UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

GUILHERME DO NASCIMENTO OLIVEIRA

Ordered Stacks of Time Series for Exploratory Analysis of Large Spatio-Temporal Datasets

Thesis presented in partial fulfillment
of the requirements for the degree of
Doctor of Computer Science

Prof. Dr. João Luiz Dihl Comba
Advisor

Prof. Dr. Rafael Piccin Torchelsen
Coadvisor

Porto Alegre, November 2015

CIP – CATALOGING-IN-PUBLICATION

Oliveira, Guilherme do Nascimento

Ordered Stacks of Time Series for Exploratory Analysis of Large Spatio-Temporal Datasets / Guilherme do Nascimento Oliveira. – Porto Alegre: PPGC da UFRGS, 2015.

97 f.: il.

Thesis (Ph.D.) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2015. Advisor: João Luiz Dihl Comba; Coadvisor: Rafael Piccin Torchelsen.

1. Time Series. 2. Bike Sharing. 3. Running. 4. Spatio-Temporal Data. 5. Urban Data. 6. Visualization. 7. Exploratory Data Analysis. I. Comba, João Luiz Dihl. II. Torchelsen, Rafael Piccin. III. Título.

“Of all the intellectual hurdles which the human mind has confronted and has overcome in the last fifteen hundred years the one which seems to me to have been the most amazing in character and the most stupendous in the scope of its consequences is the one relating to the problem of motion.”

— Sir Herbert Butterfield

ABSTRACT

The size of datasets became the major problem in data analysis today. As urban sensing becomes popular, datasets of spatial and temporal nature become ubiquitous, leading to several concerns regarding storage and management. It also creates a shift of paradigm in data analysis, as datasets that once represented a single series of measurements ordered in time are now composed of hundreds of series with ever increasing sampling rates. Also, as urban data usually presents inherent geographic disposition, most analysis tasks requires the support of proper spatial views. It becomes another problem, once that displaying technologies do not advance at the same of pace that sensing technologies do, and consequently, there is usually more data than visual space to represent it. After conducting exhaustive research on temporal data analysis and visualization, we improved a compact visual representation of time series to support the exploration of large spatio-temporal datasets. Our proposal exploits the compactness of such representation to allow the use of a map to represent the spatial properties of the data in a coordinate scheme while presenting, in a comprehensible manner, hundreds of series simultaneously, with full temporal context. We argue that such solution can effectively support many exploratory tasks in an intuitive manner. To support this claim, we show how the idea was conceived, and improved along the development of two design studies from different application domains, and validated by the implementation of prototypes used in the exploratory analysis of several datasets with 3 different data structures.

Keywords: Time Series, Bike Sharing, Running, Spatio-Temporal Data, Urban Data, Visualization, Exploratory Data Analysis.

Pilhas Ordenadas de Series Temporais para a Exploração de Conjuntos de Dados Espaço-Temporais

RESUMO

O tamanho dos conjuntos de dados se tornou um grande problema atualmente. À medida que o sensoriamento urbano ganha popularidade, os conjuntos de dados de natureza espacial e temporal se tornam ubíquos, e levantam uma série de questões relacionadas ao armazenamento e gerenciamento destes. Isso também cria uma mudança no paradigma de análise, uma vez que os conjuntos de dados que antes representavam uma única série de medições ordenadas no tempo, agora são compostos por centenas dessas séries, com uma taxa de amostragem que está aumentando constantemente. Além disso, uma vez que os dados urbanos normalmente apresentam disposição geográfica inerente, a maioria das das tarefas requerem o suporte de representações espaciais apropriadas. Este se torna outro problema, visto que as tecnologias de exibição de imagens não avançam na mesma velocidade das tecnologias de sensoriamento, de modo que consequentemente acaba-se tendo mais dados do que espaço visual para representa-los. Após conduzir uma pesquisa exaustiva a respeito de análise de dados temporais e visualização, nós melhoramos uma visualização compacta de series temporais para auxiliar a exploração de grandes conjuntos de dados espaçotemporais. Nossa proposta aproveita a compacticidade de tal representação para permitir o uso de um mapa para representar os atributos espaciais dos dados, de modo coordenado, enquanto representação, de forma compreensível, centenas de series simultaneamente, com total contexto temporal. Nós apresentamos nossa proposta como sendo capaz de auxiliar várias tarefas de caráter exploratório de forma intuitiva. Para defender essa afirmação, nós mostramos como essa ideia foi desenvolvida e melhorada ao longo do desenvolvimento de dois estudos de design visual em diferentes domínios de aplicação, e validamos com a implementação de protótipos que foram usados na análise exploratória de vários conjuntos de dados com 3 representações diferentes.

Palavras-chave: Series Temporais, Compartilhamento de Bicicletas, Corridas de Rua, Dados Espaço-Temporais, Dados Urbanos, Visualização, Analise Exploratória de Dados.

CONTENTS

LIST OF ABBREVIATIONS AND ACRONYMS

AF	Atrial Fibrillation
BSS	Bike-Sharing System
BBSS	Balancing Bike-Sharing Systems
CPU	Central Processing Unit
DF	Dominant Frequency
EDA	Exploratory Data Analysis
ECG	Electrocardiogram
GPU	Graphics Processing Unit
HR	Heart Rate
LCHS	London Cycle Hire Scheme
MHR	Maximum Heart Rate
NYC	New York City
OD	Origin/Destination
PVI	Pulmonary veins isolation

LIST OF FIGURES

1 INTRODUCTION

Appearing constantly in every field of science and engineering as well as implicitly in many nontechnical activities of ordinary people, time series are the most ubiquitous kind of data available nowadays. We usually associate the notion of *time series* with the most basic and explicit instance of such data like temperature measurements through the day, however such data is far more available than most people realize, as it lies implicit in almost every dataset. The reason is that time itself is a variable that can be assigned as a key attribute to most data entities, thus turning a wide range of dataset into potential time series, and the study of such model of data is of great importance.

Recently, the accelerated advance of sensing technologies, cities are becoming a major source of data. These urban datasets reflect city dynamics and usually can tell a lot about how people live. The problem is that the ever increasing sampling rates at which data is conceived and the fact that it usually has many attributes associated turn the analysis almost unfeasible without heavy computation. With such massive amount of information, the trend is to learn from the dataset as whole instead of analyzing individual elements. The actual challenge in data analysis is dealing with the high amount of data, relate different elements and available variables, and discover useful information that is implicit. The Data Mining field provide a wide range of tools to analyze data when the objective is known in advance, however when this is not the case a proper information visualization technique is a better solution. Actually, Andrienko and Andrienko [7] point that the first step in the general paradigm for using computational tools (see diagram in figure 1.1) is to look at the data, so we can understand what computational tools could be useful. The purpose of data visualization is to provide a first overview of the dataset so that the analyst can notice behaviors and patterns, spot outliers, make better choices of computational models, improve those models, and estimate initial parameters.

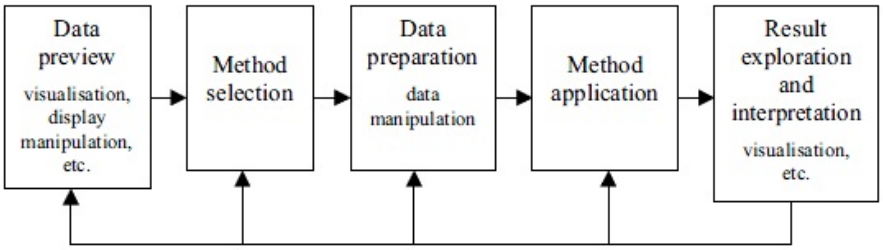


Figure 1.1: [Steps in computational paradigm] The steps in the computational paradigm. (from [7])

Now, through the perspective of data visualization, the problem is that display technologies do not keep up with sensing ones. As a result, there is much more data to analyse

than visual space available to represent it. In addition, the increasing complexity of the datasets also leads to incresing visual requirements. For instance, urban data usually has spatio-temporal properties, which requires representation in both spatial and temporal frames.

Through the duration of this thesis, we worked with the visual analysis of spatio-temporal datasets and used different techniques to support many exploratory tasks. More importantly, we adopted a known technique for the representation of a set of hundreds series and adapted it to the exploration of systems of spatio-temporal series. We present some original designs created around this idea to explore many real datasets, using three different data structures from two application domains. We argue in favor of this approach as a solution that make feasible the analysis of groups of hundreds of spatio-temporal series with usual settings of display resources, and present the results of two case studies to prove our point.

This report is divided in two main parts: I Background and II Original Work. In part I, we provide context with the exposition of three topics of major importance to our works, in three chapters: 3 Time Series, 4 Time Series Visualization and 5 Design Aspects for Exploratory Analysis. The first one (Chapter 3) is further divided in two sections about how the series can be represented and how to compare different series. Those are topics of major importance. Comparison is an ubiquitous task in data analysis, but since time series are complex high dimensional data types, their comparison is non-trivial and has a strong dependence on how they are represented as means of data structure. The second chapter then reviews aspects of time series regarding their visual representation, while the last covers aspects in the design of exploratory analysis solutions.

In part II we present the main works related to our design view of compressed time series together with the rationale supporting our choice. Then we introduce the two original works we developed as design studies in two chapters: the Visualization of Running Races and the Visualization of the Dynamics of Bike-sharing Systems. Each chapter is further divided in 4 sections: 1 Related Works, to review the state of the art on the subject; 2 Data, to present the dataset acquisition and processing stages; 3 Method, to introduce our design solution and 4 Results, presenting the outcomes of our analysis using our prototypes and real datasets as use cases. Finally, we conclude with some final considerations in part 9. Also, we added one Appendix section that presents an original work on Atrial Fibrillation, in which our role was secondary, but still related to time series analysis and visualization.

Part I

Background

2 TERMINOLOGY

Our work is related to topics that share a large list of concepts. Different works tend to use the same word for different concepts and also different words for the same concept, so, to avoid confusion due to conflicting terminology along our exposition to follow, we present now a list of terms frequently used in our context along with the meaning they convey in our rationale. This list is based in the terminology used in Adrienko's book on exploratory analysis of spatio-temporal data [8].

Data: Data are records sharing the same structure. Each record contains measurements about the results of some observation or its context. The context is usually related to independent variables (mainly time and space in this work).

Structure: The ordering of the data records, with each position having its meaning. A position is also called a *component* of the data. A *value domain* is the set of all values that can appear in a data component.

Components of Data: A data component that corresponds to the observed property of the phenomenon is a *charactetistic* component or *attribute*. Their values are called *characteristics*.

Components given context about the observation are *referential* components or *referrer*. In our work, space and time are the referrers of major importance (specially the last). The value of one or more referrers, that fully characterizes the context of a observation is a *reference*.

Dataset: A set of *data*. It characterizes a phenomenon and through the data analysis of a dataset, an analyst gains knowledge about the former. The content registered in a data record is a *value*, with the ones that reflect the results of the observation is a *characteristic* while those that tell the context are *references*.

Multidimensional Data: The dimensionality of a dataset is related to the number of data components. Usually, specially in spatio-temporal analysis, only the number of referrers is considered, attributes are not taken as dimensions of a dataset. This constraint is normally dropped in scenarios where space is not part of the referrers set.

Independent and Dependent Variables: Since the context for observations may be chosen arbitrarily, and do not depend on other aspects, referrers are independent variables. The attributes are dependent ones, as the context determines the characteristics observed.

Data Function: A function defining a correspondence between references (values of referrers) and characteristics (values of attributes).

$$f(x_1, x_2, ..., x_M) = (y_1, y_2, ..., y_N) \tag{2.1}$$

with M as the number of referrers in the dataset, N the number attributes, $x_1, x_2, ..., x_M$ the refferers (independent variables) and $y_1, y_2, ..., y_N$ the attributes.

Behavior: The resulting configuration of characteristics that arise from a data function applied to a set of references, considering the relations between those references as well. It is related to how the characteristics change in response to changes in the references. For example, the behavior over a period is the sequence of characteristics corresponding to the ordered time instants in the given period. With space as referrer, the behavior is a distribution of the characteristics over an area.

Pattern: A construct that present the essential features of a given behavior in a simpler fashion, without the specification of every reference and corresponding characteristics. It can be described in natural language, formally or visually. Examples are increasing trends of a numeric attribute over time, and concentration of the same characteristic in a small area.

Task: Tasks are questions about data to be answered considering its referrers and attributes. They are composed of two parts: the target, that is what information we want to obtain, and the constraints, that represent the conditions the information needs to conform to.

Tasks concerned about individual references and characteristics, are named *elementary*. The *elementary level of analysis* is the search of answers for elementary tasks. Opposed to elementary tasks, *synoptic* tasks consider sets of references and the corresponding behavior of the attributes. In a synoptic task, the set of references is considered a whole entity. *Synoptic level of analysis* is the primary objective in exploratory data analysis.

Comparison Task: The identification of relations between elements, that can be references, characteristics, sets of references, or behaviors. Comparison is probably the most frequent task in exploratory analysis. Some example of comparison tasks are differentiation (if two elements are equal or not), ordering and distance (when it difference can be measured). There is also comparison of sets (can include, overlap or not overlap each other), and of behaviors (can be similar, dissimilar, or even opposite).

3 TIME SERIES AS BIG DATA

Analysis of temporal data used to have a single time series as its dataset: a group of data records ordered by the time of observation. But now, sensing technologies have been creating an ever increasing amount of series with increasing sampling rates as well, forcing a paradigm shift in the analysis. Time series became the data records of the datasets under analysis. This shift has major implications, mainly because time series are essentially high-dimensional; not in the sense of dimensionality given previously in the terminology we adopted, but with respect to the number of values in a series: the number of samples, defining the series size. While the comparison of canonical data types like categorical and ordinal variables, is normally straightforward, comparing different time series is not a simple task. An analyst must choose a proper function that measures the dissimilarity between series while conveying the right semantics for the context domain. Another issue is the computational complexity of the comparison function since the dataset can contain hundreds to thousands of series, each with so many samples, resorting to a complex comparison may become a problem.

We put together in this chapter a quick overview of methods, from the domains of data mining and machine learning, which are concerned with the management of time series to support data analysis tasks like querying and classification. While our original works, presented in the second part of this document, do not apply machine learning solutions, the topics presented here gives a view into the complexity of the analysis of multiple time series and are also relevant to conception of visualization designs and tools.

We first present a review of representation methods, as the structure used may have implication in the comparison procedure, then the different dissimilarity measures are reviewed with the semantics they convey for analysis tasks.

3.1 Representation

Works on representation focus in the definition of data structures to represent time series. The proposed solutions are always concerned about reducing storage requirements and improving the effectiveness and efficiency of querying and classifying the series. Keogh and Kasetty [46] claim that "in the last decade there has been an explosion of interest in mining time series data" and "literally hundreds of papers have introduced new algorithms to index, classify, cluster and segment time series". Wang [75] reinforces this belief. Both works overview previously proposed solutions and present exhaustive comparative experimental analysis for dozens techniques with many datasets. The review we present is primarily based on their works.

Wang [75] formally defines a time series as a sequence of pairs

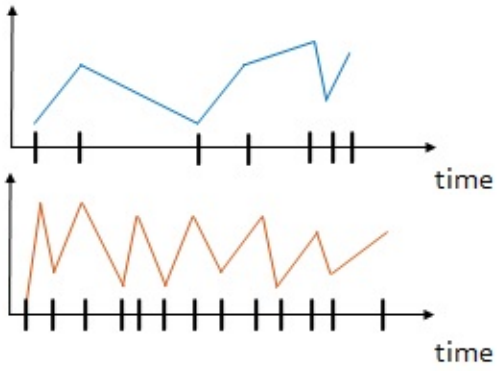


Figure 3.1: Series with different sampling rates and sizes.

$$T = [(p_1, t_1), (p_2, t_2), ..., (p_i, t_i), ..., (p_n, t_n)] \tag{3.1}$$

where p_i is a data point in a d-dimensional data space, and t_i is the time stamp at whitch p_i ocurred. If the sampling rate of the series are the same, then t_i can be omitted, simplifying the representation. Such structure is named *raw representation*. Usually, with real datasets, the sampling rates can vary between different series, it can also be variable in the same series ($t_i - t_{i-1} \neq t_{i+1} - t_i$), and their lengths n (number of samples, or size) may differ as well (see figure 3.1).

- **Data Adaptive**
  - Piecewise Polynomials
    - Interpolation*
    - Regression
  - Adaptive Piecewise Constant Approximation (APCA) *
  - Singular Value Decomposition (SVD) *
  - Symbolic
    - Natural Language
    - Strings
      - Non-Lower Bounding
      - Symbolic Aggregate approXimation (SAX) *
      - Clipped Data*
  - Trees
- **Non-Data Adaptive**
  - Wavelets*
  - Random Mappings
  - Spectral
    - Discrete Fourier Transformation (DFT) *
    - Discrete Cosine Transformation (DCT) *
    - Chebyshev Polynomials (CHEB) *
  - Piecewise Aggregate Approximation (PAA) *

Table 3.1: Representation methods (from [75])

The table 3.1 shows a classification of some major representation methods in a hierarchy. Since a representation solution is always a simplification of the raw representation,

there is usually error between both representations. The methods are divided into two classes: adaptive methods adapt locally to the content of each series attempting to minimize the reconstruction error, and non adaptive methods that handle the whole dataset equally.

3.1.1 Non Adaptive Methods

Piecewise Aggregate Approximation (PAA) divides a series into segments of equal lengths and stores for each segment the average of the values of the data points that fall within it. Other techniques represent the series as a combination of basis functions (Wavelets, Discrete Fourier Transformation, Discrete Cosine Transformation, and Chebyshev Polynomials), storing a reduced set of coefficients. The more coefficients used, the lesser the reconstruction error, however, it increases the storage requirement.

3.1.2 Adaptive Methods

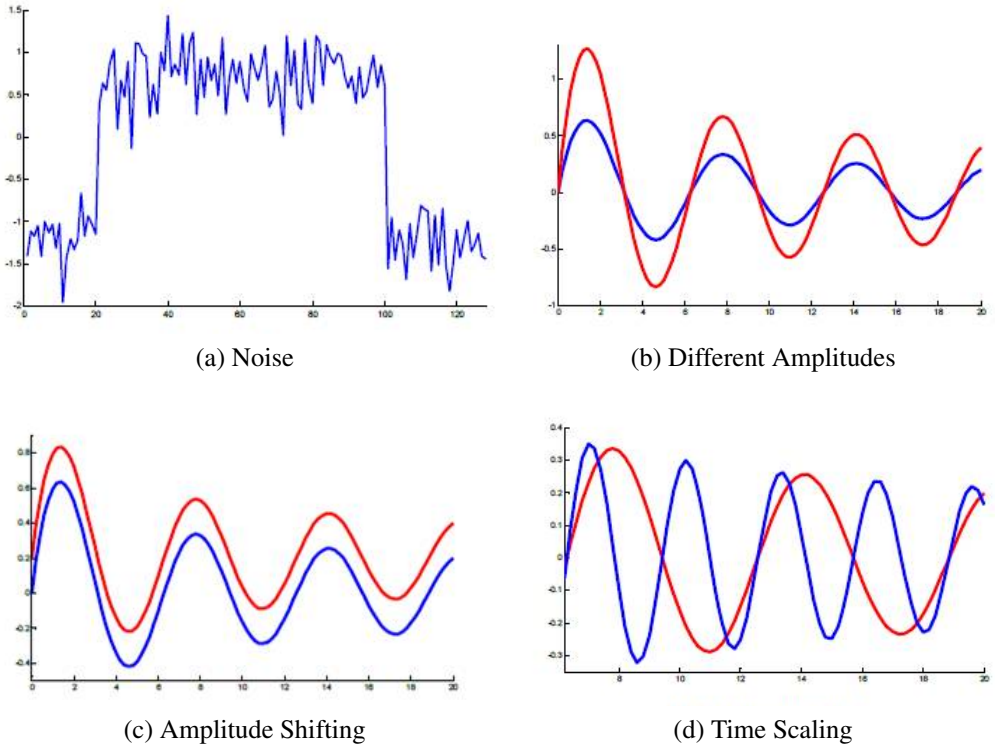
Adaptive Piecewise Constant Approximation (APCA) divides a series into segments, but differ from (PAA) as the length of the segments vary to reduce the reconstruction error. Piecewise Polynomials techniques aproximate the series by fitting polynomial curves and using the curves parameters as representation. Symbolic techniques represent each section of a time series as a symbol instead of a set of numeric values. Symbolic Aggregate Approximation, for instance, first transform the series into Piecewise Aggregation Approximation and then convert each segment to a letter.

According to [75], a characteristic that is very desirable in representation method is the one of allowing the calculation of lower bounds, that allows one to create a distance measure that when applied to compare series reduced using such representation is guaranteed to have a value lesser than or equal to the true distance (dissimilarity, difference) given if using the raw representation (no compression, reduction). The practical outcome of such property is that indexing, querying, comparing the series can be done using the reduced representations with the guarantee of no false-negatives. The methods that allow lower bounding are marked with * in the table.

3.2 Comparison

As with any data type, time series cannot be compared when represented using different structures. Comparison is the next step following representation, being the other motivation besides the reduced storage. It is a fundamental exploratory task and the base of more complex ones like querying and classification.

Figure 3.2 depicts five different issues that must be carefully addressed when comparing different series. Noise is a common issue in signal processing and analysis. Removing noise from a series makes it easier to understand its shape and properties. The problem of reducing noise is usually solved by converting the raw series using a proper representation technique. Series with similar shapes can have different amplitudes of values. In different contexts, such series can be considered similar or different. The same is valid for amplitude shifting, when they have the same amplitude of values, but centered around different averages. Both the case of different amplitudes and amplitude shifting can be ignored by applying normalization, i.e. scaling each series to the range of 0 to 1. Again, different scenarios may consider series with such discrepancies to be different or not. Time scaling is a complex problem. In some applications, like voice recognition, two series are similar if they have the same shape, even with distorted along the time axis; two people can say



(a) Noise

(b) Different Amplitudes

(c) Amplitude Shifting

(d) Time Scaling

Figure 3.2: Comparison problems

the same word, but at different speeds. It also important in the identification of patterns. For instance, in the figure, both series have the same sinusoidal behavior, but at different frequencies. In an effective pattern querying framework, if the red series were given as input, the system should return the blue one as matching the query.

