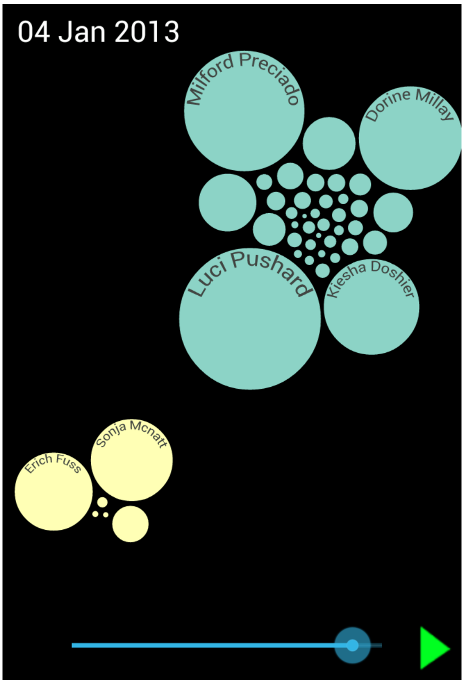


Figure 5.3: The Social Network view shows social contacts as a bubble graph. Bubbles radius represents the social interaction estimation, and bubble color represents the community.

Bubbles are decorated with the name of the user. The text is scaled according to bubble size and zoom level. If the text is too small, it is not shown at all. This way, the user can see more details as he progressively zooms in. The visualization supports gestures for navigation. The pinch gesture allows to zoom, and the scroll gesture allows to pan the canvas. Double-tapping changes between the two layout modes.

5.6 Privacy concerns

One of the privacy principles of SensibleJournal is to provide access to personal data only to his owner. The data mining techniques introduced in this chapter allow to infer information about self, but also about other people. In particular, the inference of social community membership creates connections between other people, even though the current user has no access to samples that record these connections. These estimations are based only on the data accessible by the current user, and are in principle equivalent to what the current user could see in real life. This means that the app reveals no private or sensitive information about other users. The app provides only information that the current user could have observed in person.

Chapter 6

Experiment

This chapter briefly describes the experimental setup used by SensibleJournal, including deployment, data collection and user survey.

6.1 Deployment

Starting from October 2012, SensibleJournal was deployed to N=136 first-year DTU students. All participants were provided with a Galaxy Nexus smartphone. SensibleJournal is available on the official Android Play Market[1]. The distribution on the official market greatly simplifies the release of new versions, since as soon an update is available, it is pushed automatically to all users that have the app installed. This guarantees that all participants use the latest version of the app.

I progressively developed the app by adding more and more features over the months, according to the schedule in Table 6.1.

[1] https://play.google.com/store/apps/details?id=dk.dtu.imm.sensiblejournal&hl=en

October 2012	Movement view
November 2012	Stats view
January 2013	Spiral Timeline view
February 2013	Social Network view

Table 6.1: Dates for the release of different features of SensibleJournal

6.2 Data collection

The SensibleDTU data is collected using the *Funf Open Sensing Framework*[2]. The app runs as a service in the background, and periodically acquires data from the phone sensors. The data is periodically uploaded to the central SensibleDTU server. Phone sensors include Bluetooth, WiFi, GPS, battery, phone calls and SMS.

The sensor data is an approximation of the information about reality. GPS coordinates estimate physical location and Bluetooth scans estimate proximity to other people. As every approximation, it is subject to errors. Several categories of problems can affect the collected data:

- hardware limitations: the GPS position is always approximate to some meters, and Bluetooth scans may miss some devices
- software errors: the SensibleDTU collector may contain bugs that prevented the collection of some samples, or may have processed incorrectly some values
- human errors: people may forget their phone at home, lose it, the battery could die, the Bluetooth and GPS sensor could be turned off by accident or on purpose, the app could be uninstalled

Moreover, the usage collector stored data about the interaction with SensibleJournal.

6.3 Participants survey

In February 2013 I asked participants to answer an online survey regarding their experience with SensibleJournal. The survey contained the following questions:

[2] https://code.google.com/p/funf-open-sensing-framework/

1. how often do you use SensibleJournal?
2. which visualization did you find most useful/interesting?
3. did you discover something new or interesting about yourself?
4. if you do not use SensibleJournal, why not?
5. do you have suggestions on how we could improve SensibleJournal?
6. what kind of information or visualization would you like to have in SensibleJournal?
7. does SensibleJournal motivate you to participate in SensibleDTU?
8. would you like to use SensibleJournal, even if you were not part of SensibleDTU?

Chapter 7

Results and Discussion

Evaluating the effectiveness of visualization is challenging because the interpretation is a subjective process, and different people may gain different information from the same data or visualization [Pla04]. This chapter provides an evaluation of the proposed visualizations in three ways:

1. an analysis of the provided insights
2. a qualitative evaluation using a participants survey
3. a quantitative evaluation based on usage data for the app

7.1 Insights evaluation

In this section, I provide an evaluation of the different visualization by highlighting which insights the user can gain, as suggested in [Nor06]. For privacy reasons, I replaced all personal names and addresses with placeholder values.

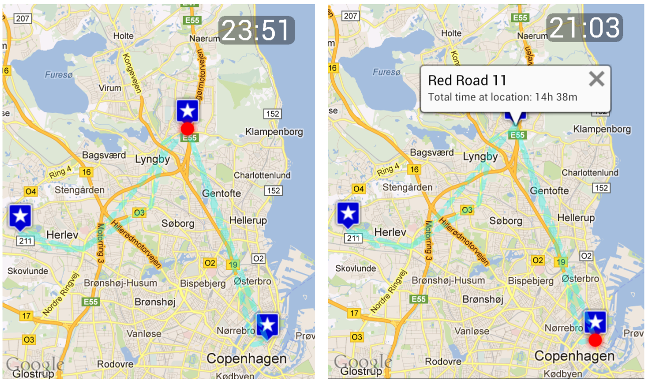


Figure 7.1: Details from the Movement view. The view shows a static overlay of the complete path for the day, composed by the trajectories between three POI marked with star symbols. Tapping on a POI reveals the time spent at that location