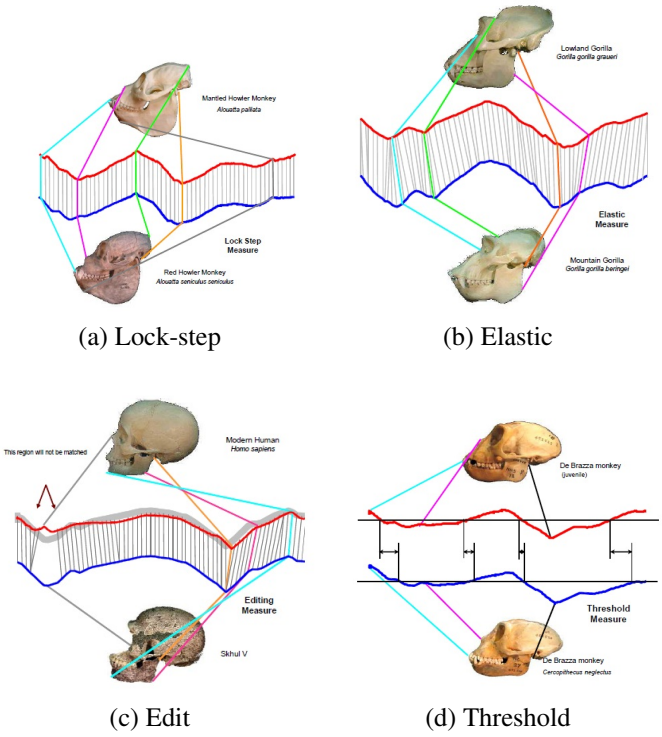
The measures are divided in two major groups: lock-step measures and elastic measures. The table 3.2 presents the hierarchy of the major dissimilarity measures. The term dissimilarity is usually used instead of similarity since a value of 0 means that the series are equal, and increases together with the difference between the subjects. Figure 3.3 exemplify the rationale behind some of the class of measures.

- **Lock-step Measure**
  - L_p-norms
    - L_1-norm (Manhattan Distance)
    - L_2-norm (Euclidean Distance)
    - L_{inf}-norm
  - DISSIM
- **Elastic Measure**
  - Dynamic Time Warping (DTW)
  - Edit distance based measure
    - Longest Common SubSequence (LCSS)
    - Edit Sequence on Real Sequence (EDR)
    - Swale
    - Edit Distance with Real Penalty (ERP)

Table 3.2: Different measures (from [75])

3.2.1 Lock-step Measures

Lock-step measures compare samples of the same index in both series one by one; thus the series need to have the same length, and samples with the same index must be coherent (correspond to the same time). The Euclidian Distance is the most used lock-step measure [75], but other forms of the L_p-norms are also used. Its set of advantages includes being intuitive to understand, easy to implement, parameter-free and having linear computational complexity. The drawback is that since the mapping between samples in the comparison is fixed, such measures are sensitive to noise, to misalignment in the time axis and to local time shifts.



(a) Lock-step (b) Elastic

(c) Edit (d) Threshold

Figure 3.3: Time series are used to represent the shape of a image by means of the variation of its curvature. Different dissimilarity measures are used to compare the shape of the figures. (from [75])

3.2.2 Elastic Measures

Elastic measures differ from lock-step ones by not being rigid (as the name implies) in the mapping between the samples of the series. These measures are concerned with the problem of global and local temporal shifting, allowing the series to "stretch" or "compress" (related to time) to achieve a better match. The dynamic time warping allows 1-to-many mapping between samples, thus not restricting the series to have equal lengths. The major drawback is its quadratic computational complexity, but "many lower bounding measures have already been devised to speed up searches using DTW and it has been shown that the amortized cost for computing DTW on large datasets is linear" [75]. Like DTW, edit distances allows 1-to-many mapping, but also opens the possibility of not matching some points.

4 TIME SERIES VISUALIZATION

We present a study of information visualization solutions regarding time-related data, ranging from general techniques proven to be applicable in several use cases to very domain-specific tools, including the most accepted taxonomies and works that represent the state of the art in the subject. We start by introducing a taxonomy regarding time and data models, and how they are represented visually.

Working with time-oriented datasets makes room for several temporal questions, some examples of common analysis tasks are:

- When was something greatest/least?
- Is there any pattern?
- Are two or more series similar?
- Do any of the series match a pattern?
- Simples and faster access to the series.
- Does data element exist at time t (or period p)?
- When does a data element exist?
- How long does a data element exist?
- How often does a data element occur?
- How fast are data elements changing?
- In what order do data elements appear?
- Do data elements exist together?

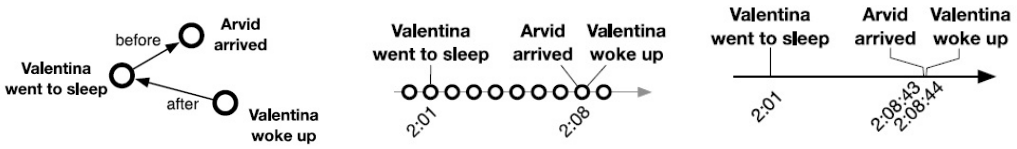
These questions, the way they will be solved, and their answers, vary according to the time model used and the type of data tied to the time elements. In this chapter we present the taxonomy suggested in [4, 5] to guide the creation of effective time-oriented data visualizations by understanding the inherent issues of each class. The classification is based on three criteria: time, data and representation. Time and data criteria are related to the dataset itself, representation, however, depends of the tasks one would want to perform and is also known as visualization aspects.

4.1 Time

The time criteria corresponds to the model used to represent time in the dataset, what are the properties of the time axis. A study on different representations of time can be found in [39] but for practical purpose the categorization used to classify time was the one in [30]. Time criteria is divided in scale, scope, arrangement, viewpoint, granularity and primitives.

4.1.1 Scale (ordinal, discrete and continuous)

Scale can be seen as the level of detail of the time axis. Ordinal scale is the level of less detail, where the only relation about two elements is relative order, i.e., we know which happens before or after but cannot determine how long is the time span between them. Such relation is a qualitative temporal relation, with no quantitative information. Discrete time present time values that can be mapped to integer values so that temporal distance between elements can now be evaluated, however there is no information of events that might have happened between two consecutive points. Continuous scale extends the discrete case by mapping time values to real numbers thus between any two point in time, a third one exists. Figure 4.1 show events occurring in the different time scales.



(a) (b) (c)

Figure 4.1: (a) Ordinal scale. (b) Discrete scale. (c) Continuous scale. Diagram from [5]

4.1.2 Scope (point and interval)

Scope defines the temporal extent of the basic time primitives. Point-based primitives are analogous to a Euclidian point in space, that has area equals to zero, having no duration, as representing a date by a single instant at that day. Interval primitives have inherent duration , i.e., representing the date as the whole time span of the day from 00:00:00 to 23:59:59.

4.1.3 Arrangement (linear and cyclic)

Arrangement relates to the shape of the time domain. The linear arrangement is the ordinary perception of time as flow infinitely from past to future in a linear fashion. The cyclic arrangement on the other hand is a set of time values that recur, this way, a given time value precedes and succeeds another one at the same time, i.e., the seasons of the year, winter comes after summer but also before as the pattern repeats itself. Figure 4.2 illustrate the different arrangements.

4.1.4 Viewpoint (ordered, branching and multi-perspective)

As the name implies, viewpoint is concerned with the different views of the time axis, how many views exist. Ordered viewpoint is the ordinary variant: the events happens one after the other in the time axis. This type of view can be totally ordered or partially

(a) (b)

Figure 4.2: (a) Linear arrangement. (b) Cyclic arrangement. Diagram from [5]

ordered, in the first one there is no overlapping of events, as for the partially ordered case, one event can start before the last one ends. Branching viewpoints present subdivisions in the time axis to model multiple simultaneous alternative scenarios. This variant is modeled as a directed graph where edges direction dictates the order of the time flow. In [30] a variant of branching time is suggested, called multiple perspectives, that differs from the original one in the sense that in branching viewpoint only one of the multiple alternative scenarios will actually happen, multiple perspectives on the other hand represents the time axis in multiple views, useful for example to structure eyewitness reports. Both multiple perspectives and branching time introduce the need to incorporate probability in the visualization once that some branches can be more likely to happen than others. Figure 4.3 shows a diagram of each structure.

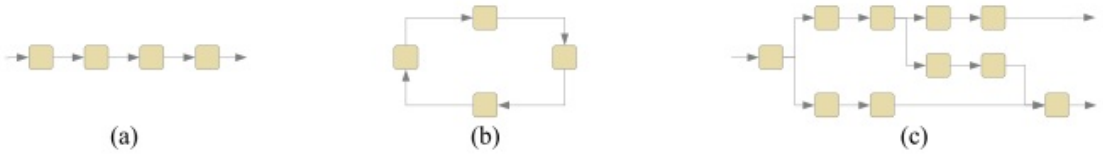


Figure 4.3: (a) Linear time. (b) Cyclic time. (c) Branching time. Diagram from [4]

4.1.5 Granularity (multiple, single and none)

The granularity is the level of abstractions for the time values available. A calendar is the best example of multiple granularity: a day is the lowest abstraction level (a chronon) , days are clustered in a week, weeks are clustered in months, and months in years. The levels of abstraction higher than the chronon level are the granules. Mappings between granularities can be regular, i.e., 1 minute always map to 60 seconds, or irregular, i.e., not every month maps to 30 days. In a representation with single granularity there is only the chronons level, and when there is no abstraction the granularity is none. Granularity is useful to map the underlying time domain to more meaningful time values, especially when dealing with cyclic time like the Gregorian Calendar. Figure 4.4 exemplify the multiple granularity of a calendar.

4.1.6 Temporal primitives (point, interval and span)

One property of the time model is the type of primitive that represent the time axis. There are two options: use time points, that mark exact time instants, or time intervals,

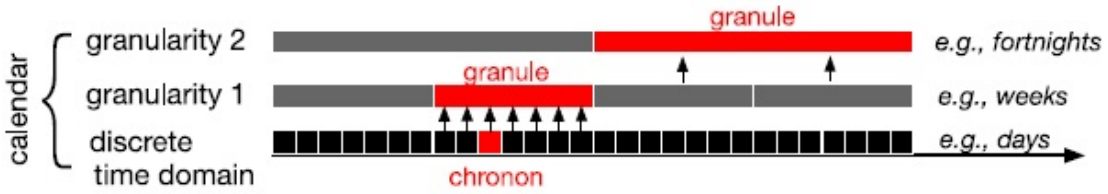


Figure 4.4: Three levels of granularity in the Gregorian Calendar. Diagram from [5]

which delimit a range of the time axis. A time point can be represented by a single values, a interval on the other hand requires two time points, one for when it starts and the other for when it ends, or a time point and a duration of how long it lasts. The time primitive used is of major importance in the visualization and analysis as the possible temporal relations change with the representation. Figure 4.5 shows the different temporal relations as described in [39]. Time points and time intervals are considered anchored primitives since they have defined begin and/or end in the time axis, but a third type of primitive exists: a span. A time span is a unanchored time primitive that has only a duration, like a interval without the start or end point.

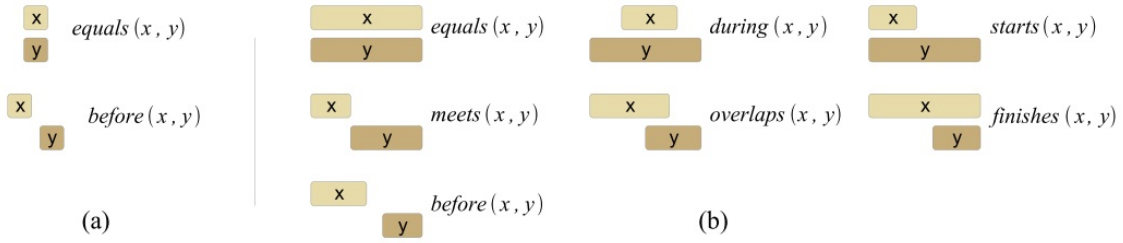


Figure 4.5: (a) Possible temporal relations between two time points. (b) Temporal relations between time intervals. Diagram from [4]

4.1.7 Determinacy

The last aspect of the time model is the determinacy. It address the level of certainty about the given time value. Uncertainty might come from inexact knowledge, like imprecise time events or future plans ("one or two days ago", "maybe it will take two weeks"), and from change in the granularity level ("I'll call you tomorrow", tomorrow is a single time value in the granularity of days, but when changing the granularity level to hours, there will be a range of 24 time values where the call can happen).

The variants of each aspect of the time model are not clearly defined in the dataset thus must be wisely chosen in order to create an insightful visualization. To understand the general trend of the dataset a linear time model is usually better, as for identifying seasonal events and variations cyclic time is more effective.

4.2 Data

The data criterion addresses what type of data is tied to the time axis, to the time primitives. The properties are: scale, frame of reference, kind of data, and number of variables.

4.2.1 Scale (quantitative and qualitative)

The data scale defines the nature of the data domain. The variants are quantitative data and qualitative data. Quantitative data is associated to a metric, that can be discrete or continuous, so different values can be compared numerically. On the other hand, qualitative data has no metric, so difference between values cannot be measured.

4.2.2 Frame of reference (abstract and spatial)

Another important aspect of a data element is the existence of a spatial relation. A spatial frame of reference implies that the data is tied to some spatial location, when no such attachment exists the data is said to be abstract. The type of frame of reference is a very important property to design a proper visualization, once that spatial data already has some clues for the visualization layout, as for abstract data, there is no such information and an effective layout must be found.

4.2.3 Kind of data (states and events)

Data can express two different situations: a state and a event. Event data highlight changes in a scenario and state data defines a span with no changes between events. An data entry like "the plane landed" is of the event kind, a similar entry of state kind would be "the plane is on the ground". The two variants are just different ways of representing a data element, but is important to make clear which one of them is used in the visualization.

4.2.4 Number of variables (univariate and multivariate)

The last property is the number of values associated to each time element. Each time element can be connected to a single value (univariate) or multiple data elements (multivariate). A wide range of time-series visualization and methods focus on the univariate case, but multivariated series are becoming a lot more frequent and make room for difficult tasks and problems like comparison and correlation of the different variables and finding different visual mappings to support visualization of the multiple variables.

Figure 4.6 summarizes the different properties of a data element.

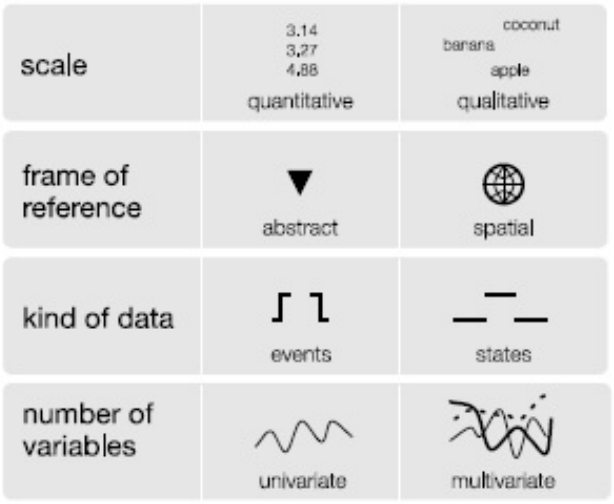


Figure 4.6: Summary of the different data properties. Diagram from [5]

4.3 Representation

In [5] the Representation aspect is named Visualization Aspects. It is our final objective: to represent a dataset with the characteristics addressed in the previous sections in such way that it can be visually analyzed. Aigner et. al. starts with the characterization of the problem as a set of three basic questions that will be subject of the next three subsections.

4.3.1 What is presented

The "what" question refers to the dataset. It is concerned with what kind of data are we interested in represent and how we model the time flow to which this dataset is tied to. The variants for each of the previous characteristics presented previously in the taxonomy of both time and data aspects must be known. For data they are: scale, frame of reference, kind and number of variables; and for time: scale, scope, arrangement and viewpoint.

4.3.2 Why is it presented

The common thought is to expect that there is a representation that will fit well when applied to any time-oriented dataset. The "why" question is probably the major cause that leads this assumption to be false. This question is concerned with the tasks that one will wish to accomplish when analyzing the dataset. The tasks are usually represented as a series of questions that an user will try to answer when interacting with the visualization. The expected tasks are closely related to the application domain, and that is why time-oriented visualizations are usually very domain specific. Task models are common in the field of human-computer interaction [23, 65]. In [53] McEachren presented a description of tasks specific to time-oriented domains, we list some of them bellow.

- **Temporal location**

  Looking for the occurrence of a given data element along the time axis. Starting from a data element, the user looks for one or more time points or intervals, i.e., "When did the value was 0?"

- **Time interval**

  Evaluate how long is the time span of a given data element. Starting from a data element the user look for the length of an time interval, i.e., "How long this event last?".

- **Temporal pattern**

  Identify occurrences of a data element or set of data elements in the time axis or in a given time interval. Starting from a given pattern of data elements, the user looks for a measure of how often such pattern appear along the time axis, i.e., "How often does the rate changes like this?".

- **Rate of change**

  Evaluate how fast a data element is changing related to the associated time element. Starting from a data element, find the magnitude of change over time, i.e., "How much is the cost increasing?".

- **Sequence**

Identify the order of occurrence of two data elements in the time axis. Starting from a data element, identify its temporal order regarding a second one, i.e., "How was born first of the two?".

- **Synchronization**

  See if there are data elements tied to the same time point or interval. Starting from a set of data elements look for time elements tied to more than one data element from the set, i.e., "Which of them work at the same time?".

In these examples the more evident distinction is between identification tasks and localization tasks. Identification regards the case where we look for data values, localization on the other hand is searching for time elements (when something happens is time). In [7] Andrienko presents a taxonomy for visualization tasks, such taxonomy is shown in figure 4.7.

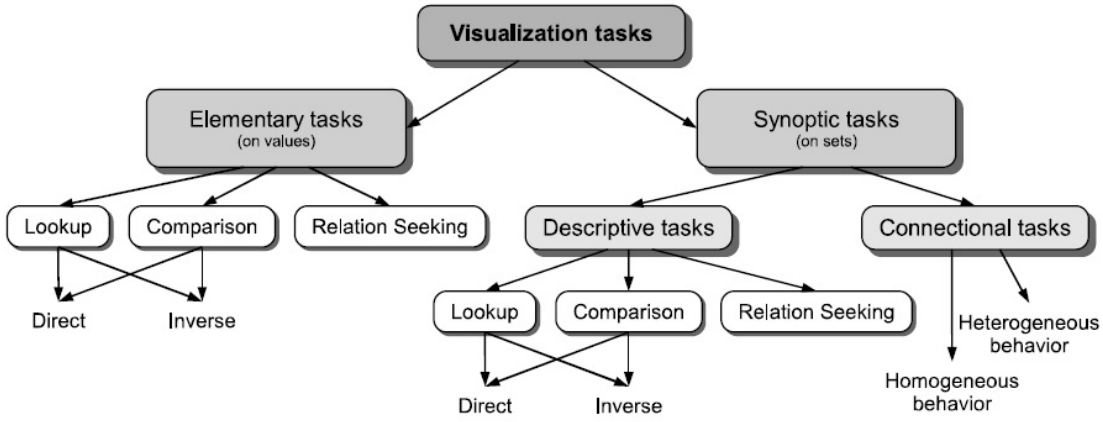


Figure 4.7: Tasks taxonomy. Diagram from [5]

The taxonomy is based in two notions: the references, which is the domain where the data has been collected (time in our scope), and the characteristics, which are the collected data itself. The first level divide the tasks in elementary and synoptic tasks. The elementary class include the tasks that deal with data elements in the individual level, it can be a single value or a structure of data, but must be a single data element, data is considered separately and not as whole. The synoptic tasks class is the opposite, it addresses tasks that handle sets of data elements, involving a general view and considering the set as whole.

In a second degree, elementary tasks are further subdivided in lookup, comparison and relation seeking tasks. A lookup task try to find a event, it can be direct, when searching data elements, or inverse, when searching for time elements. Relation seeking, as the name implies, intend to find relationships between different elements. Comparison tasks can be seen as relation seeking too, but when the relation one is looking for is not known a priori, so we want to analyze each of the properties of both elements. Comparison tasks can also be direct or inverse.

The synoptic class subdivides in descriptive and connectional tasks. Descriptive tasks try to evaluate the properties of the set, that can be assembled of references (time elements) or characteristics (data elements), and further differentiate in the same categories

of elementary tasks: lookup, comparison and relation seeking. The connectional tasks, however, create links between two or more sets of elements based on the relational behavior of a set of underlying variables. The class of connectional tasks subdivides in homogeneous and heterogeneous tasks depending if the related variables are from the same set of reference or no, repectively.

4.3.3 How is it presented

This last question is about our final objective: how to visually represent a time-oriented dataset. Now it should be already clear the dependence between the answer to this question and the previous ones about data and time models, and user tasks. There is already a large number of visualization of time-oriented data, each representing a different answer to the "how" question. In the next chapter we will present some of the most interesting and recent solutions to the visualization of time-series. For the time being, we keep following the taxonomy from Aigner et. al. [5] by focusing on two base criteria to classify presentation methods: the mapping of time and the dimensionality of the presentation space.

Time mapping

To be visually represented, time itself must also be mapped, like every abstract data, to spatial entities, and its relevant attributes to visual attributes of the chosen entity. The common choices in this mapping is geometric entities and its colors, but the natural human perception of time can also be exploited by mapping the dataset time axis to physical time, to the dynamics of the presentation. We now have two options: mapping time to space or mapping time to time itself. The first one results in a single representation where time and data must share the same space, the second one, on the other hand, leaves more room to the spatial mapping of the data, however, the representation will evolve, changing over time, and time elements will be mapped to each stage of the final animation. The first approach is named static and the second as dynamic. It is important to note that the availability of interaction in the visualization has no influence in this classification, static representations can have interaction support and dynamic ones can provide no interaction at all.