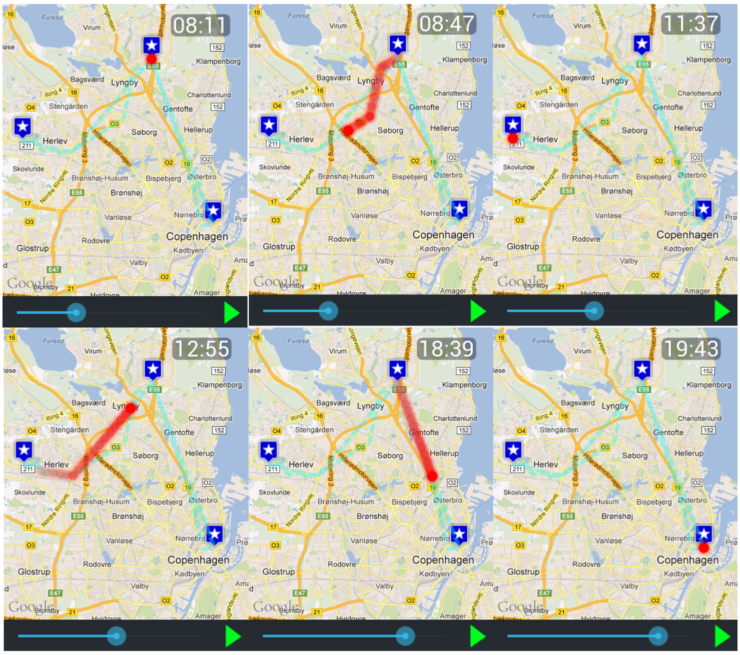


Figure 7.2: Six frames from the animation, from left to right, from top to bottom. The animation provides information on how the location changes during the day, and about the speed of movement.

7.1.1 Movement view

Travel path and POI. Figure 7.1 shows the static overlay of the complete path for the day, composed by the trajectories between three POI marked with star symbols. Tapping on a POI reveals the time spent at that location (Figure 7.1, right panel).

Location change over time. Figure 7.2 shows six frames from the animation, placed side by side from left to right, from top to bottom. Playing the animation provides information on how the location changes during the day. The frames sequence highlights how the user is stationary before 8:00, then travels from 8:00 to 9:00, then is stationary again until 12:00, and then travels between 12:00 and 13:00, and between 18:00 and 19:00. Figure 7.2 also visualizes the speed of movement. A longer trail for the dot representing the user indicates a higher speed.

7.1.2 Spiral Timeline view

Time spent at POI. The predominance of one or more colors indicates the most common POI, that is the ones where the most time is spent. In fact, distinct POI have distinct colors assigned, and more frequent locations are more

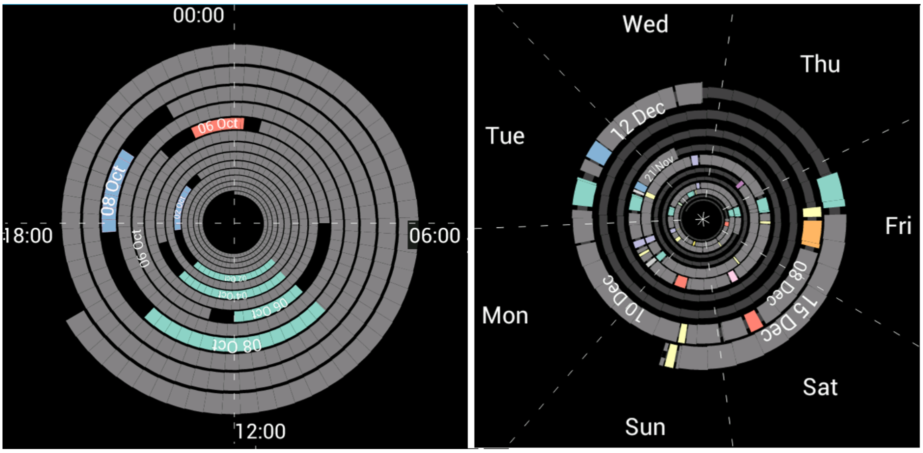


Figure 7.3: Two details from the Spiral Timeline visualization for periods of 24 hours (left) and 7 days (right). The visualization provides insights about periodic and sequential patterns, anomalies in the occurrence of events, and time spent at POI.

present on the spiral. In the left panel of Figure 7.3 it is possible to see that the gray color POI (corresponding to the location home) dominates the visualization since it is where most time is spent. The figure also shows that the light-green arcs represent the most common POI.

Periodic patterns. Arcs on the spiral that are aligned on the same sector and have the same color indicate a recurring pattern. Depending on the period, they can be patterns in the time of the day, or day of the week. In the left panel of Figure 7.3 it is easy to spot two recurring patterns: one from 9:00 to 15:00 (the light-green arcs) and one from 18:00 to 20:00 (the light-blue arcs). In the right panel of Figure 7.3 it is possible to spot a pattern for Tuesdays and Fridays (the light-green arcs).

Sequential patterns. Sequences of arcs that are repeated several times indicate a pattern of POI which are typically visited in sequence. For example in the right panel of Figure 7.3, the light-green arc closely precedes a light-blue arc, suggesting that the two corresponding POI are typically visited in sequence.

Uncommon stop locations. Arcs which have different colors and alignment than others indicate uncommon or infrequent stop locations. Such locations can be easily spotted, such as the red arc in the left panel of Figure 7.3.

Anomalies in patterns. Arcs which have the same color and are partially aligned which each other indicate an anomaly in a recurring pattern. The left panel of Figure 7.3 shows four light-green arcs in the bottom section. One of the arcs is only half the length of the other two, and that indicates how that stop location has a duration of about half of the others. In other words, the user spent at that location much less time than usual on that day.

7.1.3 Social Network view

Social interaction overview. Bubbles sizes give an indication of the most seen users, and bubble colors and clusters represent communities. From the left panel of Figure 7.4 it is easy to see who are the top contacts, and identify the two distinct communities.

Detailed community information. Zooming into a community gives detailed information about its members. The central panel of Figure 7.4 shows an example.

Detailed social interaction comparison. Switching to sorted grid mode allows to compare with ease the users which had the most and the least inter-

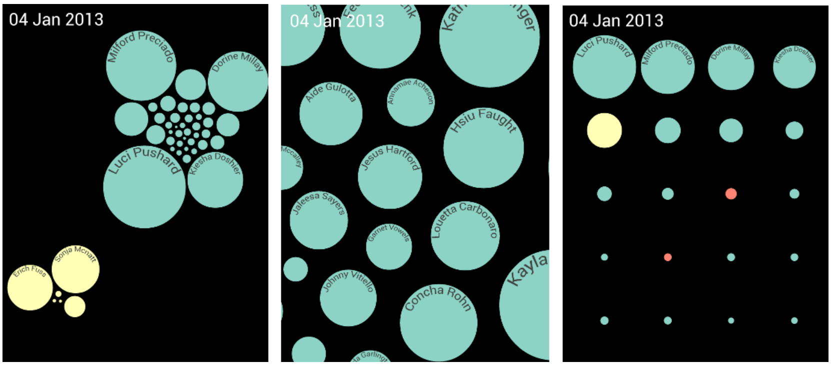


Figure 7.4: Three details from the Social Network visualization. Bubbles sizes give an indication of the most seen users, and bubble colors and clusters represent communities.

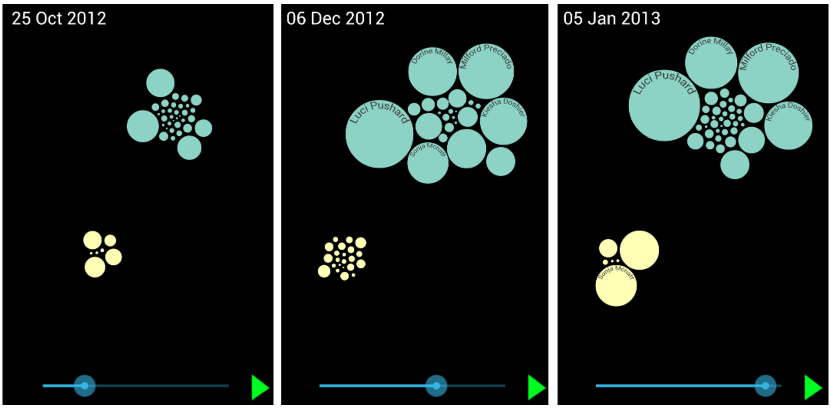


Figure 7.5: Three frames from the Social Network animation. It is possible to see how the bubbles sizes increase over time, which indicates how the social ties get stronger. It is also possible to see that bubbles change communities, indicating that the social groups change structure.

actions (Figure 7.4, right panel)

Changes over time. The animation playback gives an overview of how the social network changes over time. Figure 7.5 shows three different animation frames, side by side. It is possible to see how the bubbles sizes increase over time, which indicates how the social ties get stronger. It is also possible to see that bubbles change communities, indicating that the social groups change structure.

7.2 Participants survey results

In total 45 participants (33%) answered the online survey. 58% reported using the app less than once a week, 31% about once a week, and 11% more than once a week. 73% of the participants answered that the app motivates them to participate in the study. 73% answered that they would use the app even if they were not part of the study, and 42% reported that they had learned something new about themselves by using the app.

51% of the participants reported preference for getting insights on movement and time spent at different locations ("it is very fun to see how much of my week is spent in different places", "it is very fun to see where you work and how much or little you actually move in daily life"). 7 participants reported that they learned about their sedentary behaviour ("[I discovered] how long I stay in my own apartment", "I sit too much", "I move too little!", "I'm very still during the day"). 15% had preference for the spiral visualization, and found it useful to explore patterns in their movement behaviour.

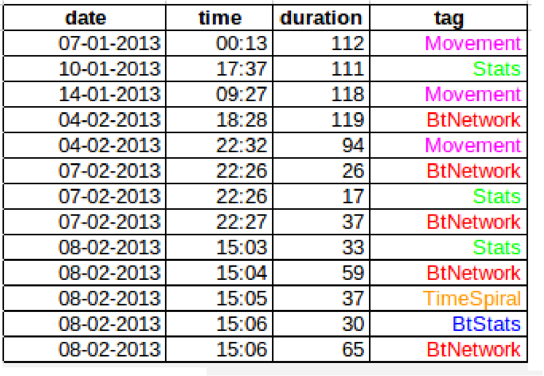
31% had preference for the social visualizations ("Social network [view] is exciting because you see how much you've been with a few people", "It's fun to see with whom I spend time", "it's fun to see [...] who you talk to").

Some participants reported more complex considerations ("you get better insights into your everyday", "my weekday movement looks very similar compared to the weekend", "It is interesting to see how little we move, but how many you talk to"). One participant was surprised to see how much time (s)he was in proximity of other people that were not the closest friends.

19 participants (42%) reported that the app was too slow and took too long to load the data. Many stated that they did not use the app very much due to this problem.

7.3 Usage analysis

The usage log allows to empirically measure how the users interacted with the app. Each log record contains a `userid`, a `timestamp` and a `tag` for the part of



date	time	duration	tag
07-01-2013	00:13	112	Movement
10-01-2013	17:37	111	Stats
14-01-2013	09:27	118	Movement
04-02-2013	18:28	119	BtNetwork
04-02-2013	22:32	94	Movement
07-02-2013	22:26	26	BtNetwork
07-02-2013	22:26	17	Stats
07-02-2013	22:27	37	BtNetwork
08-02-2013	15:03	33	Stats
08-02-2013	15:04	59	BtNetwork
08-02-2013	15:05	37	TimeSpiral
08-02-2013	15:06	30	BtStats
08-02-2013	15:06	65	BtNetwork

Figure 7.6: Part of a log for a single user. Each entry contains a timestamp and a tag for the part of the app that was accessed.

the app that the user accessed. Figure 7.6 provides an example of a log from a user.
If a user accessed view A at the timestamp t_A and view B at the timestamp t_B, then the time spent on A is $t_B - t_A$. First, the records are grouped by `userid` and sorted in ascending `timestamp` order. From this sorted sequence, the time spent on different parts of the app is extracted by calculating the difference between the records timestamps. Events with duration < 5 seconds are discarded, since they probably represent a launch of the app by mistake. Events with duration > 600 seconds are also discarded, since they can either mean that the user left the phone unattended, or an error in the log samples.

Grouping the events by tag, it is possible to extract the breakdown of time spent per visualization for each user (Figure 7.7).

Grouping events by week, it is possible to analyze the weekly changes in the usage of the app. Figure 7.8 shows the change in usage over time. It is possible to see for example that the Social Network visualization has rapidly gained popularity among users shortly after its release. It is also possible to see two usage peaks shortly after the release of new versions of the app (1 Jan 2013 and 4 Feb 2013).

Events can be also represented on a grid, where each row represents a different user, each column represents a different week, and cells are colored depending on the total amount of time spent. Figure 7.9 allows to spot usage patterns for different users, such as regular activity or long periods of inactivity.