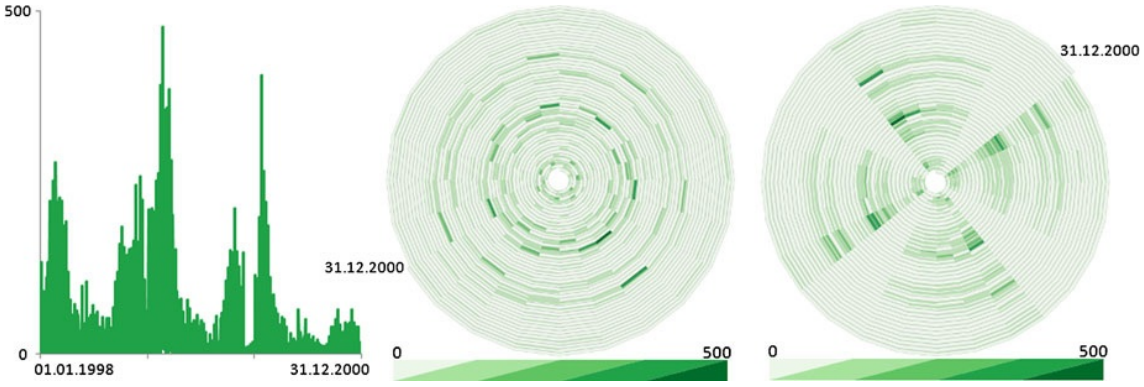


Figure 4.8: Bar graphs are one of the most common way of visually representing time-oriented data by assigning the time axis to horizontal dimension of the display space. Cyclic graphs represent the time axis as a spiral and are usually used to represent cycic time. Diagram from [5]

To the visual mapping of time the most common approach is assign one of the display dimensions to represent the time axis resulting in the widely known line charts, bar charts

and alike. This representation is very intuitive and practical but when trying to fit a large amount of data, or when the display area is small, this approach is not satisfactory. In [42] Javed et. al. compare 3 types of such representation of time-series regarding perception and spatial needs, also introducing a novel one called braided graphs (see Figure 4.9). Another common method to represent time in space is using both display dimensions to define a time spiral. Time spiral is the usual choice to visualize cyclic time. Figure 4.8 shows both representations of time in space. Different and less usual mapping schemes include angle, line width, brightness, containment, connections and legend, examples are shown in figure 4.10.

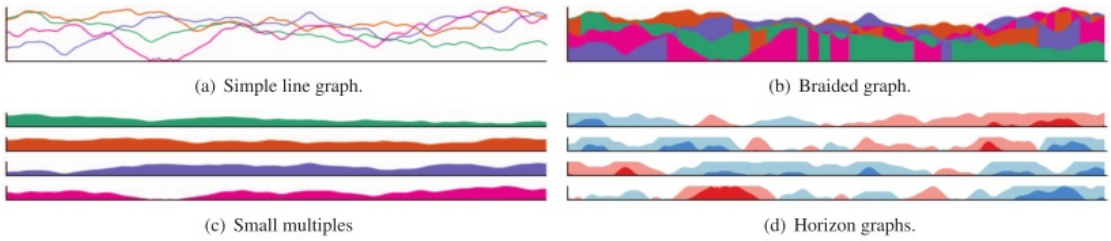


Figure 4.9: Different types of line graphs compared in by Javed et. al. in [42]. d) Braid graph, a new graph proposed by Javed et. al. that changes the order that the graphs are drawn to reduce the overlapping of series. Image from [42]

As said before, sometimes screen space is not enough to represent both data and time, it is quite frequent to happen when dealing with geographic referenced information, multivariated data and graphs networks, so an alternative is representing time with the physical time, creating an dynamic presentation, to leave more space for the data elements mapping. Dynamic presentations can be seen as animations or slide shows depending on the number of frames shown per second. The dynamic approach may seen to be the better choice as it represent time in the way we are used to understand, however there are issues that must be noted. One example is that one might expect to be a one-to-one mapping between time elements and a frame of the representation in such way that time is represented authentically, but that is not what usually happens as time elements may be aggregated in a single frame due to a high amount of time elements, or additional frames (and time elements) may be created for interpolation when there is few real time elements in the dataset. Another issue is that the speed of the representation must be chosen carefully to avoid false impression of the dynamics. Animations that are too fast will be difficult for the user to keep up with, losing important information, and the ones that are too long may become boring to follow, so it is important to provide means to control the time flow. Also using a large dataset, like multivareted ones may result in an overflow of information that the user cannot follow.

Dimensionality of the presentation space

The presentation space can be 2D or 3D. 2D representation make use of the dimensions of the display and restrict the mapping options to 2D shapes. The 3D case adds a new axis to encode information, enabling the use of 3D geometry for visual mapping, but require a projection stage to map the 3D geometry to 2D space, once that the display devices have only two dimensions. 3D representations exploit the human visual system in its natural capacity to perceive our three dimensional world, however, there is no consensus on if adding a new dimension adds to the visualization, due to issues, like distortion by perspective projection and occlusion, introduced by 3D representations. On the other

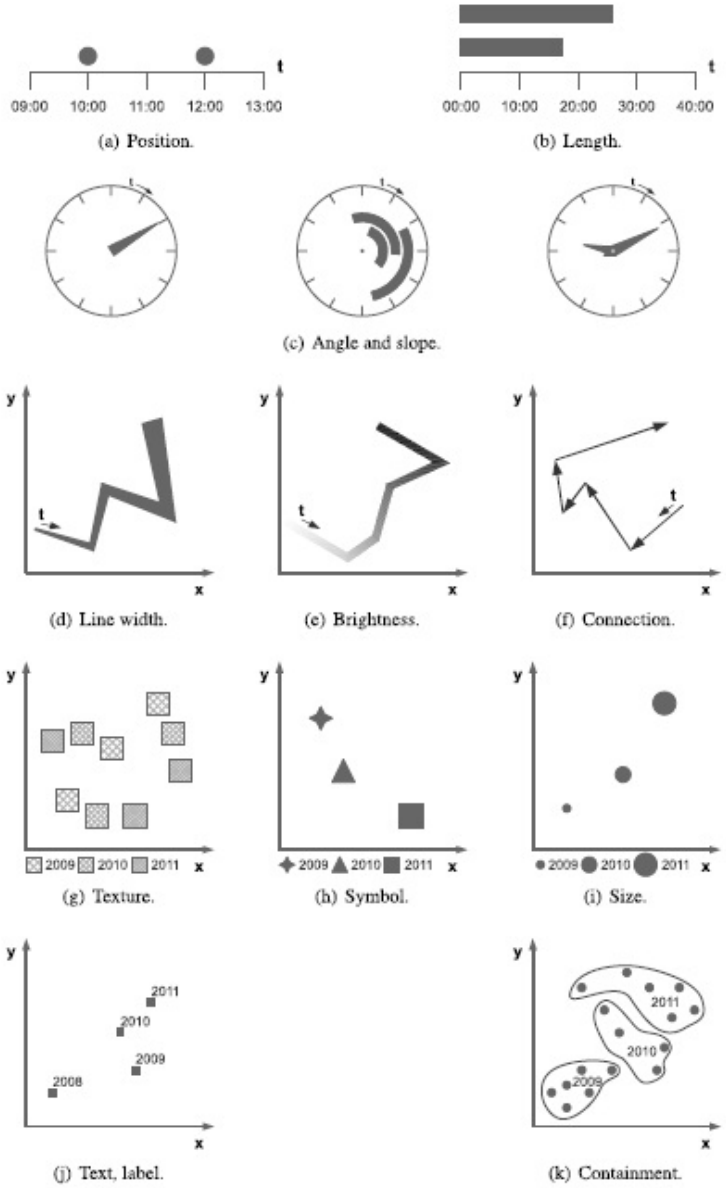


Figure 4.10: Different ways of mapping time to space. Diagram from [5]

hand, the 3D approach can be useful in some cases, like when working with spatial data that already provides spatial information for layout and would benefit from one more. Multivariated and large datasets that easily create cluttering in 2D space also would benefit from an extra dimension, given that some mandatory interaction means and visual cues are provided. In [25], Elmqvist et. al. present a study on techniques to handle 3D occlusion. Figure 4.11 shows a diagram of the answers to the three important questions in the creation of visual representations.

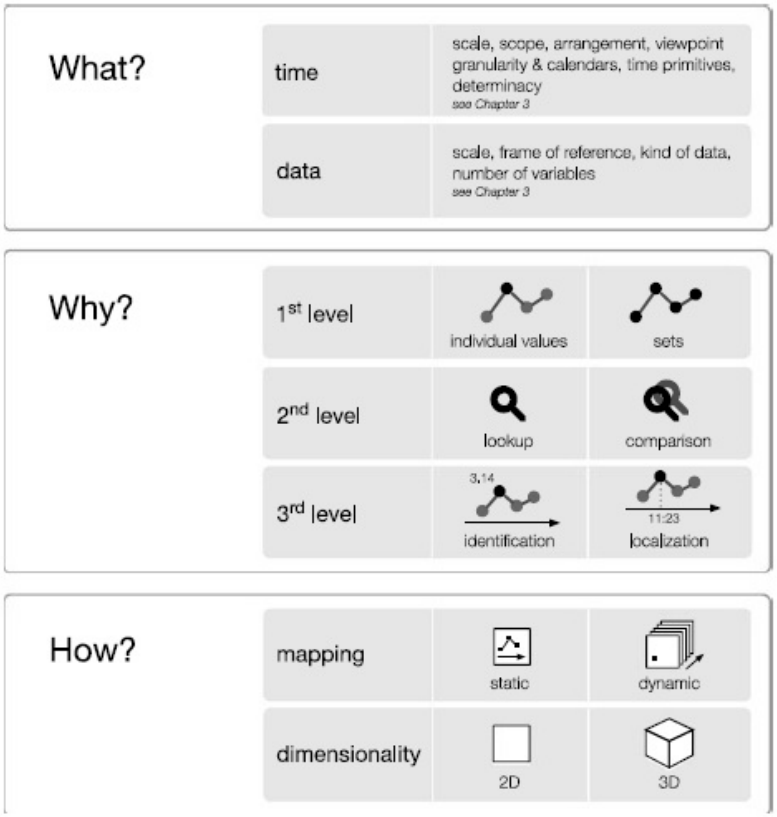


Figure 4.11: Different ways of mapping time to space. Diagram from [5]

5 DESIGN ASPECTS FOR EXPLORATORY ANALYSIS

When the temporal data is augmented with spatial attributes, a different set of tasks appear as the result of the spatio-temporal entanglement. In their book, Andrienko and Andrienko [9] present a systematic overview of the exploratory analysis of spatio-temporal data. Their work is our major reference on the subject, and we present now some of the concepts presented, on which our design procedures were based.

One key notion, is the definition of the functional view of data structures. The book [9] introduces the concept of a dataset as a function that express a correspondence between referential and characteristic components (referrers and attributes) , the same way that a mathematical function maps values from one input domain to an output one. Figure 5.1 shows this functional view of the dataset as a *data function*, as they call it.

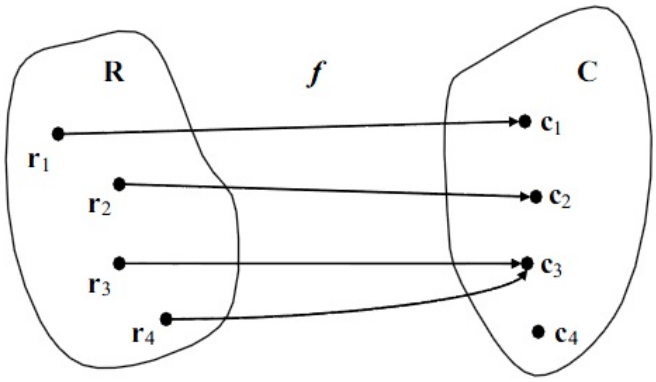


Figure 5.1: Functional view of a dataset. Data as function of correspondence between elements in the references set and elements in the characteristics set. (from [8])

R is the set of all references (combination of values of the dataset referrers, a tuple with a value for each independent variable, i.e. GPS coordinates and time stamp) while *C* is a set of characteristics (values of attributes, dependent variables, i.e. density, precipitation). Function f is the correspondence between each element of the reference set f and a specific element in the characteristic set *C*. The diagram was designed to convey three properties of the data function:

- each reference set element corresponds to a single characteristic set element;
- characteristics corresponding to different reference elements can coincide;
- some combination of characteristics may not appear in the dataset, i.e. there is no references combination that corresponds to them.

5.1 Tasks

The functional view of the data is used to identify the types of tasks usually involved in the exploratory analysis. The typology proposed by Andrienko [8] is designed around the ideas expressed by Bertin in his work on semiology of graphics [16], distinguishing tasks by their level of data analysis ("reading level" in [16]) but adding the division of data components between referrers and attributes. For completeness, we show a table (Table 5.1) that depicts Bertin's [16] reading level view of tasks. In this view, tasks type are defined as the combinations of 3 reading levels in space and time domains.

Time \ Space	Elementary level	Intermediate level	Overall level
Elementary level	What is the population density at location P at time t_i?	In which neighbourhoods is the population density d_2 at time t_i?	Where does the highest population density occur at time t_i?
Intermediate level	How does the population density develop at location P from time t_i to time t_j?	In which neighbourhoods is the population density d_2 during the time period from t_i to t_j?	Where does the highest population density occur during the time period from t_i to t_j?
Overall level	What is the trend in population density at location P over the whole time?	Which are the neighbourhoods where the population density remains at d_2 during the whole time?	What is the trend in high population densities over the whole time?

Table 5.1: Bertin's view of tasks according to reading levels (from [8])

Andrienko [8], define tasks as questions composed of two parts: the target and the constraints. The target is the information we want to obtain, while the constraints are the conditions this information needs to fulfill. The two parts can be seen as unknown and known information, in this order. So the objective is finding the unknown information that correspond to the specified one. Recalling the dataset functional formulation, a task can be expressed by finding the elements (or elements) in the reference or characteristic sets, that correspond to a given element (or elements) in the other set. Following this rationale, they differentiate elementary tasks from higher level ones.

Elementary tasks address individual references or characteristics, they are not concerned about reference set or characteristic sets as a whole, but in their elements. Tasks that look for insight from sets of characteristics and references, viewing them as whole are named *Synoptic* or *General*. This type of tasks focus on behaviors and patterns, instead of data elements. Both elementary and synoptic tasks can also be of the *Lookup* class or the *Comparison* class.

Lookup tasks aim at finding values of data components that correspond to given values of other components according to the data function. These can be *Direct*, when references are specified and the target are the correspondent characteristics, or *Inverse* when we look for references related to given characteristics. Comparison tasks try to determine what

are the relations between characteristics, or between references. Comparison can also be direct, when comparing characteristics, or inverse, i.e. comparison of references.

	Elementary	Synoptic
Direct Lookup	On a given date, what is the price of stock X?	During a given time interval, what was the trend of the stock price?
Inverse Lookup	Find municipalities that had 300 000 or more inhabitants in 1991.	Find time intervals in which the stock price increased.
Direct Comparison	On a given date, did the stock price exceed €1000?	Compare the behaviors of the stock price during the first and the second week.
Inverse Comparison	Did the stock price reach €1000 before or after the given date?	What are the relative positions of the clusters of high and low crime rates?

Table 5.2: Examples of tasks by types.

The typology of tasks presented in [8] is a generalization to represent tasks in exploratory analysis in any domain. We will follow the one given another work from Andrienko et al. [10] that specializes the one introduced so far to the case of exploratory analysis of spatio-temporal datasets. Figure 5.2 summarizes the typology we adopted from [10]. In this typology, a task type is defined as a combination of a search level, a search target, and a cognitive operation. The cognitive operations are identification (lookup in the general typology presented in [8]) and comparison (the same in [8]). Search level is related to the plurality of the reference and characteristic sets, being elementary or general (Synoptic). The differentiation comes from the two search targets:

1. when $\rightarrow$ where + what. Time is given while the other data components (space and other attributes) need to be discovered and described.

2. where + what $\rightarrow$ time. Time needs to be discovered while the other data components are used as constraints.

5.2 Tools

Both works follow the typology of tasks with a typology of exploratory tools to support them. In the general analysis from [8], the tools that support exploratory analysis, are classified as follows:

- **Visualization Tools:** In this category represent data in visual form with graphs, plots, diagrams, maps and others. Data elements are translated to graphic features like display positions, shapes, sizes and colors. Solutions put together techniques from the field of information visualization and human perception.

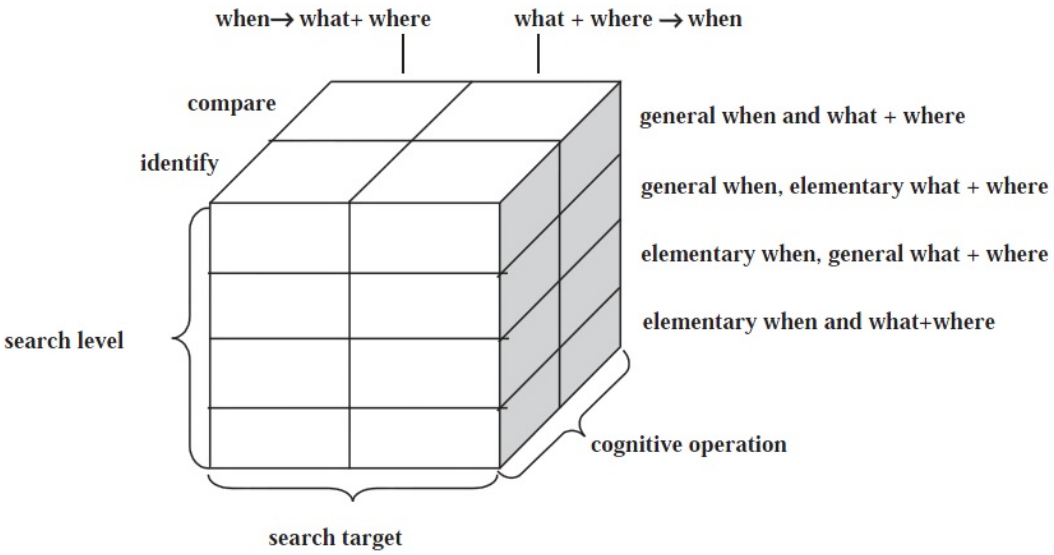


Figure 5.2: Task typology for spatio-temporal data analysis. (from [10])

- **Display manipulation:** This class represents techniques to support the interactive modification of the visualizations, enhancing the displayed result to make it easier to perceive features of the data analyzed. The visual result is modified through the manipulation of the function that convert data into visual properties, the visual encoding function. Examples of display manipulation tools are: ordering, eliminating excessive detail, classification, zooming, and changing the parameters of encoding functions. Examples of display manipulation tools are:

  - *ordering, reordering and arrangements* to change the position of the displayed elements, with the purpose of providing a clearer overall view of the distribution of the characteristics of a dataset and of relations between different attributes.
  - *generalization* to get rid of irrelevant detail and noise to reveal the main features of a behavior.
  - *classification* is a type of generalization that visually groups references with similar characteristics in sets that are then regarded as being identical.
  - *zooming* to reduce the number of data components represented in the display so that the remaining ones can be displayed with higher expressiveness.

- **Data manipulation:** tools to derivate new references and characteristics from the initial ones. The two major purposes of these tools are to simplify the data, making it easier to analyze; or to improve the dataset by adding different aspects of the original data components.Examples of operations for data manipulation are:

  - *smoothing* to reduce noise.
  - *interpolation* to eliminate discontinuities.
  - *aggregation and integration* to reduce the amount of data under analysis.
  - *interpolation and extrapolation* to introduce additional references and estimate their corresponding characteristics.

- *normalization* to standardise values, thus achieving compatibility between different attributes.

Querying: Tools to automate the search for answers to questions specified by the analyst. Common uses are searching for references that correspond to given characteristics or finding the characteristics of specified references. The techniques in this class are usually concerned with providing precise answers quickly enough so that queries can be performed dynamically.

Computation: This class includes computational techniques from the field of data-mining and machine learning. While data manipulation tools transform data into a more suitable form for further analysis, computational tools are usually focused in extracting essential data features.

Part II

Original Work

The problem we are interested is the one of representing many time series (with spatial properties) at the same time, for visual inspection. In the previous part, we covered some topics related to the analysis of time series, to create an understanding of why the visualization of multiple time series at once is important. In Chapter 3 we presented aspects of working with time series nowadays in the data mining domain. The problem then was to represent large amounts of temporal data (databases of thousands of series) in such way that it can be managed properly, through means of representation and comparison methods. We gave an overview of the major options for both tasks. Then, Chapter 4 brought a review of aspects concerning the visual representation of temporal data from the domain of information visualization. Beginning with the properties of the temporal attribute itself, then to those of the data attached to it and finally to how to represent them visually. The last chapter (Chapter 5), ended the first part with important aspects related to the design of exploratory solutions. In that section, we reviewed the classes of tasks that analysts perform during exploration of datasets, and the tools that compose exploratory systems to support performing them.

In this second part we introduce the solutions we designed to support the exploration of spatio-temporal data, considering that in our context of data analysis, datasets consist of systems of time series.

6 VISUALIZING GROUPS OF TIME SERIES WITH SPATIAL PROPERTIES

There are two common approaches to visualizing temporal data that has spatial properties. The first one is the space-time cube design. In this method, the data is visualized in a 3-dimensional setting where the two horizontal dimensions represents the spatial coordinates while the vertical one is used for temporal representation (some examples are shown in figure 6.1). The second consists of combining multiple views, usually a representation of the dataset from the temporal aspect coordinated to a map that show the spatial features; each view allowing for navigation in temporal and spatial domains respectively. We have opted for this coordinate views approach as 3-dimensional settings are known in the field of information visualization for requiring more complex interaction from the analyst [4], and also because such visualizations can also be used in coordinate settings, thus benefiting from improvements in this scheme.

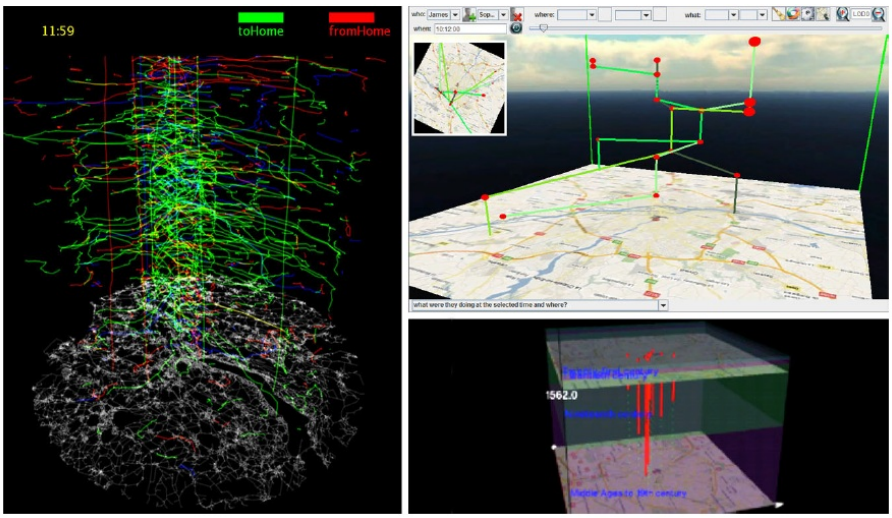


Figure 6.1: [Space-time cube applications] Some applications using the space-time cube design to visualize simulation data (left), trajectories (top-right) and history of points-of-interest (bottom-right). (from [84])

In figure 6.2 we show an example of a system with coordinate views (taken from [10]), to explore a dataset of unemployment rates in the provinces of Italy over a period of 14 years. The bottom component represents the temporal components of the dataset, plotting a line graph for each time series as the unemployment rates of each municipality. The use of this kind of component is frequent in spatio-temporal visualizations to support navigation in the time axis and the inspection of the different time series, as shown in

the example where selecting a province in the map highlights the respective series below. But as the number of series increases the visualization becomes cluttered, losing informative power as one cannot tell series apart from one another, nor identify patterns or spot discrepancies.