The collected usage data enables many other types of analysis, for example how

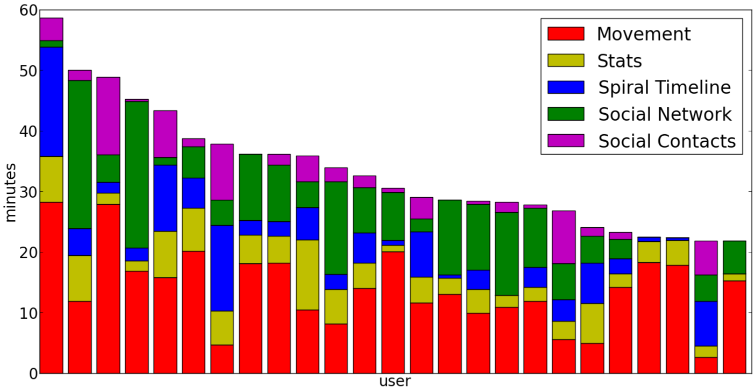


Figure 7.7: Time spent on the different visualizations by user.

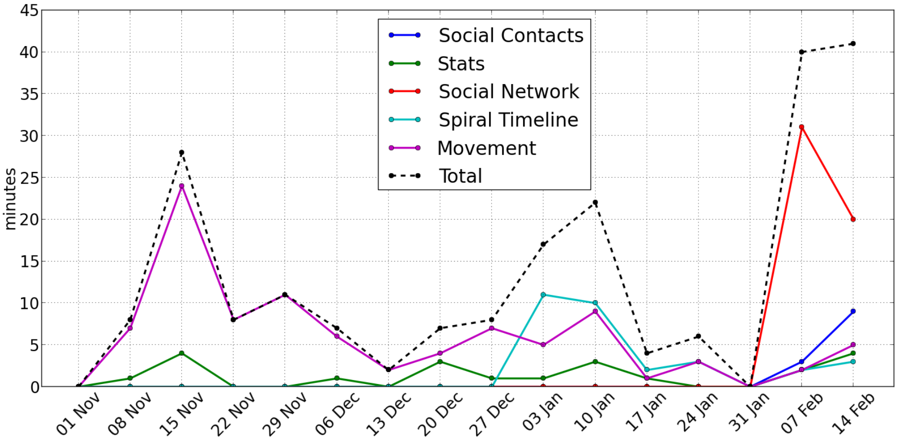


Figure 7.8: Usage of different visualizations by week.

long usage sessions last and how frequent they are, or which hours of the days or days of the week the app is used the most.

7.4 Evaluation summary

The results of the three evaluation methods suggest some interesting conclusions.

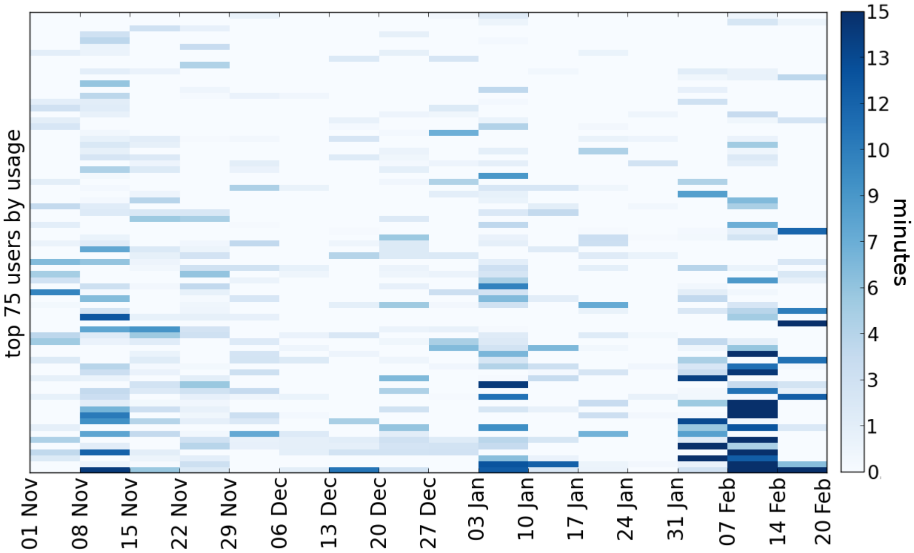


Figure 7.9: Patterns of usage by user. Each row represents a user, each column represents a week, and cells are colored depending on the total time spent.

First of all, most users are quite interested in self-tracking, and SensibleJournal motivates them in performing reflection and exploratory analysis. Users were especially interested in the social visualizations, which have quickly become the most used.

SensibleJournal allowed most users to understand more about their movement patterns and social interactions. Some users could not find any interesting information, but most users were able to gain at least some simple insights about their behaviour. Several user were able to gain some deeper understanding of periodic patterns.

Usage patterns vary a lot from user to user. Some people used the app only a couple of time during the whole period, while others use it quite frequently. The app was typically used once a week. The main reason for this was the performance problems that most users experienced. Another reason may be that the information provided by SensibleJournal does not change much from day to day, so it is more natural to get weekly updates.

Chapter 8

Conclusions

8.1 Limitations and future work

In order to better evaluate the effectiveness of the proposed visualizations, more exhaustive users feedback should be collected. It should be understood which visualizations users prefer, and which insights users can get from them. This could be done by controlled experiments involving users. Users could be given some data analysis tasks, and their interaction with the system would be recorded.

The proposed visualizations use two separate data sources (location and Bluetooth) and cover two facets of the user behaviour: movement and social interaction. These two aspects are very related since location often determines social interaction. A future development for the app is to provide information regarding the connection between these two aspects. The spiral visualization could show the Bluetooth meetings as another layer of data. The Bluetooth visualization could show in which locations the social connections happen. There are indeed many different ways to mine and visualize location and Bluetooth data. Participants feedback would also guide the development of visualizations depending on their needs.

As personal informatics system, SensibleJournal provides several complex and detailed visualizations to help the user reflect on his activity. However, the app does not give any particular suggestion regarding the actions that the user could take about his/her behavior. Future work could include an action recommendation system integrated with the visualization interfaces.

There is a significant challenge in understanding the ever-growing amount of digital information[1,2]. The conjunct usage of data mining and visualization techniques can be used to extract invaluable insights from such large amount of data, from financial trends to business intelligence [K+02]. The proposed visualizations could be generalized in order to be applied to several other fields:

- visualization of movement has application in personal movement tracking, animal migration, warehouse management
- visualization of periodic patterns has promising applications in business intelligence, event analysis and climate studies
- network visualization has application in social network analysis, network communication and graph analysis

8.2 Contributions

In this thesis I described the implementation of SensibleJournal, a personal informatics system for smartphones that provides feedback on personal mobility and social interactions. I developed a working prototype of the app and deployed it to N=136 users.

I described data mining techniques for extracting semantic information from mobile sensor data: how to extract movement mode and POI from location data, and how to infer social interactions and communities from Bluetooth data.

I proposed several novel visualizations for quantified-self data, including: a spiral view for analyzing periodic movement patterns, and a bubble view to for analyzing the temporal evolution of social networks.

A qualitative evaluation of the system shows that the visualizations provide several insights about the user behaviour, including frequently visited locations, periodical movement patterns, and social interaction changes over time. A survey among 33% of the participants has given initial indications that SensibleJournal

[1] http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/execsum.htm#summary%5D

[2] http://www.emc.com/collateral/analyst-reports/idc-digital-universe-201005.pdf

helped users to quantify several aspects of their behaviour, and that there is an interest in such personal informatics systems.

As part of the work on this thesis I have written two research papers together with my supervisors. One paper has been accepted for the *CHI 2013* workshop and another paper is currently under review for the *Mobile HCI 2013* conference. The two papers are included in Appendix A and B.

Appendix A

A Mobile Personal Informatics System with Interactive Visualizations of Mobility and Social Interactions

Currently under review for the *Mobile HCI 2013* conference.

Authors: Cuttone, A., Lehmann, S. and Larsen, J.,

Abstract: In this paper we describe a personal informatics system for Android smartphones that provides feedback on personal mobility and social interactions through interactive visualization interfaces. The mobile app has been made available to N=136 first year university students as part of a study of social network interactions in a university campus setting. We describe the design of the interactive visualization interfaces enabling the participants to gain insights into own behaviours. We report initial findings based on device logging of participant interactions with the interactive visualization app on the smartphone and from a survey on usage with response from 45 (33%) of the participants.

Appendix B

QS Spiral: Visualizing Periodic Quantified Self Data

Accepted for the *CHI 2013* workshop.

Authors: Larsen, J., Cuttone, A. and Lehmann, S.

Abstract: In this paper we propose an interactive visualization technique *QS Spiral* that aims to capture the periodic properties of quantified self data and let the user explore those recurring patterns. The approach is based on time-series data visualized as a spiral structure. The interactivity includes the possibility of varying the time span and the time frame shown, allowing for different levels of detail and the discoverability of repetitive patterns in the data on multiple scales. We illustrate the capabilities of the visualization technique using two quantified self data sets involving self-tracking of geolocation and physical activity respectively.

QS Spiral: Visualizing Periodic Quantified Self Data

Jakob Eg Larsen
Tech. University of Denmark
Dept. of Applied Mathematics
and Computer Science
Matematiktorvet, Bld 303B
2800 Kgs. Lyngby, Denmark
jel@imm.dtu.dk

Andrea Cuttone
Tech. University of Denmark
Dept. of Applied Mathematics
and Computer Science
Matematiktorvet, Bld 303B
2800 Kgs. Lyngby, Denmark
andreacuttone@gmail.com

Sune Lehmann
Tech. University of Denmark
Dept. of Applied Mathematics
and Computer Science
Matematiktorvet, Bld 303B
2800 Kgs. Lyngby, Denmark
slj@imm.dtu.dk

CHI'13, Apr 27–May 2, 2013, Paris, France.

Abstract

In this paper we propose an interactive visualization technique *QS Spiral* that aims to capture the periodic properties of quantified self data and let the user explore those recurring patterns. The approach is based on time-series data visualized as a spiral structure. The interactivity includes the possibility of varying the time span and the time frame shown, allowing for different levels of detail and the discoverability of repetitive patterns in the data on multiple scales. We illustrate the capabilities of the visualization technique using two quantified self data sets involving self-tracking of geolocation and physical activity respectively.

Author Keywords

Quantified Self, Personal Informatics, Time-Series Data, Data Visualization, Periodic Events

ACM Classification Keywords

H.5.m [Information interfaces and presentation (e.g., HCI)]: Miscellaneous.

Introduction

Recently self-tracking has gained increased attention and uptake with the availability of smartphones and low cost

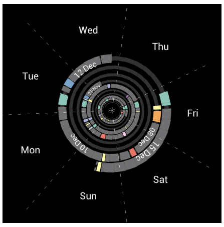


Figure 1: The QS Spiral interactive visualization with color coded data points shown on a time spiral in a 7-day view.

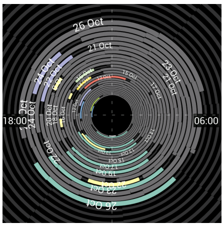


Figure 2: In this configuration the time scale of the QS Spiral is shorter and more details are shown in a 24-hour view.

wearable devices, such as, FitBit[1], Basis[2], and Zeo[3]. These devices are gradually lowering the barrier for people to engage in self-tracking and lifelogging. As suggested by Li et al.[4], self-reflection is an essential part of personal informatics. Self-reflection can be facilitated through data visualization enabling the user to interact with the collected data to receive feedback on self-tracking and to be guided in understanding of behavioral patterns. Different approaches to visualization of quantified self data has been proposed, ranging from traditional statistical charts to abstract art[4].

An inherent property of lifelog self-tracking data is the periodic properties of the data with patterns on different temporal scales including days, weeks, months, and years. Moreover self-tracking data may potentially be characterized as multi-dimensional depending on the number of signals being tracked. Our motivation is to create an interactive visualization that captures these specific properties of quantified self data. Thus in this paper we propose the data visualization approach *QS Spiral* that aims to capture both the continuity and periodic properties of quantified self data by introducing visualization of time-series data on a spiral, enabling the user to interact with and interpret the data through simple touch-based gestures.

Related Work

In visualization of lifelogging and quantified self time series data traditional charts have been the norm, but several other approaches have been proposed to enable easier and faster perception and exploration. For instance physical activity indicated by the growth of a flower on the FitBit wearable device, but also abstract visualizations or avatars have been proposed. While visualization of time-series data in a spiral structure has been proposed in [3], [5], and [6] visualization of quantified self data is novel. Although Spark[2] is an example where abstract representations of physical activity data include configurations with a spiral structure and concentric circles of data points, this visualization is fairly abstract and aimed as ambient art, and does not communicate details nor periodic events.