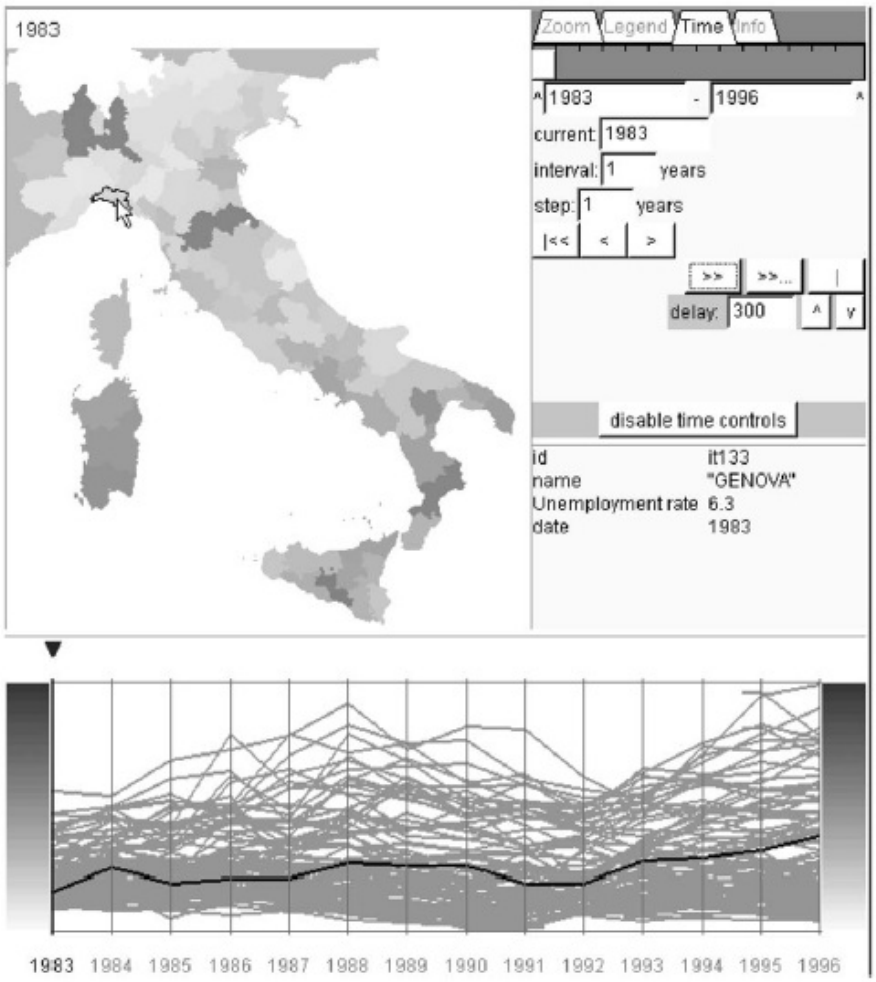


Figure 6.2: [Spatio-temporal coordinate views] An interactive time-series graph dynamically linked to a map. (from [10])

Many works have addressed the representation of multiple time series altogether, like CloudLines [49], Horizon Graphs [69], Braided graphs [42], TimeWheel and MultiComb presented in [73] (see figure 6.3). CloudLines [49] describes a technique to visualize multiple time-series, where values are represented by the width of a line. Applying this approach to datasets composed of a few dozen series becomes confusing due to the visualization area required to represent the line thickness. Horizon Graphs [69] reduces the spatial requirement by stacking sections of the area graphs, but it still cannot represent sets of hundreds of series. Braided graphs [42] succeeds at visualizing multiple series in a single frame through sectioning, ordering, and color labels. However, the scheme quickly leads to confusion when the number of series increases. While these solutions can effectively represent many series in a single view, the spatial requirement is still too much for actual datasets that can easily have hundreds to thousands of series. Sets of only a few dozen series can visualized with Horizon Graphs or CloudLines, while such amount is already too much for presentation as Braided Graphs and TimeWheels.

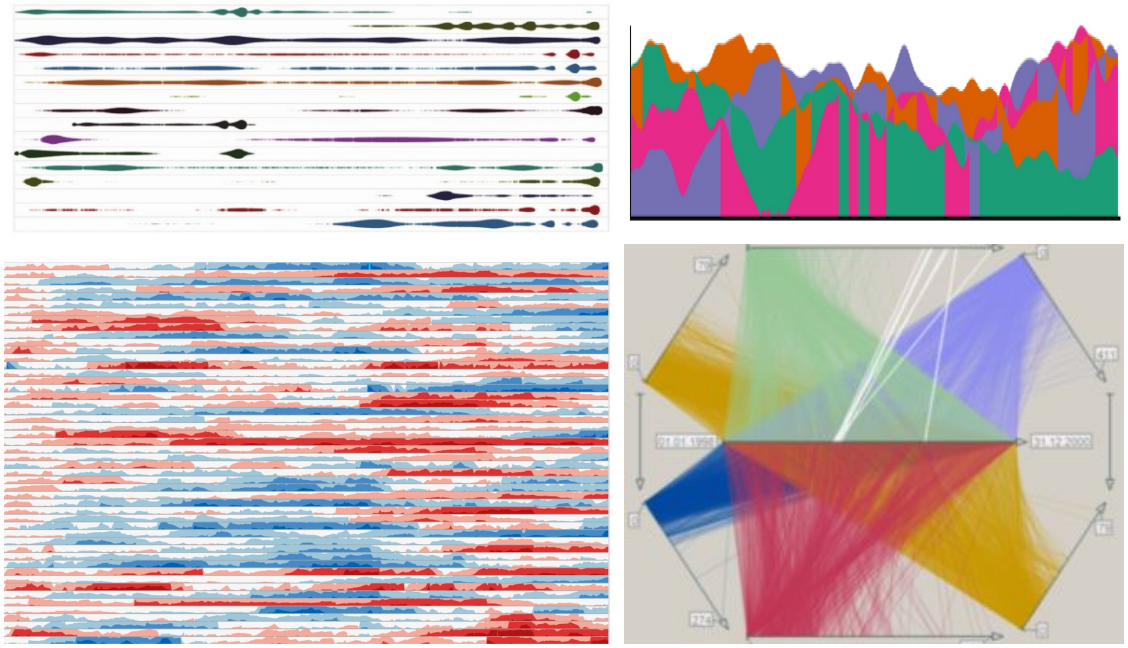


Figure 6.3: [Multi-series visualization techniques] Visualization techniques to represent many series in a single view: (top-left) CloudLines [49], (top-right) Braided graphs [42], (bottom-left) Horizon Graphs [69] and (bottom-right) TimeWheel [73].

Many works like Sequence Surveyor from Albers et al. [6], LiveRac from Mclachlan et al. [56] and Hao et al. [40] represent time series as horizontal stripes, in which varying color represents the change in value through time according, to a color scale, as in a heatmap [32]. Exploiting this representation, Kincaid and Lam [48] compressed the stripes to horizontal lines of even 1-pixel thick to visualize groups of hundreds of series (see figure 6.4).

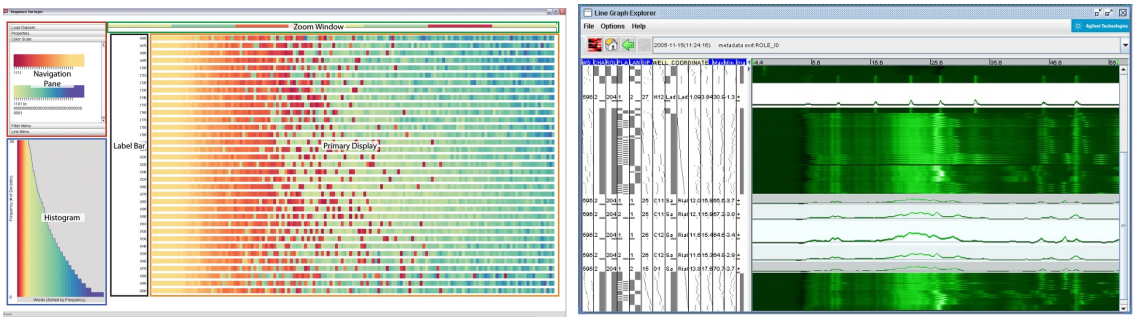


Figure 6.4: [Compressed view of multiple series] (Left) Sequence Surveyor [6] uses color to map values of time series represented as a horizontal stripes. (Right) Line Graph Explorer [48] further compress each series into a single line to show hundreds of series in the same view.

We saw this compressed view as an interesting asset in the design of coordinate views for spatio-temporal datasets and applied it in two case studies, from different application domains, to validate our point. We made a series of adaptations to the general design to conform to series with geographic coordinates represented by three different data structures with distinct semantics. The first case study supports the analysis of datasets of raw series with measurements about hundreds of runners as they compete against each

other in a running race. We show the adjustment of the compressed view to a scenario of irregularity, where series have a high and varying amount of samples. Also, we used animation to represent the temporal dimension with the real physical time, allowing us to use the visual space to map distance, creating an intuitive visual representation of the race. The second case study explores the dynamics of systems of bike sharing in a city. We modified the compressed view to two data structures aggregated to frame the series into common regular representation; the first with measures of level of resources (bikes) at different locations (stations) in the system, and the second with events (trips) containing spatial properties, i.e. origin, destination, direction and distance. Figure 6.5 shows a quick overview of our adapted views. In the next sections, we present each work separately. Both studies resulted in the implementation of prototypes to validate the applicability of the proposed designs to the analysis of several real datasets.

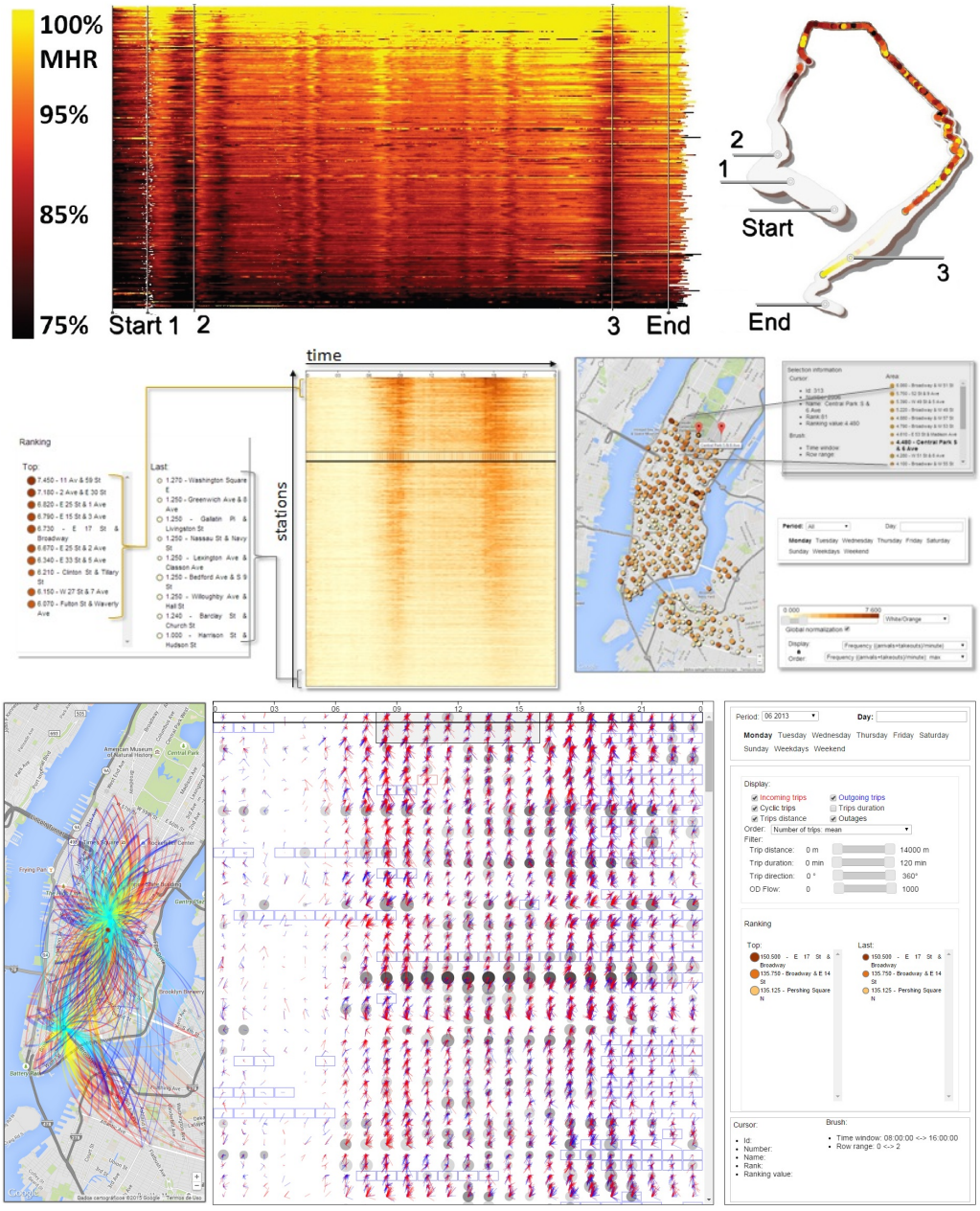


Figure 6.5: [Compressed series for spatio-temporal data] Our adaptations of the compressed representation to different datasets. From top to bottom visualization of: the progress of runners in running races, the balance of bikes among the stations in a bike-sharing system, and the trips between the stations.

7 RUNNING RACES

HR monitors were introduced in the 70s to help athletes record heart-rate activity during competition which could be used in a subsequent analysis to improve performance. Such devices comprise a Heart Rate (HR) monitor (incorporated into a wrist receiver) and a chest strap transmitter. Data recorded can be as complex as the time-series containing heartbeat during the entire exercise, as well as other information such as speed, geolocation, etc. The affordability of such devices has made them popular recently. In addition, activity recorded from monitors can be uploaded to computers, where they can be inspected or shared with others (e.g. a persons physician).

Drawn by the novelty and popularity of the data, we wondered how visualization could help people to understand better their performance when exercising. The focus of current visualization tools, at the time, was on the visualization of a single activity, often lacking the ability to compare multiple activities. However, inspecting multiple activities at the same time can be very useful to compare the effort of different runners in a given activity, or to compare the effort of a single person against others in a shared activity, so we chose to address this problem in this first work. The analysis of such dataset is challenging since it contains the multivariate time-series data generated by HR monitors for multiple runners.

In collaboration with an expert in exercise physiology, we formulated questions to be answered about the running race, guiding the conception of the visualization designs. To validate our designs, we implemented an interactive tool, created use cases with real data from different running races, and assessed how helpful each design was in answering the questions posed by the expert.

7.1 Related Works

There are several studies that correlate fitness levels with well-being [18, 17, 21]. The Physical Activity Guidelines Report [41] presents proofs of the importance of physical fitness. In [21] consistent evidence is given about the direct association of myocardial infarction to physical inactivity and that people with low fitness levels have a higher risk of developing cardiovascular diseases. Running helps improve physical fitness, and there are studies [38, 15] that correlate heart health to the time one takes to run a given distance. Running data, therefore, can provide important indicators of overall fitness. For this analysis, the heartbeat of each individual can be normalized as a percentage of the individual's maximum heart rate (MHR), and different effort levels can be identified using this information. In 7.1, from [14], four main HR zones are identified, each with different characteristics. Training programs often rely on defining how long or how far a runner should stay in each HR zone. Exercising at HR zone 4 for a long period can be dangerous,

and there is a great concern about mortality and cardiac diseases [47, 55, 79]. Related to this is the study on human limits [57, 43] and strategies in different aspects like hydration [13] and energy consumption [72].

HR Zone	Effort Index	Effort Level	Pace	Fuel Source
1	60-75%	Easy	Slow	Primarily Fats
2	75-85%	Moderate	Moderate	Carbs and Fats
3	85-95-%	Difficult	Fast	Primarily Carbs
4	95-100%	Very Hard	Sprint	All Carbs

Figure 7.1: **Heart Rate Training Zones.** Effort index given as percentage of the maximum heart rate. (from [14])

There is already a wide variety of tools to manage fitness data. However, whether their purpose is the management of the exercise records of a single casual runner or the analysis of data of several professional elite athletes by the perspective of a fitness trainer, those systems do not support the analysis of several exercises altogether. While some give statistical summaries of the dataset and support the overplotting of a few line graphs, most only allows the analysis of a single exercise at a time. In figure 7.2 we show some popular systems to explore data from running exercises. Training Peak's WKO+ tool support the management of the exercises of many runners. It provides several charts to summarize their individual data and allows the plotting of different variables as line graphs in the same reference frame. However, this view is usually cluttered due to the number of line graphs that overlap. Also, there is no support for the comparison of exercises of different runners, even though the tool manages data of groups of them. RubyTrack puts together the exercises of a runner into a intuitive training profile, it also shows many variables in the same view with lines or bars, and provides a view of the course as well, but again, the user cannot view a set of exercises at the same time. Garmin Connect provides similar functionalities of those of RubyTrack with the addition of an animated view of the exercise progression in the map. However, it also suffers from the same limitations.

Figure 7.2: **Running Data Analysis Tools.** Some popular tools for analysis of running data (from left to right: Training Peaks WKO+, RubyTrack and Garmin Connect). Designed as activity managers, these tools provide statistics about the collection of activities of a user, and can visualize each activity in detail separately, but there is little support to the visual analysis of many activities at once.

7.2 Materials and Method

7.2.1 Data

The Garmin Connect website [35] stores a massive amount of public training data, uploaded by users, that can be filtered according to date, total distance, location, etc. Each activity has a time-ordered sequence of trackpoints exported in the Garmin's Training Center XML format (TCX). Trackpoints are samples of a time series, each with a time instant as key property plus several other variables. The number of variables available depends on the model of the HR monitor. We constrained this range to HR, speed, altitude, latitude, and longitude. Different monitors and other accessories can store other data (e.g. cadence and calories), but their low availability could significantly reduce the size of our datasets. Figure 7.3 shows an example of the content of a TCX file with a diagram of it's structure as a list of trackpoints.

Figure 7.3: **TCX file** Structure of a TCX file. Each file represents an activity as a ordered set of trackpoints. Each trackpoint stores the value of a number of variables for a given unique timestamp.

We implemented a web crawler to add a level of automatization in our data acquisition stage. The process of fetching public exercises from Garmin Connect website is shown in figure 7.4. The website provides a system to query exercises using a set of parameters to filter the results returned by the engine. To fetch the files of runners that took part in a given race, we zoom the map view to the place where the race happened and restrict the time window for filtering to the beginning of the event. We also filtered by total distance, to remove runners that did not finish the race. Once Garmin's engine return the activities that passed the filtering (listed in area highlighted in green in the figure), we activate the crawler (bookmark in red) to go through the pages of results downloading each TCX file (in the download bar list in orange). We applied this process to gather data from two short races and two longer races. For the longer ones, we got data from the event in 2 consecutive years for comparison, resulting in 6 datasets, each with data of one event in one year. Each dataset is comprised of a set of TCX files, with each file having the data of a single runner in the same race. We used datasets of sizes ranging from 60 to 483 runners. The number of trackpoints of each runner in a dataset depends on the monitor sampling

rate, and the distance of the race. In the smallest dataset, the number of trackpoints per runner was between 200 and 1000 with an average of 89 trackpoints per minute (1.48 Hz). For the largest dataset, in both distance and number of runners, it ranged from 3000 to 7000 with an average of 218 trackpoints per minute (3.6 Hz).

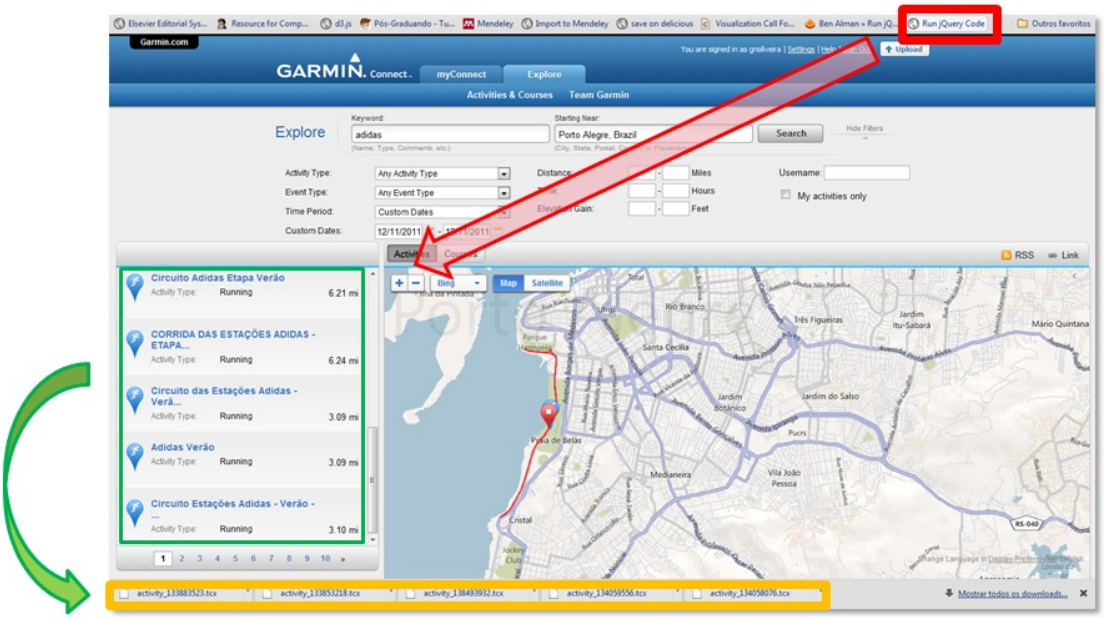


Figure 7.4: **TCX Crawler** Using a web crawler to acquire tcx files. After filtering public activities by place, time, and other properties in Garmin Connect, we active the crawler script to automatically download the activities returned.

Another information available in each TCX file was the maximum heart rate (MHR), which is a data informed by the runner. The MHR is used as a standardization to classify the intensity level of the activity, which is used to stablish a relation between health and fitness condition [41]. For running exercises this standardization is expressed as a percent of a person's aerobic capacity (VO2max), or as a percent of a person's measured or estimated MHR. Another approach is to use the VO2 reserve and HR reserve instead of VO2max and MHR respectively. The reserve is the difference between the maximum value of the variable (VO2max and MHR) and the measured value when the person is at resting state. As we do not have VO2 data available nor the measurements at resting state, we use the HR value normalized as percentage of the runner MHR. The MHR is provided by the user to Garmin's system, but even though the runner may have entered a poor estimation of the MHR value, our analysis focus on the variation of intensity along the race, so a biased MHR makes little difference. Also, the provided MHR is only used if in fact it is higher than the maximum value found in the current exercise. The advantage of using a trusty MHR estimate is to be able to evaluate how effort demanding such exercise was to the runner.

7.2.2 Desiderata

The conception of the visualization designs was driven towards helping answering questions about a street race using public data from several participants. The questions are usually asked by someone organizing a race or by a physiological specialist. Those have been selected based on the experience of one of the authors, who is a physiological

researcher, that usually searches for the answers without a visualization tool. Below, we enumerate such questions:

1. Is there a predominant effort level in the race? Where does the effort level changes?
2. Which parts of the race require more effort?
3. Are there any common patterns among runners during the race? Can we identify the source of such patterns?
4. Can we identify the running strategies for a given race?
5. Are there people running at dangerous effort levels?
6. How can we compare races?

7.2.3 Design

Once our goals were defined, we created distinct visualization designs, which are described first individually, and later by their common functionalities.

7.2.3.1 Visualization Design 1: Line Graph Heatmap

The building blocks of the visualization designs are the activities trackpoints. Trackpoints are samples of the athlete's state collected by the monitor device at regular intervals. Common to the visualization designs is the rendering of consecutive trackpoints using a solid circle (particle). Each trackpoint is rendered only if the current visualization time is within the range of the trackpoint. After rendered, each trackpoint fades according to user parameters.

The first visualization displays multiple time series (runners) similar to a conventional line graph with a set of improvements. The vertical axis represents the effort level, while the horizontal axis represents the distance from the start line, see Figure 7.5. The color of the trackpoint represents the HR value of the trackpoint in relation to the MHR and is mapped to a color gradient (black, red, orange, and yellow). The color mapping scheme is designed to relate HR to the respective effort zones. HR training corresponds to the use of HR data to customize a given workout to improve runner's performance [20, 31, 14]. The cardiovascular system reflects the body stress at any given moment, and by keeping track of the HR one can estimate the effort at any given time. Differences in HR measurements, when compared to rest state, can provide immediate feedback on how tired the body is. Those differences indicate how hard the body is working, and how adapted it is to a given workload.

The intensity of the exercise is closely related to the actual HR reading (as a percentage of the MHR) [14]. Based on the HR it is possible to identify the effort level, energy source, and performance-related fitness that will benefit from the activity. Notice from 7.1 that the range of the different effort zones is not the same. We used layers with the same thickness to lay emphasis on the variation in the zones of higher effort, which is the focus of most of the questions presented before. In other words, the changes in the low effort zone (black layer) become smoother while the change of the visualization in the high effort zone (yellow layer) is more noticeable.

The color of each trackpoint is linearly interpolated to a given color-scale based on the layer it belongs. Each layer keeps track of how many runners are inside its effort zone at any given time. This is visually represented by filling a part of the layer area, bottom to

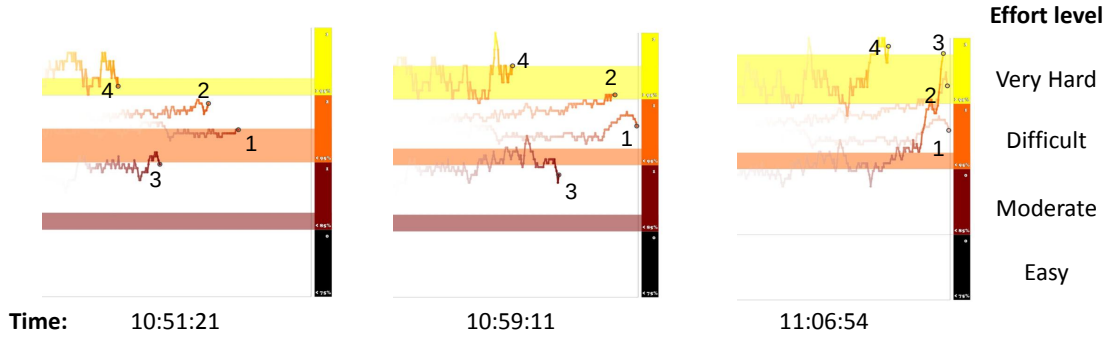


Figure 7.5: **Line Graph Heatmap** Consecutive frames of the line graph heatmap showing how the number of runners in each effort zone is represented. In the four vertical layers, from yellow to black, we display the different effort levels. From left to right we display the HR for 4 runners (illustrated as points 1 to 4) as function of distance. We used horizontal strips in each effort layer with varying height to encode the number of runners in a given time at that effort level. Such design allows to verify how many runners are in a given effort zone, and how the runners relate to the others.

top, proportional to the percentage of runners inside it, with a faded version of the layer color (see 7.5). This allows to keep track of a runner and see if his condition is similar to others. Furthermore, we can see the distribution of effort at different times during the race (Q1). One problem of the line graph heatmap is information overlap. To highlight the actual state of each runner we add a border to the current trackpoint. To reduce the clutter that comes from the overlap with past trackpoints, an adjustable decay factor provides a tradeoff between history length and readability that can be modified during visualization.

7.2.3.2 Visualization Design 2: Linear Heatmap

The second visualization component in this solution is our adaptation of the compressed view as discussed before. In this application, it represents each runner's activity as a horizontal line with colors to indicate the effort level and horizontal space to represent distance covered. The prior design, the line graphs heatmap, is useful to see the overall effort level at different times, but not at different places along the course. If the decay was reduced until the trails become the whole line graphs, such comparison would be possible, however, the overlap of lines would still persist and complicate such task. The purpose of this new design is to provide a way to compare the effort and speed of the runners, while avoiding data overlap. The trackpoints are positioned from left to right according to the distance covered and rendered when the animation time surpass the trackpoint key time. The result is a set of horizontal lines starting as dots in the left and increasing in length to the right as the runners get closer to the goal, as figure 7.6 exemplifies.

We also used a different colormap to represent the speed in the trackpoints. The datasets usually contain outliers (measurement errors in the monitor device) in the speed values way above the average speed. In this case, normalizing the values based on the minimum and maximum would bias the result, bringing almost every trackpoint to the same small portion of the color range. To solve this issue, we use a blue-white-red colormap, where white is the average speed in the whole dataset, red trackpoints that are N times the standard deviation above the average speed, and blue the ones that are N times the standard deviation below it, with N being a parameter that can be modified in real time. Without the overlapping of runners, we can now identify the predominant effort

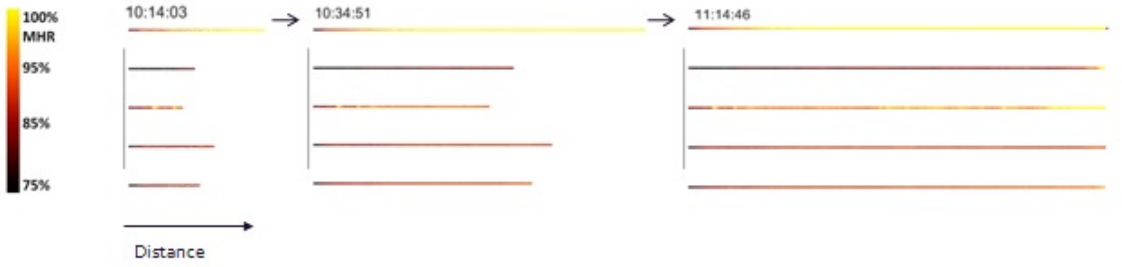


Figure 7.6: **Linear series scheme.** Series represented as horizontal lines. Length represents the position of the runner at the time.

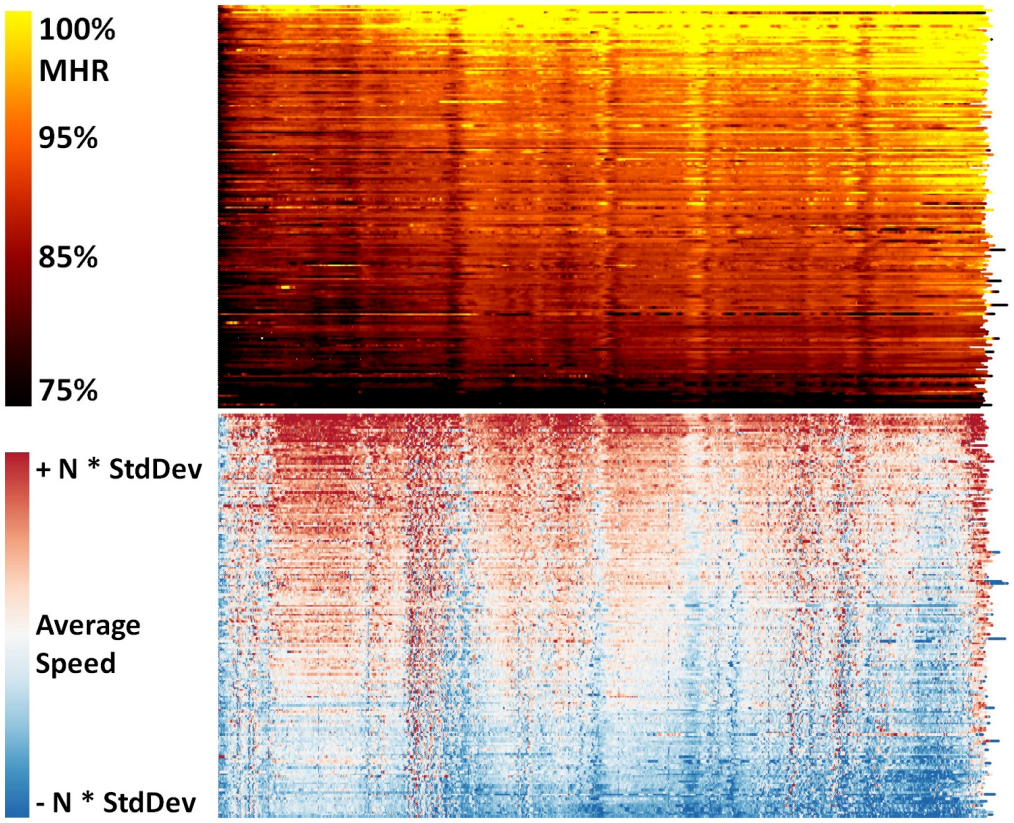


Figure 7.7: **Linear Heatmap** for the heart rate, measured as percentage of maximum heart rate, (top) and speed (bottom). (Average speed = 10 Km/h; Std Dev = 1.4 Km/h; N = 3)

level in the race, as well as in a given part of the course (now in distance, not in time as in the line graph heatmap) (Q1); find the section of the course that demands more effort (Q2); identify common patterns of effort variation among the runners and look for an explanation to them in the slope and speed variation (Q3); and use the linear heatmap with the speed color mapping to see the running strategy of the race, i.e. how runners change their speed during the race (Q4) (see Figure 7.7).

7.2.3.3 Visualization Design 3: Augmented Track View

The third component view focuses on the geo-spatial data of the dataset. The main purpose is to view and annotate the race course to understand the patterns and events

found in the other views. This design is basically a sketch of the race course with the trackpoints representing the runner's state along the competition (see Figure 7.8). The trackpoint position (in the visualization) is based on the trackpoint latitude and longitude to form the shape of the race course. This course sketch is created using the geo-spatial data from a single chosen runner, which is usually very similar among runners. Inclination and even altitude itself have great influence in the runners performance and are usually the main cause for the variation in effort patterns shown in the other visualizations. To represent altitude we encode it using shadows. We draw the full course three times, with different offsets in some direction to simulate shadowing. The offset is proportional to the trackpoint altitude at that geo-spatial location. The three layered layouts with the altitude-based offsetting makes the course looks like a surface extruded from the plane and under a directional light source. A problem that may arise from the directional offset is that only the top layer will be visible when the course direction is too close to the light direction. To overcome this, the altitude is also mapped to the course width as an additional hint, making the higher sections of the course wider, as it would be if a bird eye view of a 3D representation of the course was used (figure 7.8). Finally, the runners trackpoints are rendered, on top of the extruded course, with the same motion trail animation of the line graph heatmap.

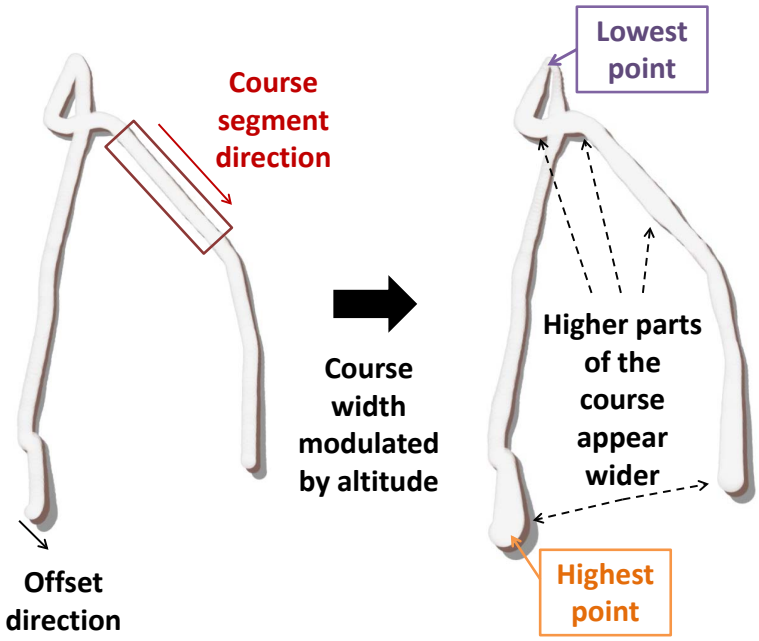


Figure 7.8: **Extruded course with proportional altitude**: In cases where regions of the course match the offset direction, shadows are occluded by the top layer itself, hiding the slope information. We create a second representation of the altitude by making the course width relative to the altitude at each point, and higher altitude sections become wider.

7.2.3.4 Correlated Features among Visualization Designs

We used common features among the visualization designs. The goal was to correlate information between designs in such a way that a pattern that is only visible in one design can be exposed in another, as shown below.

Filters can be applied in the linear and line graphs heatmaps to reduce the visual overload to reveal interesting patterns. By choosing a distance interval the user can customize

the visualization to show, in that range, only the trackpoints whose data is between a defined span of values. The range and parameters of each filter can be defined at any moment during the visualization and, since the filtering is computed at runtime, the result is immediately visible. Filters can be dragged along the horizontal axis and overlapped to create more complex filtering schemes. The filters are important to identify behaviors, for example, athletes pushing their HR close to their MHR (see figure 7.9). From this we can identify regions of the course where the behavior begins and ends. Based on that, we can re-define the course to avoid stressing too much the majority of the athletes or an athlete can prepare an activity strategy based on past experiences.

Distance markers are used to correlate patterns found in the different views. They are tools for annotation along the race course, for example, to insert a description label for a section of the race, or to mark a point where there is a an increase in effort level. Figure teaser.pdf shows an example where distance markers are used to correlate events in the linear heatmap effort view to the location in the track view, thus allowing to investigate causes of some of the patterns found in the linear heatmap (Q3).

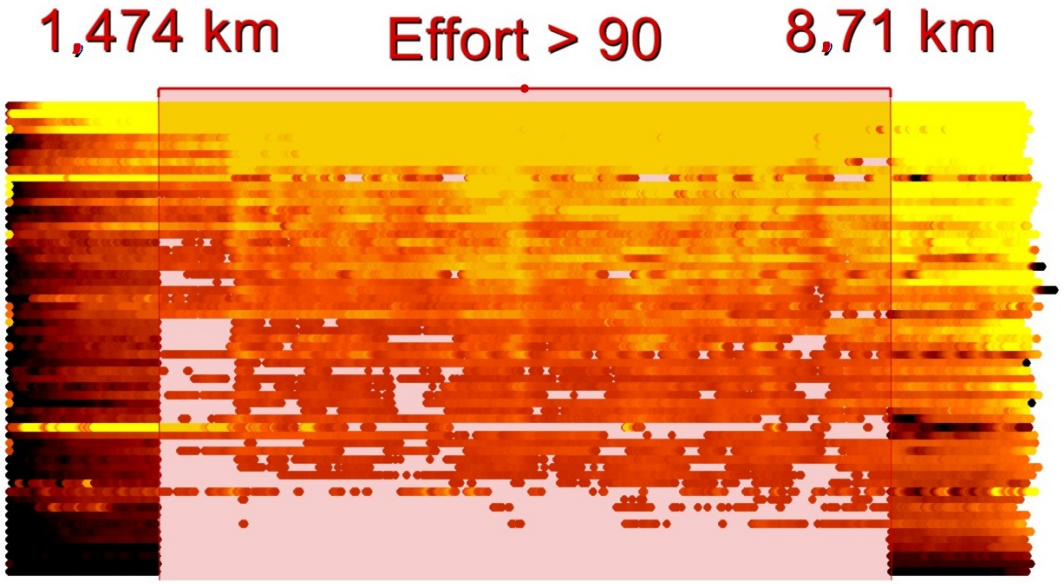


Figure 7.9: **Filter** to remove trackpoints with effort rate below 90% of MHR.

The visualization designs focus on visualizing the dataset as a whole (all runners), however, there are situations in which it is interesting to follow a particular runner or make a more detailed comparison between the performances of two runners. The runner tracker is a tool to provide an instant summary of a runner's state during the race. The main informations that can be visualized are the MHR and the variation of effort rate, speed, and altitude. The properties charts keep a small history of the last values of the variable and inform its extreme values, also, the area chart baseline is defined by the runner's mean value during the race, so the chart appears upside down when the value is below the average. In the altitude chart the base line is always 0, representing the sea level. We opt to show altitude itself, since this way, inclination also becomes evident in the altitude area graph. Runner trackers can be used in all the three views (figure 7.10 shows a runner tracker in the course view) and always point to the last rendered trackpoint of the runner in the animation.

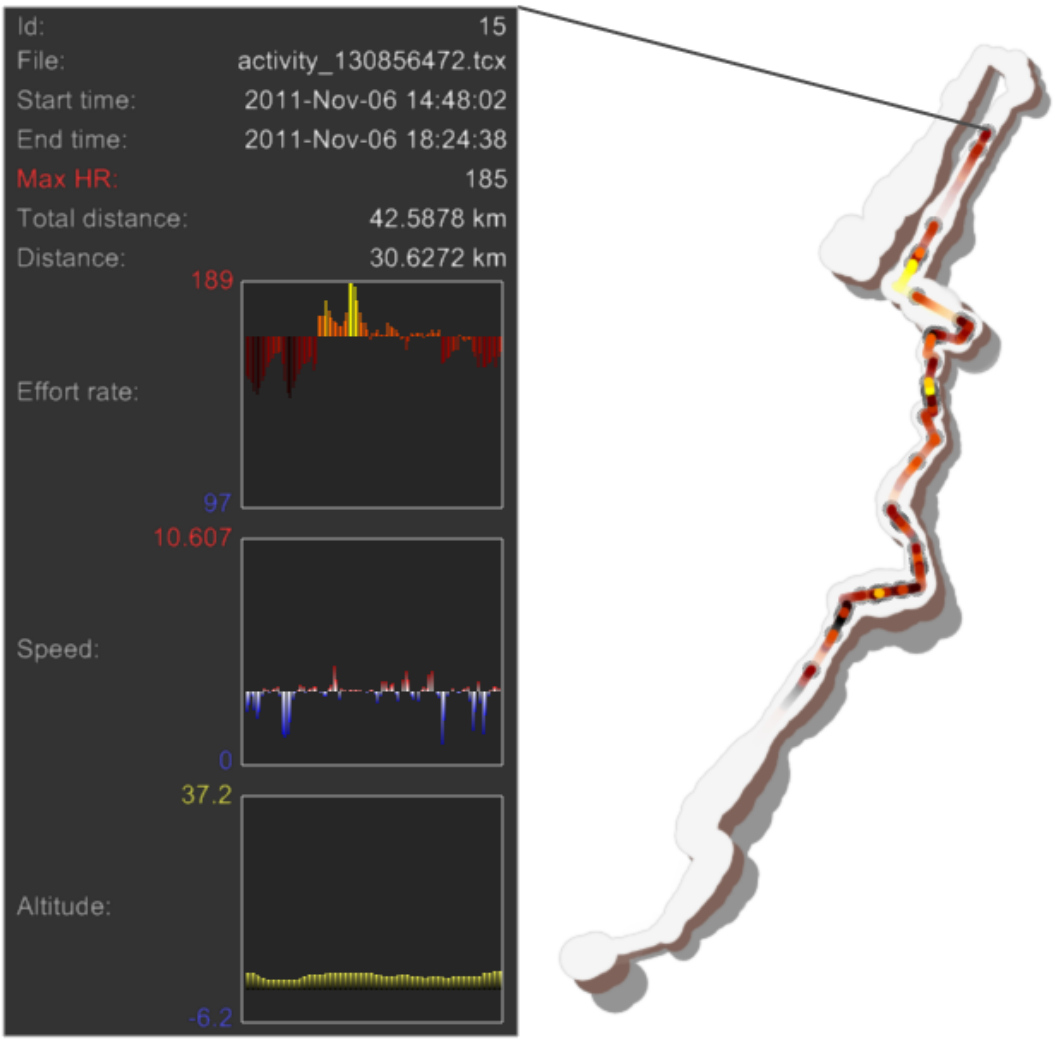


Figure 7.10: **Runner trackers** Tracker following a runner in the course view, displaying runner id, name of the activity TCX file, MHR, among other info.

7.3 Results

The building blocks of all three visualization designs are the activities trackpoints. Trackpoints are samples of the athletes state collected by the monitor device at regular intervals, containing the current values of HR, speed, altitude, longitude, and latitude. Common to the visualization designs is the rendering of consecutive trackpoints using a solid circle (particle). Each trackpoint is rendered only if the current visualization time is already past the trackpoint's, thus creating an animation of how the runners' states changes over time. Figure 7.11 shows one of the proposed design, the Linear Heatmap. The main purpose of this view is to visualize the dataset while avoiding overlap. Each activity (runner) becomes a horizontal line, made of particles positioned from left to right according to how far the runner was from the initial position when the respective trackpoint was registered. The particle color can represent the runner effort,as a percentage of his estimate maximum heart rate (MHR), or its speed related to the average speed of all runners in the race.