QS Spiral

Our proposed visualization technique is shown in Figure 1 and 2 and builds on top of a clock-dial metaphor with a circle corresponding to a time span (hour, day, week, or year). The continuity of the time series is captured by the continuous spiral timeline, with the present in the outer ring, and the past towards the center. In the present configurations one full circle represents 24 hours or 7 days as shown, and the spiral is logarithmic with the distance between each turn decreasing in geometric progression. This way the outer arcs are thicker, and become progressively thinner as they approach the center, so that most space is allocated for information in the outer ring, representing the most recent time frame. The start- and end time of each event are converted to angles and the corresponding arc is drawn on the spiral. Data points can be shown using different color coding of the timeline as well as symbols or even animations. Depending on the amount of information that needs to be embedded per time frame, the thickness of the timeline can be varied by zooming in or out. Thin lines optimizes the visualization for overview and discovery of patterns on longer time frames whereas thick lines optimizes the visualization for details on a shorter time frame. For navigational support an overlay with time of the day or day of the week is

[1] http://www.fitbit.com
[2] http://www.mybasis.com
[3] http://www.myzeo.com
[4] The work of Laurie Frick is a good example of art based on self-tracking data sets http://www.lauriefrick.com/

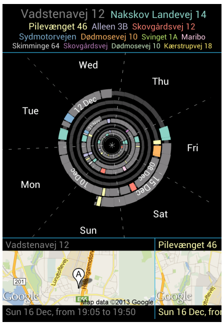

Vadstenavej 12 Nakskov Landevej 14
Pilevænget 46 Alleen 3B Skovgårdsvej 12
Sydmotorvejen Dødmosevej 10 Svinget 1A Maribo
Skimminge 64 Skovgårdsvej Dødmosevej 10 Kærstrupvej 18
Wed
Thu
Tue
Fri
Mon
Sat
Sun
12 Dec
21 Nov
10 Dec
08 Dec
15 Dec
Vadstenavej 12
Pilevænget 46
201
Lundtoftevej
Helsingørmotorv
A
E55
Google
Map data ©2013 Google
Sun 16 Dec, from 19:05 to 19:50
Sun 16 Dec, fro

Figure 3: A week view showing the geolocation data captured over a 4 month duration. A tag cloud of color-coded geolocations are shown at the top and a timeline of map segments at the bottom. Tapping the map segments will highlight those geolocations in the spiral.

added on top of the spiral.

Prototype

We have built a prototype of the *QS Spiral* visualization for Android smartphones enabling us to experiment with touch-based interfaces as the basis for the interactive visualization. The single tap gesture allows the user to highlight specific data elements. A pinch gesture allow the user to zoom into a particular part of the data and swipe gestures allow to pan, resulting in a zoomable map metaphor. The double-tap gesture switches between different time views. The combination of these gestures allow to focus into the details of a specific period, or provide overview of a larger time frame.

Case Studies

To demonstrate the capabilities of the interactive visualization we have experimented with different types of quantified self data sets. Two data sets are used to illustrate the capabilities of visualizing and interacting with quantified self data, and in particular to discover periodic trends inherent in the data.

Geolocation Self-Tracking Data

In the first example we use geolocation data captured with sensing software on an Android smartphone[1]. The software continuously acquire and log data from the embedded location sensors (GPS and WiFi). From the location data, static locations are extracted according to movement speed and grouped into logical groups using the DBSCAN clustering algorithm. The data was captured over a duration of 4 months. A screenshot from the interactive visualization is shown in Figure 3. In this interactive visualization the display is separated in three parts with the *QS Spiral* in the center. On the top is a tag cloud showing the most frequently visited places in colors corresponding to the colors shown in the spiral. The bottom of the display contains time series of places with map segments showing the places visited. When the user taps a visited place the corresponding places will be highlighted in the spiral.

Physical Activity Self-Tracking Data

The second example data set contains physical activity data captured using the FitBit step counter for a duration of 12 months. A screenshot from the interactive visualization is shown in Figure 4. In this interactive visualization the display is separated in two parts with the spiral in the center showing the physical activity level measured in number of steps color coded with color intensity corresponding to the number of steps taken in that time frame, as indicated by the legend shown at the top.

Discussion

The interactive visualization facilitates exploratory data analysis and helps identify periodic patterns. In both cases we found that that periodic events and the daily/weekly rhythms were discoverable using the *QS Spiral* visualization, but also that the repetitive nature of the data makes deviations stand out clearly. The spiral visualization represents events with similar period aligned along similar angles of the arcs, resulting in aligned sections of the spiral. Moreover, the color coding of events highlights the relation between groups of events. The combination of angle and color allows one to explore patterns of events "at a glance". As examples using the interactive visualization of geolocation data we discovered two distinct periodic patterns during (9am to 3pm and 7pm to 10pm). In addition patterns on Mondays and Wednesdays, and a pattern on Tuesdays and Fridays were found. Events with the same color belonging to the same

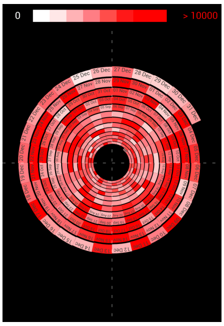


Figure 4: A four week view showing physical activity data captured over a 12 month duration, with the color density corresponding to the number of steps taken daily.

region but with different duration signal an irregularity in duration. Events not in the same region as others with the same color signal an irregularity in a periodic occurrence, and isolated events are easily discovered too. Also the choice of the scale of period changes the perception of periodicity, where a period of 24 hours highlights patterns for events happening in the same time of the day, while a period of 7 days highlights events happening in the same day of the week. In the interactive visualization of physical activity repetitive patterns were found indicating lower activity levels during weekends compared to week days. Also higher activity levels were discovered in the last quarter of the year.

In the present data sets, the 00:00am–06:00am hour time period is almost empty meaning that a substantial part of the visualization area is unutilized. Integration of multiple data sets including geolocation, physical activity, sleep, calendar data and other personal information data sources with temporal properties could utilize the full space and allow the user to discover correlations between different quantified self data streams. To make quantified self data useful a large volume of data needs to be easily accessible, so that rhythms and progress can be monitored. An interactive visualization is essential to engage users by providing feedback at a glance and the *QS Spiral* is considered a compact representation that fits on a small accessible display such as a smartphone display or even a wearable/wrist-worn (self-tracking) device. Future work include experiments with additional self-tracking data sets (with different resolution) and variations of interface configurations (such as distortions and animations) and to further explore the design language to map touch gestures to navigation of the interactive visualization.

Conclusions

We have presented *QS Spiral* as an approach to visualize and explore quantified self data with the aim for users to discover patterns and the periodic properties of the data. The visualization has been demonstrated with geolocation and physical activity self-tracking data sets.

Acknowledgments

This work funded in part by the SensibleDTU project (The Villum Foundation).

References

[1] Aharony, N., Pan, W., Ip, C., Khayal, I., and Pentland, A. Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing* (2011).