In figure 7.12, the Linear Heatmap is used with the other two visualization designs to analyze the same race. The Line Graph view connect the consecutive particles of the

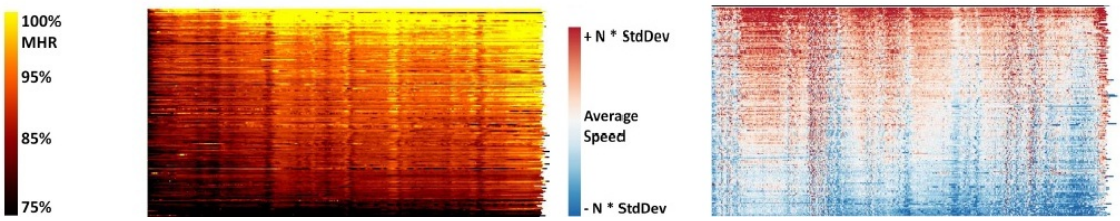


Figure 7.11: **Linear Heatmap.** Linear Heatmap for the heart rate, measured as percentage of maximum heart rate, (left) and speed (right). (Average speed = 10 km/h; Std Dev = 1.4 km/h; N = 3)

same runner creating a line for each runner, that fades with time. The particle horizontal position represents the distance, as in the Linear Heatmap, however, now both the color and the vertical position represent the actual effort level. Indeed, the purpose of this design is to locate runners in the effort zones, which are of major importance for training. The last design, the Course View, uses fading particles, that are positioned according to their trackpoints latitude and longitude values, to show the runners progresses on top of the race's course. The figure also shows a filtering of particles based on their HR and speed values. By combining both filters we visualize only those particles in which the effort was too high when moving slowly, which can mean a steep ascent in the course or a fatigued runner.

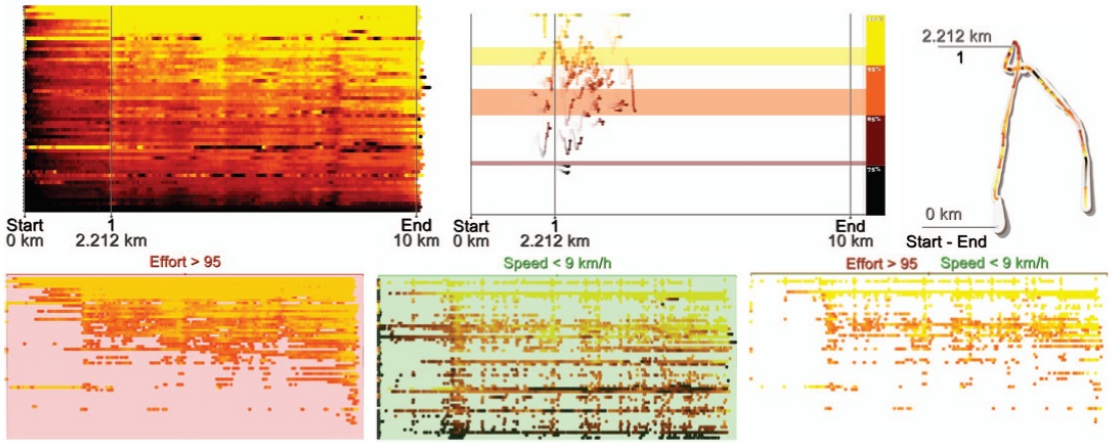


Figure 7.12: **Adidas Summer Run in Sao Paulo.** Top row: one alignment of increase in effort marked in the linear heatmap; the same position is marked in the line graph heatmap, the thickness of the stripes of the effort zones shows that the majority of runners is above 85the track view turns out to be the beginning of a ramp. Bottom row: combination of filters to find trackpoints that indicate a possible risk state.

We downloaded training data from GarminConnect of three different races: 10 km (also called 10K), 15K, and 42K. To allow the comparison of a race in different years, we obtained data for the 15K and 42K from two consecutive years. We presented an analysis of those races as use cases to validate our set of visualization. In figure 7.13 we visualize the course with the profiles of HR and speed variation of the São Silvestre race, a 15K race that takes place every year in São Paulo, Brazil, on December 31st. We downloaded two datasets of this race (2010 and 2011). Since the race course changed in 2011, it is possible to verify different patterns when comparing the visualization for

the years of 2010 and 2011. The distance markers in this figure are used to correlate changes in the effort and speed in the linear heatmap to the respective spots in the course view. In comparison to the 10K race (figure 7.12), we observe that high effort readings decrease, since runners have to sustain effort for a longer period, and therefore run at a lower intensity level. This use case illustrates the ability to compare different races.

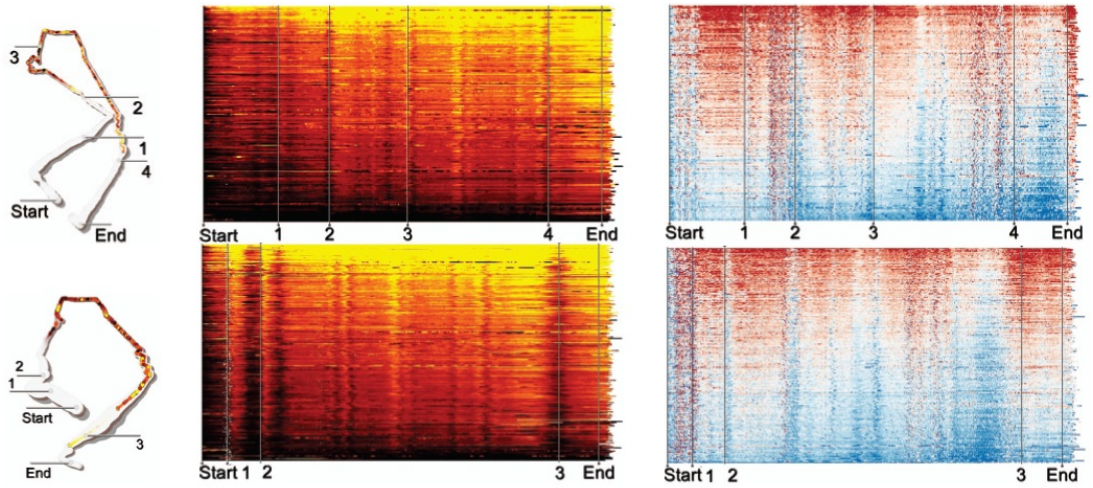


Figure 7.13: **Sao Silvestre Race in Sao Paulo.** Augmented track view and linear heatmaps used to compare the datasets from 2010 (top) and 2011 (bottom). Distance markers are used to point alignment patterns in the linear heatmaps, showing that different courses create very different profiles of effort and speed.

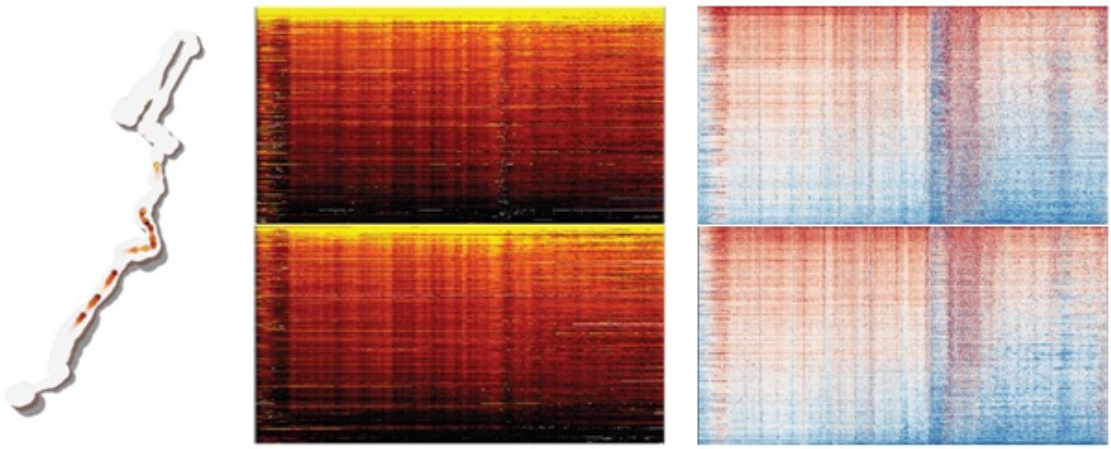


Figure 7.14: **NY Marathon 2010 and 2011.** Augmented track view and linear heatmaps used to compare the datasets from 2010 (top) and 2011 (bottom). With no change of course, the profiles remain the same.

Back to the desiderata, we summarize below how our designs helped answering these questions:

Q1: The predominant effort level of a race tells how hard the runners are performing. Effort levels change during the race, and its is important to identify where and why such changes occur. Using the line graph heatmap, we keep track of the number of runners in each effort zone at any time, independent of their location in the track. This information

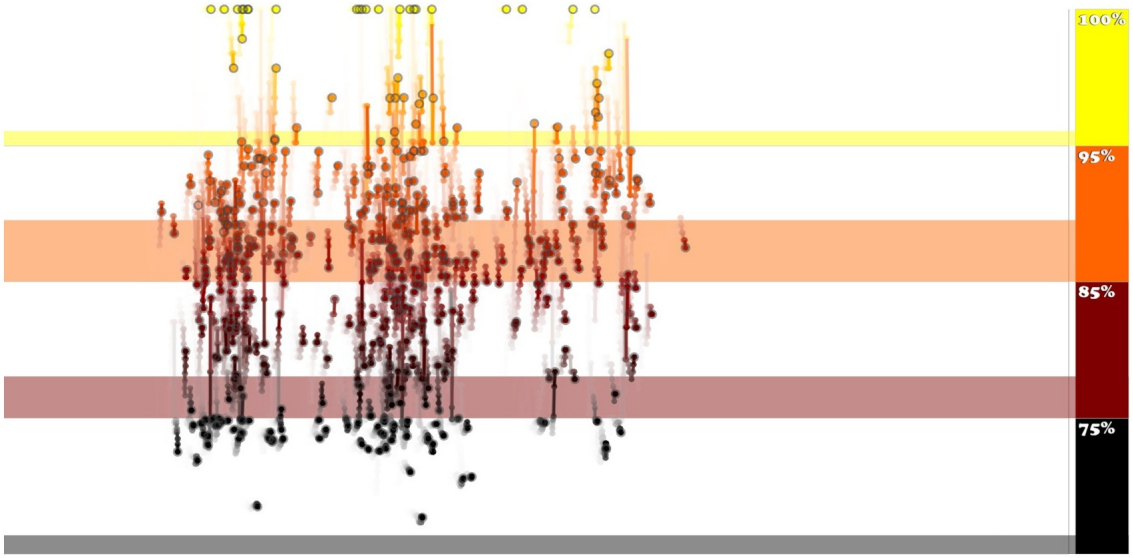


Figure 7.15: **NY Marathon Pace Groups** Reducing the trail length in the Line Graph Heatmap reveal the subdivision of runners in the three pace groups.

is conveyed in a stripe with its thickness proportional to this number. Alternatively, the linear heatmap allows to observe changes in effort level throughout race length by looking at the color changes along the horizontal axis. The case studies we used allowed us to observe such patterns in races of different distances and also verify the difference in overall effort between races of different lengths. For example, comparing the 10K race against the marathons we observe that runners tend to sustain higher effort levels in the shorter race.

Q2: The most effort demanding part of a race can be identified in the linear effort heatmap as the vertical section where the particles have the brightest overall color. The use cases revealed that they are usually sections of long or strong increase in altitude, or the race final dash, where competition increases toward a better classification.

Q3: With the linear heatmap we can observe the overall behavior as noisy vertical stripes created by strong changes in color. The distance where such changes occur can be marked also in the race track and appears in both the effort and speed linear heatmaps, so we can verify if a increase in effort is caused by increase in running speed or the start of an uphill section of the track.

Q4: The speed linear heatmap shows the running strategy of the race. It shows alignments in color change, like the linear effort heatmap, representing increases and decreases in the overall speed and helps identifying parts of the track with high and low overall speed.

Q5: Identify people running at risky levels is a major concern. It can help in the choice of safer race tracks and running strategies, as well as in the creation of different pace groups. With filters in the linear heatmap, we could see people running slow but at high effort level and identify the section where such behavior begins. Further analysis of the same section in the speed heatmap and track view tells if such speed is below the predominant value for the section and if the cause is an increase in elevation. Speed below the local average, especially in sections of constant altitude is a indicator of fatigue.

Q6: Each of the three designs is useful to compare different races. The line graph heatmap shows the overall effort variation in time, the linear heatmap shows the effort and speed outcomes, and the track view display the race topography. This allows to compare

different races, but also to find differences in different instances of the same race. In the case studies we compared races in consecutive years. As the course of the São Silvestre race changed from 2010 to 2011, the effort and speed profiles are very different, while in the the New York marathons were basically the same .

Although we focused on running activities and the analysis of different athletes on a single race, our approach has potential to be extended to different activities and dataset types as well. Comparing different groups of athletes and analysing the history of activities of a single person are examples of different datasets that were suggested. Also, other variables already available by the monitors, cadence and calories for instance, can enhance the analysis. The same can be done for data that can be derived, like fatigue, energetic efficiency, and estimates on the risk of joint damage due to step impact along different sections of the course. Finally, information like age, weight, height, and VO2max that are usually available when working with groups of runners under the supervision of a coach and subject to physiology labs are example of data that can be analysed.

8 BIKE-SHARING

In the last chapter we used the representation of stack of compressed series to gain insight about running races by putting together series of measurements of hundreds of competing runners. In the case study we in this chapter, we made a set of modificiations to this stack design to conform to a different instance of spatio-temporal data. Now, the datasets will represent the circulation of resources in a system over time, as means of exchanges of resources between nodes in a graph and changes of resource availability in each node. In our case study, such system is represented by public bike-sharing programs.

Public bike-sharing systems are increasing in popularity in the last years with many instances already running in biggest cities around the world. The concept is a vehicle sharing system with stations located at several spots around the city with a number of bikes available. Commuters who subscribe to the program can take a bike out of any station, ride it for a limited amount of time and then return it to any station. In a one-way sharing system, there is no need to return the vehicle to the same origin station. In 2013, Larsen [50] reported an exponential growth in the last 13 years on the amount of bike-sharing resources worldwide, with more than 675 bike-sharing systems distributed over more than 500 cities in 49 countries. All bike-sharing programs in the United States, as of May of 2013, are listed in a subsequent work [51], along with plans of increasing service coverage for existing systems and deployment of new ones, including New York's Citi Bike program. The popularity of such system is explained by its sustainability properties of requiring less space, causing less pollution and being cheaper than traditional transportation modes. According to [51], membership in Citi Bike costs close to $100 per year, which is still less than the monthly subway pass. Also, users in DC found annual membership saved them close to $800 in transportation costs, being far cheaper than the cost estimate for the average person to own a car and drive it 10,000 miles a year with depreciation and gasoline expenditures included. Bike-systems are a solution to improve city life by reducing the workload of public transportation network, thus reducing the problems of traffic and pollution while providing a more reasonable alternative for commuters and a healthier lifestyle.

Different from bus and subway systems, where commuter circulation is dependent of the system timetable, bike sharing systems have no imposed regularity in the circulation as the users can ride the bikes at any time and through any path. One problem that rises from this usage scheme is that the operational staff has little control on the distribution of resources (bikes) as commuters are constantly moving them around. This control is of importance to ensure that the stations do not get full or empty so that users can get a bike from any station and also leave a bike in any station, whenever they want. Station rebalancing is used to avoid stations to become either full or empty. Trucks are used to move bikes from different locations, which raises questions on how to choose the best

route that minimize expenses such as gas consumption and time. In addition, trucks are subject to traffic conditions, and some popular stations might need to be rebalanced more often than others.

Citi Bike was deployed in New York City in May 2013 and is the largest bike sharing system in the United States, officially serving 6,000 bikes through 330 stations with a total of more than 11,000 docks [54]. Stations are distributed over the north part of Brooklyn and in Manhattan from Battery Park to the south end of the Central Park. The program has now more than 100,000 users and an average number of 36,000 trips per day (with good weather). During the early months of operation, the system faced several complaints of malfunction of both software and equipment [29]. Users also complained about the unreliability of the software that reports the status of the stations (available at http://www.citibikenyc.com/stations) and the BBSS problem put the system in "perpetual race against its riders" [28]. Different from some programs where rebalancing efforts is done during night time, when the usage frequency is minimal (or there is no service at all), Citi Bike rebalancing operations are performed during daytime in order to handle the intense commuting behavior expect from cities like New York. Trucks take on average 45 minutes to load or unload bikes, and for some stations, by the time the truck is leaving, the station is already with bad balance. Such problems are expected from the adjustment period shortly after deployment of a bike-sharing instance. We believe profiling data from this initial stage can be useful to understand how the population adhere to this new transportation mode, knowledge that can be applied when deploying other bike-sharing instances. Also, as the program popularity increases, the usage data becomes a strong indicator of circulation habits in a city, which can help to better understand city life dynamics, improve the solutions for stations rebalancing by applying instance-specific information, plan upgrades in the infrastructure and even help commuters to make better use of the program. In this work, we focus in the problem of exploring bike-sharing usage data to understand its underlying dynamics.

8.1 Related Works

There are several works with a focus in bike sharing, and they are particularly focused in understanding its behavior. Based on observation or simulation some works present modifications to optimize two main points: availability and flow. Other works present analytic tools to the system oversee which in turns can optimize the system. We categorize previous works according to three categories as follow. Our work fit in the Visual Analytics category.

Balancing Bike-Sharing Systems This section present computational approaches concerning optimal routes to visit unbalanced stations performing rebalancing operations. The work of Rainer-Harbach et al. [68] focus on the redistribution of bicycles to avoid rental stations to run entirely empty or full. They propose a neighborhood search which generate candidate routes for vehicles to visit unbalanced stations and derive the operations (bike load or unload) to be performed on the way. Tests performed on instances based on real data are used to evaluate the best among three approaches to define the operations performed on each visit. Gunther et al. [67] introduces MF-CG, a new method based on maximum-flow on graphs, to calculate optimal loading instructions for given routes in the balancing bike sharing systems (BBSS) problem. Urli et al. [74] address the problem of instance generation for benchmarking proposed approaches to the optimization problem of BBSS. They describe a process to generate such BBSS problems

input instances based on data from Citibike NYC bike sharing system. Schuijbroek et al. [71] propose a cluster-first route-second heuristic to solve both the problem of determining service level requirements at each station and finding a near-optimal vehicle route to rebalance the inventory. Results are provided using real-world data from both Hubway and Capital Bikeshare (Boston and Washington bike sharing systems, respectively). Guenther et al. [36] focus on the forecasting of future bicycle migration trends. Using historical information about individual trips they introduce and compare the performance of two predictive models (mean-field analysis and multiple linear regression) to improve the arrival forecast for a small group of docking station in the London Barclays Cycle Hire system. The work of Papazek et al. [64] introduces a method which applies a hybrid heuristic to find efficient vehicle routes to the BBSS problem. They provide computational results, with benchmark instances derived from data from the real-world bike share scenario in Vienna, showing their hybrid solution scales better than previous approaches.

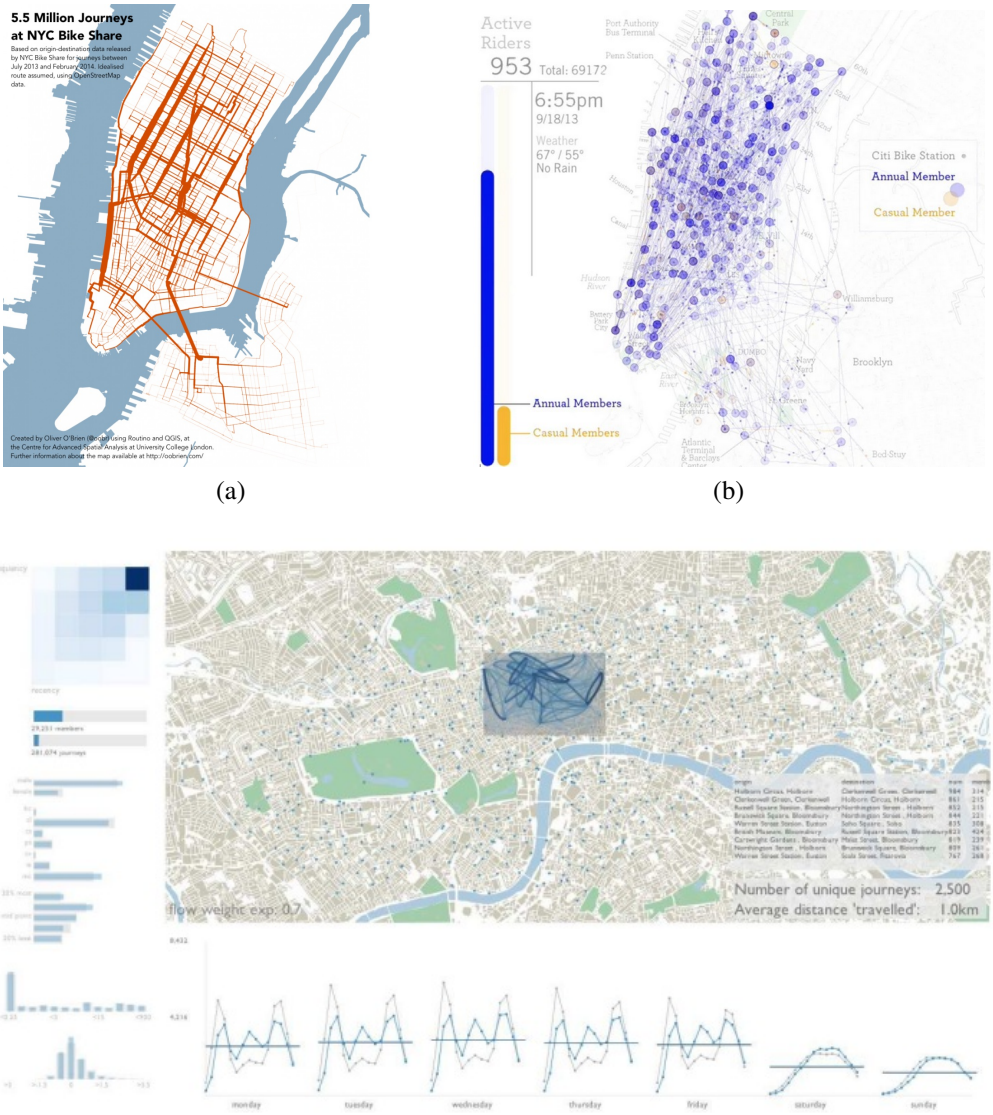
Visual Analytics This section present works that apply visualization and analytics tools to allow a researcher or system oversee to have insights about the bike sharing dynamics. In [59], O'Brien created an online map tool that show stats about a wide range of bike-share systems worldwide along with weather conditions. The World Map in [62] provides a map of the bike-sharing services around the world. Different from [59], this one shows only the program status as operational, in planning or under construction, and deactivated. O'Brien uses in [60] the map introduced before [59] to layout the geographies of the systems of several cities around the world. In [58] trip data from Citi Bike is used to estimate the flow of bike traffic in the streets of New York using OpenStreetMap [2] and the Routino router [3] to find the routes between the stations (see figure 8.1). Flow intensity are represented by the thickness of the lines in the map. We also use trip data to estimate the flow intensity in the streets; however, we use the Google Directions Service [1], code flow as colors and add a number of interaction capabilities aimed to support exploration.