[2] Fan, C., Forlizzi, J., and Dey, A. A spark of activity: Exploring informative art as visualization for physical activity. In *Proc of the CHI2012 Workshop on Personal Informatics* (2012).

[3] Hewagamage, K., Hirakawa, M., and Ichikawa, T. Interactive visualization of spatiotemporal patterns using spirals on a geographical map. In *Visual Languages*, IEEE (1999), 296–303.

[4] Li, I., Dey, A., and Forlizzi, J. A stage-based model of personal informatics systems. In *Proc of the 28th int conf on Human factors in computing systems*, ACM (2010), 557–566.

[5] Suntinger, M., Schiefer, J., Obweger, H., and Groller, M. The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. In *Visualization Symposium*, IEEE (2008), 111–118.

[6] Weber, M., Alexa, M., and Müller, W. Visualizing time-series on spirals. In *Proceedings of the IEEE Symposium on Information Visualization* (2001), 7.

Bibliography

[AAG03] Natalia Andrienko, Gennady Andrienko, and Peter Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages & Computing*, 14(6):503–541, 2003.

[ABK+07] Luis Otavio Alvares, Vania Bogorny, Bart Kuijpers, Jose MACEDO, Bart Moelans, and Alejandro Vaisman. A model for enriching trajectories with semantic geographical information. 2007.

[ABL10] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. Link communities reveal multiscale complexity in networks. *Nature*, 466(7307):761–764, 2010.

[APF+06] Balázs Adamcsek, Gergely Palla, Illés J Farkas, Imre Derényi, and Tamás Vicsek. Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.

[API+11] N. Aharony, W. Pan, C. Ip, I. Khayal, and A. Pentland. Social fmri: Investigating and shaping social mechanisms in the real world. *Pervasive and Mobile Computing*, 2011.

[ASC+09] Harith Alani, Martin Szomszor, Ciro Cattuto, Wouter Van den Broeck, Gianluca Correndo, and Alain Barrat. Live social semantics. *The Semantic Web-ISWC 2009*, pages 698–714, 2009.

[Beh97] John T Behrens. Principles and procedures of exploratory data analysis. *Psychological Methods; Psychological Methods*, 2(2):131, 1997.

[BGLL08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[BMP^{+}12] Daniel Boston, Steve Mardenfeld, Juan Susan Pan, Quentin Jones, Adriana Iamnitchi, and Cristian Borcea. Leveraging bluetooth co-location traces in group discovery algorithms. *Pervasive and Mobile Computing*, 2012.

[BT07] David Borland and Russell M Taylor. Rainbow color map (still) considered harmful. *Computer Graphics and Applications, IEEE*, 27(2):14–17, 2007.

[BZA12] Andrey Boytsov, Arkady Zaslavsky, and Zahraa Abdallah. Where have you been? using location clustering and context awareness to understand places of interest. *Internet of Things, Smart Spaces, and Next Generation Networking*, pages 51–62, 2012.

[CCJ10] Xin Cao, Gao Cong, and Christian S Jensen. Mining significant semantic locations from gps data. *Proceedings of the VLDB Endowment*, 3(1-2):1009–1020, 2010.

[DH02] Pierre Dragicevic and Stéphane Huot. Spiraclock: a continuous and non-intrusive display for upcoming events. In *CHI'02 extended abstracts on Human factors in computing systems*, pages 604–605. ACM, 2002.

[DP73] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[EKSX96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data mining*, volume 1996, pages 226–231. AAAI Press, 1996.

[EP06] Nathan Eagle and Alex Pentland. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, 2006.

[FFD12] Chloe Fan, Jodi Forlizzi, and Anind Dey. A spark of activity: Exploring informative art as visualization for physical activity. 2012.

[Fre04] Linton C Freeman. *The development of social network analysis: A study in the sociology of science*, volume 1. Empirical Press Vancouver, 2004.

[Fry07] Ben Fry. *Visualizing Data: Exploring and Explaining Data with the Processing Environment.* O'Reilly Media, Incorporated, 2007.

[GFC05] Mohammad Ghoniem, Jean-Daniel Fekete, and Philippe Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment and statistical analysis. *Information Visualization*, 4(2):114–135, 2005.

[HB03] Mark Harrower and Cynthia Brewer. Colorbrewer. org: an online tool for selecting colour schemes for maps. *Cartographic Journal, The*, 40(1):27–37, 2003.

[HB05] Jeffrey Heer and Danah Boyd. Vizster: Visualizing online social networks. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 32–39. IEEE, 2005.

[HFM07] Nathalie Henry, J-D Fekete, and Michael J McGuffin. Nodetrix: a hybrid visualization of social networks. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1302–1309, 2007.

[HHI99] K.P. Hewagamage, M. Hirakawa, and T. Ichikawa. Interactive visualization of spatiotemporal patterns using spirals on a geographical map. In *Visual Languages*, pages 296–303. IEEE, 1999.

[HSL] Theus Hossmann, Thrasyvoulos Spyropoulos, and Franck Legendre. From wireless contacts to community structures.

[IdKM$^+$06] Wijnand IJsselsteijn, Yvonne de Kort, Cees Midden, Berry Eggen, and Elise van den Hoven. Persuasive technology for human well-being: setting the scene. *Persuasive technology*, pages 1–5, 2006.

[ISB$^+$11] Lorenzo Isella, Juliette Stehlé, Alain Barrat, Ciro Cattuto, Jean-François Pinton, and Wouter Van den Broeck. What's in a crowd? analysis of face-to-face behavioral networks. *Journal of theoretical biology*, 271(1):166–180, 2011.

[JGO06] Jungwook Jun, Randall Guensler, and Jennifer H Ogle. Smoothing methods to minimize impact of global positioning system random error on travel distance, speed, and acceleration profile estimates. *Transportation Research Record: Journal of the Transportation Research Board*, 1972(-1):141–150, 2006.

[JLJ$^+$10] Bjørn Sand Jensen, Jakob Eg Larsen, Kristian Jensen, Jan Larsen, and Lars Kai Hansen. Estimating human predictability from mobile sensor data. In *Machine Learning for Signal Processing (MLSP), 2010 IEEE International Workshop on*, pages 196–201. IEEE, 2010.

[K$^+$02] Daniel A Keim et al. Information visualization and visual data mining. *IEEE transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[LDF10] Ian Li, Anind Dey, and Jodi Forlizzi. A stage-based model of personal informatics systems. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 557–566. ACM, 2010.