In [77], Wellington partition area covered by the Citi Bike program according to proximity to the nearest station creating a Voronoi diagram in which cell color's map different properties of trips beginning or ending there. Results show the distribution of trip duration, age, gender and user type (casual or annual subscriber). In [78], he use the same approach of [77] but to map number about the most used bike in the system to the date. Ferzoco [27] visualize the Citi Bike's dataset of trips to show how New Yorkers and Tourists circulates among the system in two-day period (see figure 8.1). Kaufman [45] presents the direct correlation between the number of delays in the subway system and the number of Citi Bike trips. And in [44], makes an analysis of the popularity of the Citi Bike stations between genders. The results point women preferred the Brooklyn residential neighborhoods while men were overwhelmingly represented in bustling midtown Manhattan. The author claims women typically attribute reduced cycling numbers to safety among car traffic, which explains why there is lower female participation in the use of stations distributed across some of the most congested parts of Manhattan and Brooklyn. Another work of O'Brien in [61] take a global view of bike-sharing, using data from 38 systems in different continents. A classification of the systems is proposed based on the geographical footprint and day-of-week variations in occupancy rates. The works of Beecham et al. [11] and [12] use visual analytics to look into cycling behaviors in The London Cycle Hire Scheme (LCHS). Using data from the system's customer database and a complete set of journeys records commuters are classified according to how they use the system, and a tool is introduced with coordinated views to support the

query of journeys. Chiraphadhanakul et al. [22] is interested in the impact of vehicle sharing schemes into public transportation. Their model of the public transportation system is defined as a network, and them the vehicle sharing system is added by augmenting the original network with nodes representing available sharing options. The proposal is evaluated with Boston's transit network as a use case, with data from the Massachusetts Bay Transportation Authority that includes subway, bus, commuter rail and boat, and the Hubway bike sharing system. Another contribution goes towards the devising of vehicle sharing systems based on the identification of optimal location for sharing stations and minimization of routes' travel times over the integrated network. Zaltz et al. [83] employ visualization, descriptive statistics and spatial and network analysis tools with trips data to explore bike sharing system usage in five different cities. The goal is to find similarities between the different systems, detect communities in the derived network to identify local pockets of use, and gain insight above and beyond proximity/popularity correlations.

The work of Wood et al.[81] makes intense use of visualization to gain insight of the dynamics of London's Bicycle Hire Scheme (see figure 8.1). They present three different views: one based on edges over London's map to show bike trips while fully preserving spatial context, and two other views, based on a restructuring scheme to avoid cluttering with minimal loss of spatial context, to show trips and station balance state. In a more recent work Wood et al. [80] uses the design study on bike-sharing to reason on a multi-channel approach for data visualization design. Ferreira et al. [26] also work with commuting data to better understand city life dynamics, but using taxi trips. Wang et al. [76] fit trajectories given by taxi GPS to the road network in order to derive traffic information and visualize how traffic jams propagate in the city. The work of Guo et al. [37] also present a design with coordinated views to explore spatiotemporal datasets with a reorderable matrix representation of time series connected to a map. In our work, we add to this design a novel partial reordering scheme to assist the identification of interesting elements.

Prediction, Planning and Impact Analysis The following works present statistical tools to allow system dynamics insights with minimal use of visualizations and are concerned about the impact of the vehicle sharing schemes in the city life. The work of Froehlich et al. [33] presents an analysis using data from Bicing (Barcelona's Shared Bicycling Program) to gain an understanding of human behavior and city dynamics, like ours; however, they use pure analytics tools to analyse 1.5 months of data. We make extensive use of visualization and interaction to support the exploration of a 10 month long dataset. Another work of Froehlich et al. [34] extends [33] with a comparison of experimental results from four predictive models of near-term station usage. Furthermore, a analysis of the impact of factors such as time of day and station activity in the prediction capabilities of the algorithms is presented, showing how simple predictive models can be employed to predict station status changes with an average error of only two bicycles and can classify station state (full, empty, or in-between) with a accuracy of 80% up to two hours into the future. Borgnat et at. [19] presents a study relying on statistical modeling and data mining to model the evolution of the dynamics of movement within the VÃ©lo'v BBS system and understand the flow of bikes in the city of Lyon. They use both station balances and bike trips data. Dill et al. [24] used GPS data, from a sample of 164 adults in Portland riding their bicycles, to address a number of questions about bicycling behavior with a focus on travel time and route choices, like: How often, why, when, and where do cyclists ride? How does this vary based upon rider characteristics? How do cyclistsâ routes differ from the shortest network distance? What factors influence cyclistsâ route



(a) (b)

(c)

Figure 8.1: [Visualization bike-sharing trips] (a) O'Brien [58] represented the estimated flow of trips in the streets of New York using the origins and destinations of the trips from Citi Bike Program and ideal routes suggested by Routino [3] with OpenStreetMap [2]. (b) Ferzoco [27] animated two days of trips in New York's Citi Bike Program, to compare the behavior of New Yorkers and Tourists.(c) Wood et al.[81] designed a system to explore a dataset of trips taken in the London's Bicycle Hire Scheme

Property	Value
Timestamp	2014-07-22T13:30:05.196Z
Number	72
Address	W 52 St & 11 Ave
Latitude	40767272
Longitude	-73993928
Bikes	0
Free Slots	35

Table 8.1: Station State Data Sample

Property	Value
Duration (seconds)	1547
Start Time	2013-07-01 00:00:02
Stop Time	2013-07-01 00:25:49
Start Station Number	388
End Station Number	459
Bike ID	19816
User Type	Annual Subscriber
User Gender	2 (female)
User Year of Birth	1980

Table 8.2: Trip Data Sample

choice decisions? How do personal attributes influence these decisions? What is the difference in travel time between bicycling and driving? Different from their work, the trip dataset we use does not have information about the exact route taken in each trip, only the origin and destination stations. Ogilvie et al. [63] focus on bike sharing system users in London. Using user registration data, they examine inequalities in uptake and usage levels by explanatory variables including gender, small-area income-deprivation and local cycling prevalence. In the work of Lathia et al.[52] an extensive analysis is presented by comparing data collected before and after a change of policy in the London Barclays Cycle Hire scheme. The new policy allows the casual use of the system, so anyone in possession of a debit or credit card could gain access, as opposed to the previous one when users were required to apply for a key to make use of the resources. The work shows how this change relates to a variety of effects observed around the city. [82] model the impact of the bicycle sharing system in London on the health of its users, using stochastic simulation. Results show the program has positive health impacts overall, but with clearer benefits for men than for women and for older users rather than for the younger ones. [66] present an extensive study of relevant information about more than 50 schemes in Europe with the objective of increasing the opportunities for bike sharing to be used as an instrument to foster clean and sustainable transport mainly in urban areas. It also includes guidance and recommendations for planning, implementation and optimisation, along with studies by different countries.

8.2 Materials and Method

8.2.1 Data

The actual state of all active stations in the Citi Bike system can be queried anytime by fetching the JSON feed[1]. The file provides the actual balance of every station of the system and is updated every time the balance of a station changes. The feed consists of a list of state entries, one for each station in the system. Each entry has the station id, name (its address), amount of bikes available, amount of free parking slots, latitude, longitude, and time stamp defining the moment of last change (see Table 8.1). We have been tracking changes in the first JSON feed, at an interval of 30 seconds, since June of 2013 and storing them in a Postgres database, giving us a total of more than ten million updates about the state of 330 stations over more than 240 days. Trip data is also provided at Cibi Bike website, but in files with tables of monthly history of trips (see Table 8.2).

The event-based nature of the state changes (an event being a commuter parking bike or taking one of the station), creates a time series of irregular sampling interval. We define S_N as the time series representing the usage activity for one station in a given day where there were N events, with S_n^B being the amount of available bikes, S_n^F the amount of free slots and $S_n^C = S_n^B + S_n^F$ the capacity of the station after event n, t_n being the time stamp of the event n with $t_n - t_{n+1}$ not constant. Furthermore, we define $S_n^O = S_n^B - S_{n-1}^B$ as the operation performed at event n, the number of bike arrivals $S_n^{Ba} = S_n^O$ if $S_n^O > 0$ ($S_n^{Ba} = 0$ otherwise), the number of bike taken out $S_n^{Bo} = abs(S_n^O)$ if $S_n^O < 0$ ($S_n^{Bo} = 0$ otherwise), and the station balance $S_n^L = S_n^B / S_n^C$.

To be able to better work with the time series of daily activities, we apply piecewise aggregate approximation to resample the series into regular intervals of 15 minutes (15 minutes showed to be a satisfactory aggregation interval for the given 24hrs length of the series and the stations' rate of usage). We divide the time span of the series into intervals of 15 minutes, and exploit the fact that for any time stamp t between t_n and t_{n+1}, the station state is the same as the one registered at t_n, to fill the 15 minutes resampling intervals that may happen to have not a single sample S^n. I.e., given a resampling interval $R_m = [t_m, t_{m+1}]$ with $t_{m+1} - t_m = 15$ minutes, we define the set of samples in R_m as $R_m^S = \{S^n | t_m \leq t_n < t_{m+1}\}$, however if $R_m^S = \emptyset$, than R_m^S becomes $R_m^S = \{S^n | n < t_m, n+1 > t_{m+1}\}$. The resampled series R is created by aggregating the samples R_m^S in the intervals R_m using the arithmetic mean of each variable separately. I.e., for available bikes, $R_m^B = \sum S^B / |R_m^S|$ with $S \in R_m^S$. For the resampled series R, we also include the bike arrival frequency $R_m^{Fba} = \sum S^{Ba} / (t_{m+1} - t_m)$, bike take out frequency $R_m^{Fbo} = \sum S^{Bo} / (t_{m+1} - t_m)$ and usage frequency $R_m^F = R_m^{Fba} + R_m^{Fbo}$.

Trip data is also aggregated but we chose a larger interval of 1 hour instead of 15 minutes due to visualization constraints that we address in the next subsection. For a given 1-hour we aggregate every trip that began and/or ended in the mean time, and then derive a whole new set of relevant measures: balance, capacity, in/out difference, number of cyclic trips, number os incoming trips, number of outgoing trips, outage state (empty, full or no outage), number of incoming origins, number of outgoing destinations, number of trips, trips duration and trips distance. Balance and capacity are the same from the dataset of stations' states, but as the average of the 4 15 minutes intervals, that correspond to the given 1 hour span. Trips related to a station fall in one of three categories: outgoing, incoming or cyclic. Outgoing trips leave the respective station to any other one, while incoming stations arrive in this station coming from any other. We call trip cyclic when

[1] http://citibikenyc.com/stations/json

the bike is returned to the same station from which it was taken. For each 1 hour interval in series we store the amount of each kind of trip separately, the three kinds summed up, and also the difference between incoming and outgoing stations. We also keep track of outages based on threshold applyed in the average balance. If the average balance is above 0.9 it is said that the station is in full outage state for that 1 hour span, and empty outage if its balance is bellow 0.1. Number of incomming origins is the the amount of different stations from which there are trips comming from, while outgoing destinations are the stations where the bikes are going to. The trips distance are not given in the original dataset, but we use the distance of the shortest path between the origin-destination pair of stations for incoming and outgoing trips, as given by Google Directions when asked for cycling routes. In the case of cyclic trips, we estimate the distance by multipling the duration by the average cycling speed of 2.7 m/s.

8.2.2 Desiderata

With previous works and the set of problems recurrent in bike-sharing systems in mind, we devised a list of tasks that a proper analysis tool should support by means of visualization and interaction. The resulting desiderata is presented below as requirements to be met by the visualization method to be presented next.

R1 Identify stations that eventually become bottlenecks in the system, frequently getting empty or full of bikes. This information can help designing changes to improve the resilience of the system, providing better service for commuters.

R2 Verify the influence of city life changes and events in the behavior of bike-sharers. As the bike sharing program's popularity increases, its dynamics becomes a relevant cue about changes in the city life routine, such that alterations in both domains can be synced and correlated.

R3 Understand how the distribution of stations roles, into source/provider and sink/receiver, changes through the day. This division of roles is recurrent in bike-sharing systems, being usually a good indicator of commercial and residential areas, and aspect of major importance when designing balancing solutions in BBSS problems.

R4 Compare the dynamics of the system at different periods since its deployment. As Citi Bike was deployed only recently, the city is still adapting to it and vice-versa. Looking into how the usage dynamics changed over its first year may give rise to valuable insight on what to expect when the system expands to cover new areas and for the following years to come. Also, it can show how the cycling behavior changes through the different seasons as the weather changes drastically.

8.2.3 Design

We propose a visualization scheme with interactive capabilities to explore data from bike-sharing systems. Our scheme is composed of the compressed view of time series, now as a timeline matrix, and a map. The matrix purpose is to allow navigation in a long temporal domain while present the series of every station at once. This coordinate view is further enhanced with a range of interactive options and specialized to operate with both series of state changes in the stations and commuting trips.

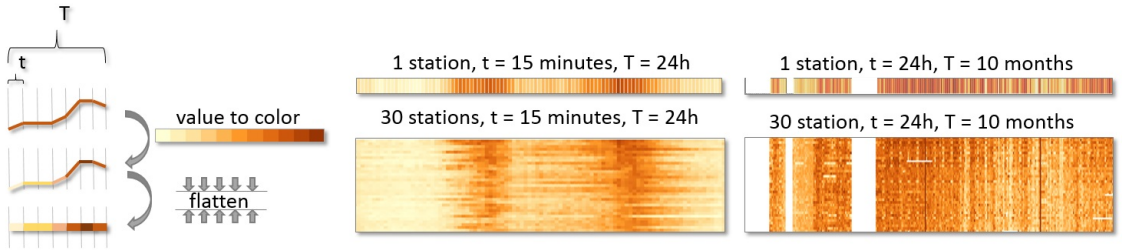


Figure 8.2: [Station's Linear Representation] **Left:** The representation of the stations' states into color-mapped rows. **Middle:** using the single day timeline to view data of one station at that day (top), and also of 30 stations stacked and compressed (bottom). **Right:** using the 10 months timeline to view data of one station (top), and also of 30 stations stacked and compressed (bottom). The vertical white stripes in the 10 months representations are gaps of missing data for part of June, beginning of July and part of September.

8.2.4 Timeline Matrix View

Our dataset provides a time series of state footprint for every station on each day of a span going from the ending of June 2013 to the ending of March 2014, resulting in more than 270 time series per station summing up to more than 8,000 series. Exploiting the typical cyclostationary nature of the bike sharing balance footprint data [19] we visualize the series after an aggregation by days of the week, thus visualizing the expected behavior for a typical week in a given span of days. Aggregation into days of the week is performed over the series outputted after the resampling scheme defined previously in Section 8.2.1. We use 13 different calendar intervals for this aggregation to create 13 datasets of typical week behavior, one for each of the following periods: *June 2013*, *July 2013*, *August 2013*, *September 2013*, *October 2013*, *November 2013*, *December 2013*, *January 2014*, , *February 2014*, *March 2014*, *Summer 2013*, *Fall 2013* and *Winter 2013*.

To visualize the series of every station without overlap, we use the compressed color coded linear representation (see figure 8.2). Each row in the matrix represents the series of states of one station through the timeline. Cells in the same column, map using its fill color the value of the chosen property for each station in the same time interval. We use the same matrix representation to view the dataset in two different temporal resolution: a 24 hours long timeline, with samples of data aggregated over a 15 minutes period (middle of figure 8.2), and a 10 months long one, with samples of aggregated for each day (right of figure 8.2).

Figure 8.3 gives an overview of the components that create the coordinate matrix view. The timeline matrix (C), in figure 8.3, takes the middle portion of the layout as it is the major interaction and informative workhorse of the design. The time period viewed can be selected in the panel (A). Options are: the average series for the different days of the week, weekdays altogether (aggregating from Mondays to Fridays) and weekends (aggregating Saturdays and Sundays), for any month or season; or the series registered during a specific day of the year. The displayed variable, ordering scheme, color ramp and extreme values can be changed anytime in the panel (B). The displayed variable can be any of those from the list of variables derived as explained in Section 8.2.1: *Balance, Bikes Available, Free slots, Frequency, Station capacity, Bikes arrival, Bikes arrival frequency, Bikes takeout* and *Bikes takeout frequency*. *Station capacity* is not always constant as expected. The sum of *Bikes available* and *Free slots* does not always lead to the same value at different

Figure 8.3: [System overview] Our coordinated view supports the exploration of stationsâ data for each day of the week, during different periods in the year, or for any particular day in the dataset (A). Each station is presented as a row in the timeline matrix (C) and in the map (E) as a circle. In the matrix, a cell color codes the value of one of a set of selectable variables at each 15 minutes interval in a day-long timeline. In the map, the circles shows the stations' location with the area being proportional to the total capacity (maximum number of bikes it can store). Rows (stations) in the matrix can be ordered by any of the displayable variables, reduced by different operators (B). The actual order is reflected in ranking lists of Top and Last 10 stations (D), and in the map as each circle's color. This example shows the average frequency of use (number of commuters leaving or taking a bike from the station per minute) of the system on Mondays. The matrix color variation profile shows the recurrent behavior on working days, where frequency peaks around 9 am and 6 pm. Ordering the stations by maximum frequency put 11 Av & 59 St Station at the top with a frequency of at least 7.4 usages per minute. Selecting Central Park South & 6 Avenue station bring up its row in the matrix and present information about it and the surrounding stations (F).

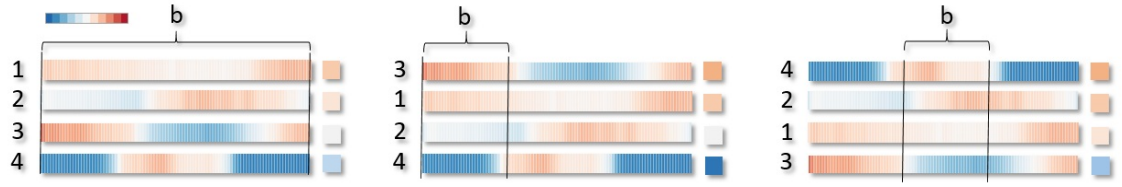


Figure 8.4: [Partial Reordering] **Left:** In the initial ordering, the reduce operator (arithmetic average in this example) is applied over the whole range of values of each row (the resulting value is mapped as the color of the square at the right side of the row). **Middle:** Decreasing the brush extent (*b*) decreases the domain of the reduce operator, thus changing the ordering. Now the stations are ranked by their reduced value during the chosen time period (morning). **Right:** Stations reordered by their average balance now during working hours. Stations *4* and *2* had more bikes than free slots, while *1* was usually evenly balanced, and *3* had a lack of bikes.

timestamps, so, keeping track of its value through time may show unexpected changes that can indicate some problem in a station. Ordering options correspond to the same variables available for color coding in the cells, reduced using one of four reduction operators (*maximum, minimum, mean* and *range*), and time-invariant properties of the stations (*Id, Name, Latitude* and *Longitude*). The map (E) shows each station as a circle whose color is associated to the index of the row of the respective station in the matrix, while the area encodes the capacity of the stations. Pointing one station in the map, highlights its row in the matrix (zoomed in a lens-like fashion), its entry in the ranking lists (if listed in one of them) and shows more info about it in the panel (F). Also, a circular region can be defined in the map to select a group of stations and list then.

When working with the 10-months timeline, each cell represent an aggregation over a longer time period: a whole day. For such long interval a simple average is not very informative so we aggregate the data for each day by six different reduce operators in this case: average, minimum, maximum, range, time of minimum value and time of maximum value. They are the same as the ones available as options to the ordering scheme, with the extras of time of minimum and maximum values. These operators tell the time of the day during which these extreme values were first registered.

8.2.5 Partial Reordering

The rows of the timeline matrix are ordered by their content. Figure 8.2.5 illustrate the ordering scheme. In the left, the rows are ordered according to the average value of the full length of their series. Shorter parts of the timeline can be used to reorder the rows. In the middle part of figure 8.2.5, the rows are ordered using the data from the first 1/3 while on the right the second 1/3 of data is used. This range is defined by the horizontal extent of a brush on top of the matrix. Simultaneously, the brush's vertical extent select the stations that should be shown in the map, so we can inspect only a relevant range of the ranking. From this set of visible stations, we keep a ranking panel with a list of the top 10 stations and other with the last 10 ones. The timeline matrix brush can also be animated as a sliding window, reordering the matrix at each step, to show how the rank of stations changes over time. The ranking history of each station is shown over the matrix as a line graph, anytime the station is selected (figure 8.5).