[LDF11] Ian Li, Anind K Dey, and Jodi Forlizzi. Understanding my data, myself: supporting self-reflection with ubicomp technologies. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 405–414. ACM, 2011.

[LF09] Andrea Lancichinetti and Santo Fortunato. Community detection algorithms: a comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[LML$^+$06] James Lin, Lena Mamykina, Silvia Lindtner, Gregory Delajoux, and Henry Strub. Fish'n'steps: Encouraging physical activity with an interactive computer game. *UbiComp 2006: Ubiquitous Computing*, pages 261–278, 2006.

[LN08] Elizabeth A Leicht and Mark EJ Newman. Community structure in directed networks. *Physical Review Letters*, 100(11):118703, 2008.

[LPDR06] Jamie Lawrence, Terry R Payne, and David De Roure. Co-presence communities: Using pervasive computing to support weak social networks. In *Enabling Technologies: Infrastructure for Collaborative Enterprises, 2006. WETICE'06. 15th IEEE International Workshops on*, pages 149–156. IEEE, 2006.

[MCLP10] Anmol Madan, Manuel Cebrian, David Lazer, and Alex Pentland. Social sensing for epidemiological behavior change. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 291–300. ACM, 2010.

[MFGPP11] Anmol Madan, Katayoun Farrahi, Daniel Gatica-Perez, and Alex Pentland. Pervasive sensing to model political opinions in face-to-face networks. *Pervasive Computing*, pages 214–231, 2011.

[New04] Mark EJ Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(6):066133, 2004.

[Nor06] Chris North. Toward measuring visualization insight. *Computer Graphics and Applications, IEEE*, 26(3):6–9, 2006.

[OGB+02] Jennifer Ogle, Randall Guensler, William Bachman, Maxim Koutsak, and Jean Wolf. Accuracy of global positioning system for determining driver performance parameters. *Transportation Research Record: Journal of the Transportation Research Board*, 1818(-1):12–24, 2002.

[PBKA08] Andrey Tietbohl Palma, Vania Bogorny, Bart Kuijpers, and Luis Otavio Alvares. A clustering-based approach for discovering interesting places in trajectories. In *Proceedings of the 2008 ACM symposium on Applied computing*, pages 863–868. ACM, 2008.

[Pla04] Catherine Plaisant. The challenge of information visualization evaluation. In *Proceedings of the working conference on Advanced visual interfaces*, pages 109–116. ACM, 2004.

[PS06] Adam Perer and Ben Shneiderman. Balancing systematic and flexible exploration of social networks. *Visualization and Computer Graphics, IEEE Transactions on*, 12(5):693–700, 2006.

[PSM07] Zachary Pousman, John T Stasko, and Michael Mateas. Casual information visualization: Depictions of data in everyday life. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1145–1152, 2007.

[PTH+06] Jukka Perkio, Ville Tuulos, Marion Hermersdorf, Heli Nyholm, Jukka Salminen, and Henry Tirri. Utilizing rich bluetooth environments for identity prediction and exploring social networks as techniques for ubiquitous computing. In *Web Intelligence, 2006. WI 2006. IEEE/WIC/ACM International Conference on*, pages 137–144. IEEE, 2006.

[PXYH05] Doantam Phan, Ling Xiao, Ron Yeh, and Pat Hanrahan. Flow map layout. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 219–224. IEEE, 2005.

[Rae09] Alasdair Rae. From spatial interaction data to spatial interaction information? geovisualisation and spatial structures of migration from the 2001 uk census. *Computers, Environment and Urban Systems*, 33(3):161–178, 2009.

[RFF^{+}08] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *Visualization and Computer Graphics, IEEE Transactions on*, 14(6):1325–1332, 2008.

[Rog06] Yvonne Rogers. Moving on from weiser's vision of calm computing: Engaging ubicomp experiences. *UbiComp 2006: Ubiquitous Computing*, pages 404–421, 2006.

[RSH00] Johan Redstrom, Tobias Skog, and Lars Hallnas. Informative art: using amplified artworks as information displays. In *Designing Augmented Reality Environments: Proceedings of DARE 2000 on Designing augmented reality environments: Elsinore, Denmark*, volume 2000, pages 103–114, 2000.

[SA09] Nadine Schuessler and Kay W Axhausen. Processing raw data from global positioning systems without additional information. *Transportation Research Record: Journal of the Transportation Research Board*, 2105(-1):28–36, 2009.

[Shn96] Ben Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pages 336–343. IEEE, 1996.

[Shn01] Ben Shneiderman. Inventing discovery tools: Combining information visualization with data mining. In *Discovery Science*, pages 17–28. Springer, 2001.

[SQBB10] Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-László Barabási. Limits of predictability in human mobility. *Science*, 327(5968):1018–1021, 2010.

[SSOG08] M. Suntinger, J. Schiefer, H. Obweger, and M.E. Groller. The event tunnel: Interactive visualization of complex event streams for business process pattern analysis. In *Visualization Symposium*, pages 111–118. IEEE, 2008.

[Tid10] Jenifer Tidwell. *Designing interfaces.* O'Reilly Media, Incorporated, 2010.

[WAM01] M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *Proceedings of the IEEE Symposium on Information Visualization*, page 7, 2001.

[WDSC07] Jo Wood, Jason Dykes, Aidan Slingsby, and Keith Clarke. Interactive visual exploration of a large spatio-temporal dataset: reflections on a geovisualization mashup. *Visualization and Computer Graphics, IEEE Transactions on*, 13(6):1176–1183, 2007.

[Yau11] Nathan Yau. *Visualize this: the flowingdata guide to design, visualization, and statistics.* Wiley, 2011.

[YCP$^+$12] Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. Semantic trajectories: Mobility data computation and annotation. 2012.

[YKSJ08] Ji Soo Yi, Youn-ah Kang, John T Stasko, and Julie A Jacko. Understanding and characterizing insights: how do people gain insights using information visualization? In *Proceedings of the 2008 conference on BEyond time and errors: novel evaLuation methods for Information Visualization*, page 4. ACM, 2008.

[ZLTX05] Kaidi Zhao, Bing Liu, Thomas M Tirpak, and Weimin Xiao. A visual data mining framework for convenient identification of useful knowledge. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[ZXM10] Yu Zheng, Xing Xie, and Wei-Ying Ma. Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data Engineering Bulletin*, 33(2):32–40, 2010.

[ZZXY10] Vincent W Zheng, Yu Zheng, Xing Xie, and Qiang Yang. Collaborative location and activity recommendations with gps history data. In *Proceedings of the 19th international conference on World wide web*, pages 1029–1038. ACM, 2010.