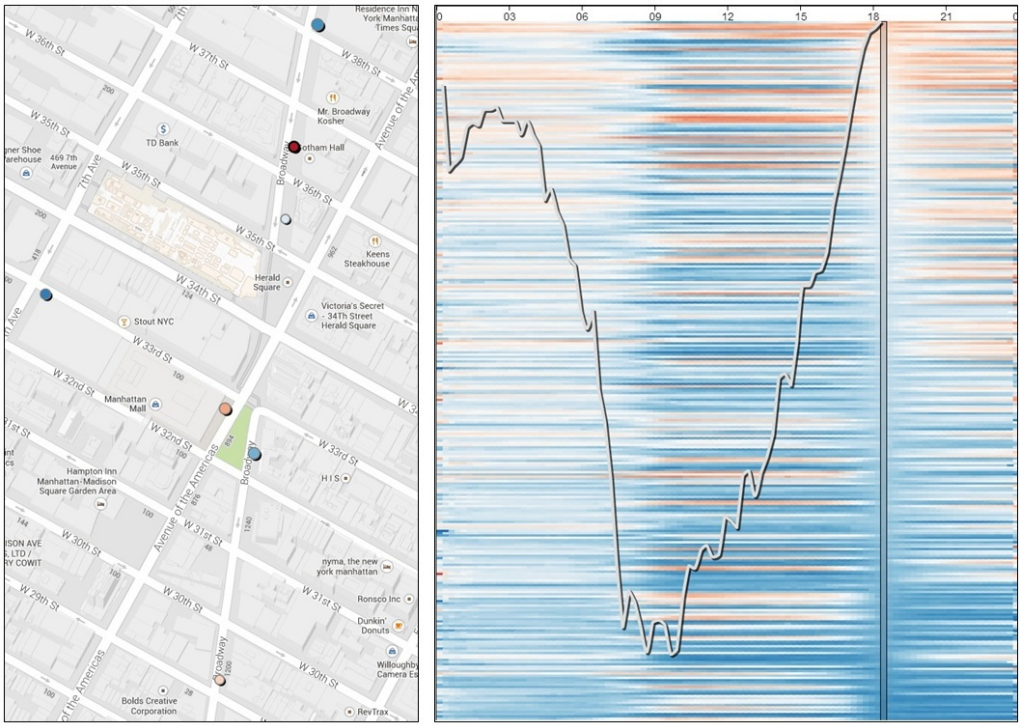


Figure 8.5: [Rank History] The history of ranking positions expected on Mondays for the Broadway and West 37 Street Station (red dot in the map) when ranked by the maximum balance. The ranking history is shown as a line graph with the positions of the station's line in the matrix as it changes when a moving window of 15 minutes from the beginning of the day up to 6 pm. It goes from one of the stations with the lowest balance at 9 am to the fullest one at 6 pm.

8.2.6 Trips Representation

Trips are a richer content to explore, there is a new set of derived variables and there is also the set of trips from which their values came. Now we use the cells of the timeline matrix to preview this set of trips, thus showing how the commuting behavior in each station changes through the day. The set of variables derived from the trips are not represented explicitly in a visual way, but used to rank the rows. In figure 8.6 we show a simplified example of these previews in a part of the timeline for two stations in period between 9 and 12 am. We chose a 1 hour aggregation interval for the trips, instead of the original 15 minutes one used for the stations' states, to create cells with more room for the visual representation of the trips. In each cell, we draw each trip as a semi-transparent line conecting the row station with it origin or destination as it would look in the map in the top-left, with incoming trips as red lines and outgoing as blue ones. Those colors are to relate with the intuition of the balance representation when exploring the stations' states: blue lines leave the selected station, deacreasing its balance towards darker blue in the balance color scale, while red lines are the oposite case. Cyclic trips are a special case. They are semi-transparent station-centered gray circles that cover the area that a bike, in a straight line at an average speed of 2.7 m/s, could have reached given its duration. It is a rough estimative. Also, outage state of the station is given by the cell border color, again red meaning a full outage while blue an empty one. We want the timeline to give a rough view of how many trips have came and gone to each station at each hour, with its spread, so that we can identify an interesting span and select it to have a detailed view of its trips in the map. There we change to a representation using curves, with blue ones drawn in clockwise order to represent outgoing trips, while red anti-clockwise as incoming bikes. Instead of plain colors, the gradients are used to help identify which station is the selected one in the timeline matrix: the one in lighter color extreme (cyan or yellow).

8.2.7 Trips Matrix View

To have an overview of all trips in a given time range we use a matrix representation similar to Guo [37]. Each row in the matrix represents an outgoing station and each column describes an incoming station. Cells colors are mapped to a variable related to the trip between two stations, aggregated at the selected time. We provide an interface to select the current variable displaying for additional studies. The possible values are: number of trips, trips duration in seconds, the balance difference and station capacity difference. The quantity and duration of trips can be used to identify preferred stations among users. Balance difference is calculated to measure the state of the stations where the trip took place. To do this, we take the difference of balance from the incoming station and the outgoing station. A balance difference will have values in the range [-1,1]. A value of 1 indicates that the incoming station is full and the outgoing station is empty (critical case). In the other hand, a value of -1 shows the opposite; a full station at the origin and empty station at the destination (ideal case). A value near zero means the two stations involved in the trip have the same balance value. The capacity difference will help us distinguish trips that occurred from bigger stations to smaller ones, the opposite and trips between stations with equivalent capacity. Figure 8.8 gives an overview of the main trips matrix view components. On the top, the hour slider can be used to filter the hour of day to reveal patterns between trips. The flow matrix takes most of space in the layout as it is the major interaction and informative source of the design. The period shown can be selected in the panel. Options are: aggregated trips for the different days of the week,

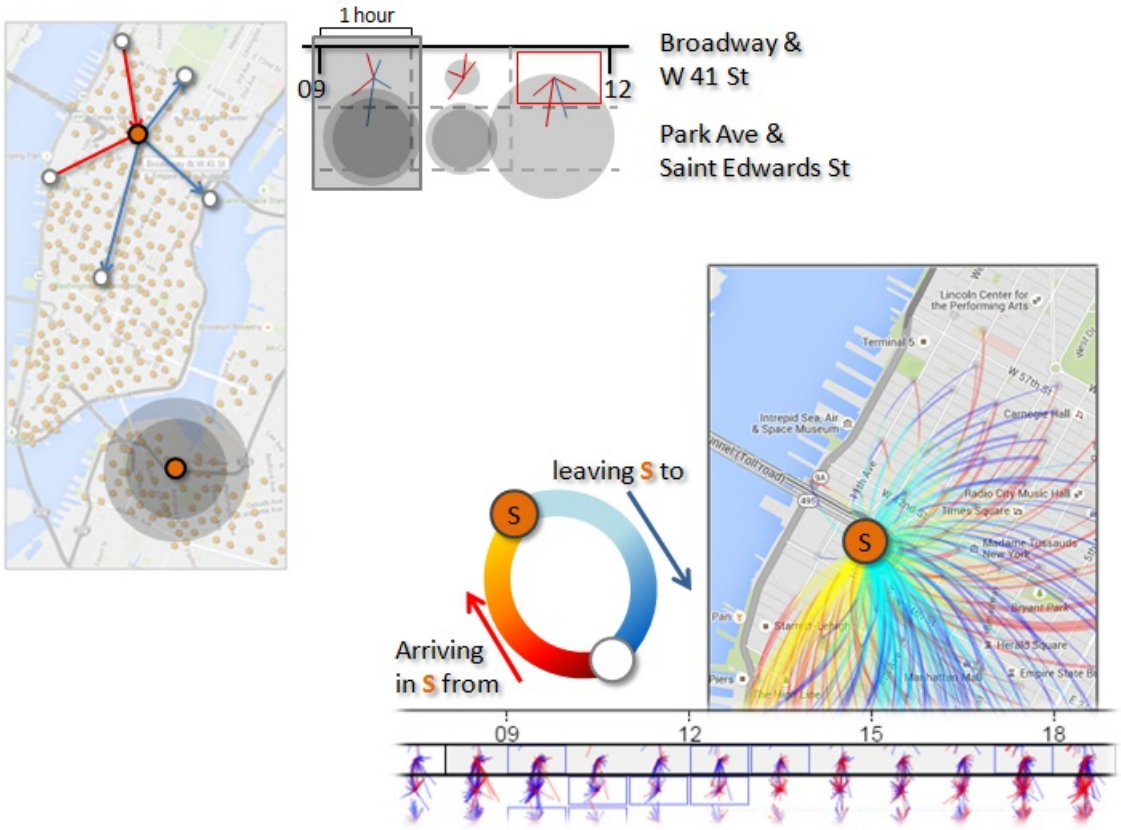
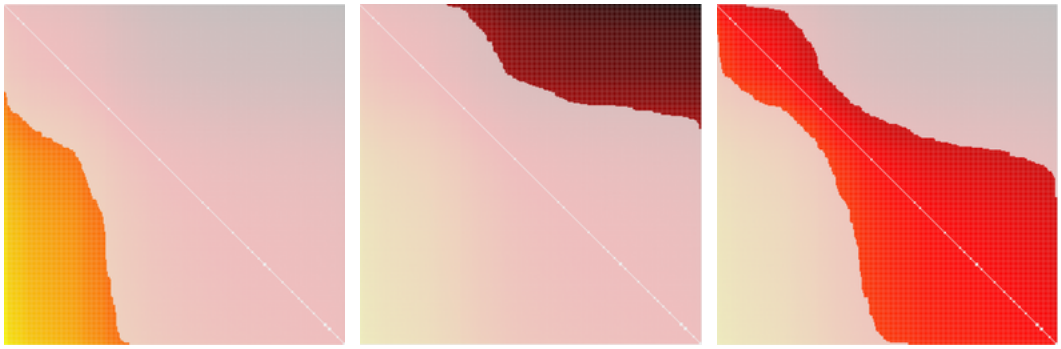


Figure 8.6: [Trips Representation] The diagram shows the representation of trip data in the coordinate scheme of matrix and map. (Top-left) In this simple example, two stations are selected in the timeline matrix in a 1-hour period. Incoming trips are given by red lines while outgoing ones by blue. Each cyclic trip is represented by a semi-transparent circle, with reach as hint of how far a biker could have gone according to the trip's duration. Each trip is also represented in the respective cell of the timeline matrix. The red border in the top cell at 11 am, indicates that the Broadway and W 41 St station spent most of this 1 hour period in a full outage. (Bottom-right) An example with real data. We improve the map representation of the trips by using curves instead of lines, to avoid confusion due to overlap. Trips leaving from the selected station are always drawn in clockwise sense ending in darker blue at the destination, while incoming trips follow anti-clockwise, with dark red in the departing station. The map shows every trip arriving/leaving to/from the top station in the matrix section that took place between 8 am and 6 pm.



(a) Highest 30% values (b) Lowest 30% values (c) Values between 40% - 60%

Figure 8.7: [Trips Matrix auto selection] Automatic selection by value percentage. Trips matrix aggregated over weekdays for September 2013. Displayed variable: balance difference.

weekdays altogether and weekends, for any month or season; or flows registered during a particular day of the year. To find patterns in the users trips, we implemented a row and column reordering depending on a station variable. The options for row/column ordering are: station trips, capacity, latitude, longitude, number of trips, trips duration, balance. The displayed variable, rows and columns ordering, and the color range can be changed anytime using the panel.

For further exploration, we propose two interactive scheme over the trips matrix view: manual selection and auto selection. The manual selection allows users to navigate over the trips matrix and select relevant patterns. The vertical and horizontal brush extent select the stations that serve as start or end points, respectively. To conserve geospatial context along with the trips matrix view, we use a map that shows each station as a circle whose color is associated with their role in the selected trips. Outgoing stations are identified by the blue color, ingoing stations with red color and stations that serve as source and sink are marked with purple color.

To highlight specific values over the trips matrix and examinate its geographic context we introduce the auto selection by parameter scheme. Auto selection is useful for making comparisons between trips matrices based on a percentage of the values range. For example, an auto selection with range 70-100% for the balance difference variable highlights the highest 30% of the range [-1,1] (figure 8.7) , being all the values between 0.4 and 1.

8.3 Results

In this section we describe several analysis we performed over the Citi Bike data. We show a high-level calendar overview analysis, moving to in-depth exploration of specific days and the expectation for days of the week in different periods (months/seasons), to the query of stations and circulation pattern of bike-sharers in NYC.

8.3.1 10 Months Overview

The Calendar View summarizes the whole state-footprint dataset to support an overview analysis as the initial stage of the exploration pipeline. By combining any of derived variables extracted from the station state feeds with a choice of how to reduce it for each

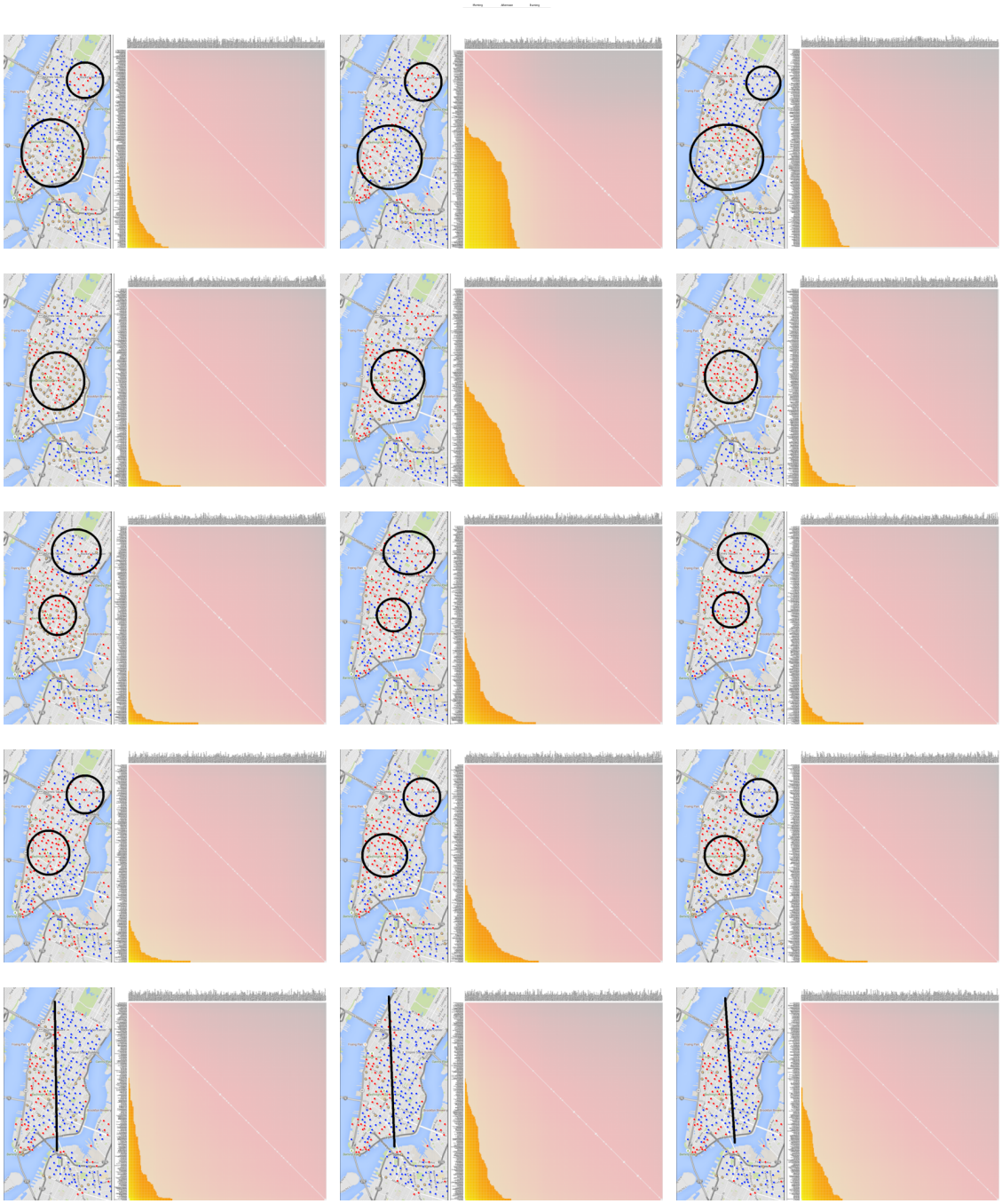
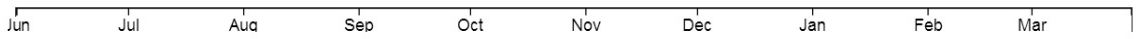
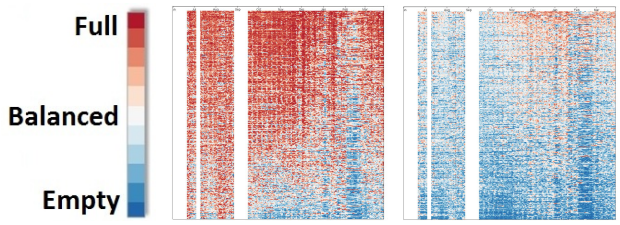


Figure 8.8: [Trips Matrix View] Map View. Stations colored by their role in the selected trips. Blue color highlights outgoing stations and red was used for incoming stations. Trips matrix view ordered by station balance. The displayed variable is the station balance difference. Trips with highest 20% values are selected. Periods analyzed: October 2013 (first row), November 2013 (second row), December 2013 (third row), January 2014 (fourth row) and February 2014 (last row).

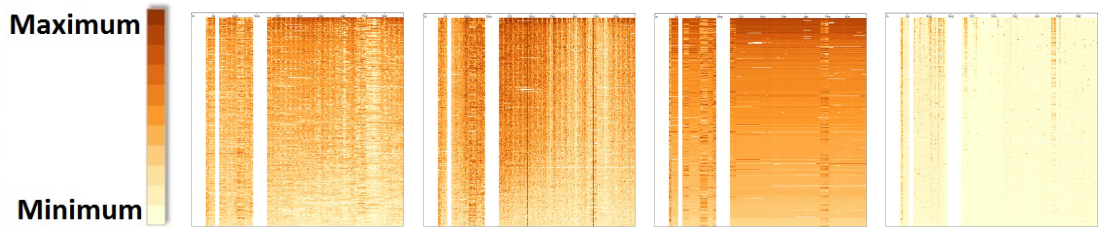
day, it gives different perspectives about how the use of the Citi Bike program developed along the 10 months of data gathered. Figure 8.9 show a set of 9 visual profiles. (a) The color scales used were chosen according to the nature of the variable displayed. (b) and (c) perspectives are based on the balance of the stations, however, the first shows the maximum balance at each day while the other shows the average. In (b) most stations



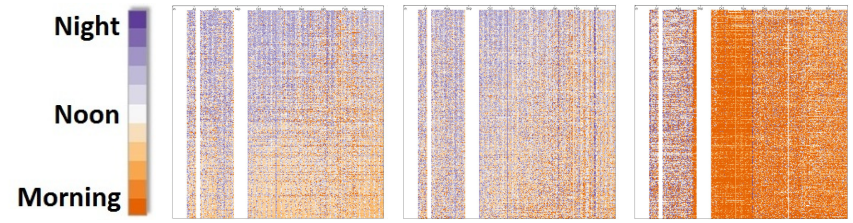
(a) 10-months Timescale Detail



(b) Maximum Balance (c) Average Balance



(d) Max Bikes Available (e) Max Arrival Frequency (f) Average Capacity (g) Range of Capacity



(h) Time of Max Arrival Frequency (i) Time of Max Frequency (j) Time of Minimum Capacity

Figure 8.9: [Calendar View Perspectives] Evolution of the system usage over 10 months given by different variables. **(a)** The scale on the top is a zoomed view of the 10-months timescale in the top of each of the 9 profiles. **(b)** The daily maximum balance showed to be high at the early months (stations usually got close to full at some time everyday) and decreased with the weather temperature. **(c)** On Fall there was a increase in the number of stations that spent most of the day almost full. **(d)** An overall decrease in the maximum amount of bikes available can be see during February for every station. **(e)** Some days show a peak in the maximum number of bikes arrival, specially at the end of January. **(f)** The capacity profile shows a number of periods where the capacity of the stations changed, which is unexpected behavior, as in August and the end January. **(g)** Profiling the daily range of capacity highlight days of odd behavior (high oscillation of capacity among the several stations) as orange columns, like the thick one (meaning several consecutive days) at the end of January, and several thin ones August. **(h)** Using a purple/orange color scale to map time of the day, this profile shows how the maximum arrival frequency began to occur earlier as the temperature decreased. **(i)** The profile of time of maximum frequency show the same overall pattern as **(h)** however it also highlight some days where there was a global maximum frequency at late ours (thin purple columns). **(j)** By visualizing the time of the day when the stations reached minimum capacity, three patterns show up: during July and August the noisy pattern indicates a unbiased distribution of minimum capacity occurrence over the hours; between the end of August and middle November there is a strong bias toward early hours (orange); from the middle of November to the end the bias towards early hours remains, but now weaker than before.

happen to be close to full at some time on a daily basis, especially at early months. This pattern changes as winter approaches and the system use decreases (trend present in most frequency related perspectives like (e)). (c) shows an overall predominance of a good balance (lighter colors) among the stations on early months. This pattern begins to change slightly with the arrival of Fall, with some stations taking more darker reds and blues. The difference between the two profiles exemplifies the flexibility provided by been able to choose different daily reducing operators for the viewed quantity.

In (d), we track the daily maximum of bikes at each station. There is a pattern of lighter color on February meaning that there were less bikes stopped at the docks, which could mean bikes were removed from the system by the program operational staff, or more bikes were circulating through the city. By looking at different profiles we see the first explanation is more likely: (e) show a decrease of the average amount of arrivals during the same period. Viewing the daily maximum of bike arrivals at the stations in (d) reveal a period of strong outlying behavior at the end of January. The column-long increase of displayed value can also be spotted in a few of the other perspectives ((e), (f) and (g)), even if less evident. The strong regularity of this anomaly, as it happens at almost every station at the same time, make operational activity or issues to be the most-likely cause. (h) perspective is also based in the arrival frequency like (e) but instead of viewing the maximum value we color code the time of the day when the maximum value was registered. The purpose of this perspective is to see which stations are destinations by morning (orange) and which ones are more popular at night (purple). (i) also display time of maximum but with total frequency (arrivals and take outs together, instead of first alone). With it, we can find out which stations are more used during early and late hours, with no distinction between roles (if the station is usually an origin or destination at the time). Both perspectives show a trend of stations becoming more popular at early hours during winter (increase of orange color during this period), but only the second one brings up the system-wide anomaly that happened at the ending of February (a dark purple column). (f), (g) and (j) are based on the stations' capacities. The capacity of the stations are supposed to be time-invariant, but these views show a different scenario. In (f) we can see when the average capacity of any given station changed, and also periods when such anomaly happened with high regularity in a system-wide fashion (e.g., at the end of February). Showing the time of minimum capacity of the stations in (j) reveal an intriguing pattern of predominance of late hours until the end of August. There is a sudden and strong change to early hours, changing again in the middle of November, with a single-day-wide purple column followed by a noisier pattern until the end of March. In (g), we use the range of the daily capacities to make easier to spot these capacity anomalies (darker columns).

8.3.2 Detailed Exploration by Period and Date

In the overview analysis with the Calendar View, we identified smooth changes along the 10-month timeline and also a series of anomalies in the different perspectives of the state footprint dataset. Resorting to the 24-hours timeline view, we can narrow the analysis down to a period of greater interest or to the very day in which an unexpected behavior was spotted.

Figure 8.10 present the weekdays and weekends profiles of frequency with monthly resolution. Comparing the 2 rows we see a clear difference between the system use during work days and weekends. Frequency on weekdays has two peaks, one around 9 am when commuters go to work, and a second one at 6 pm when they go back home. There is

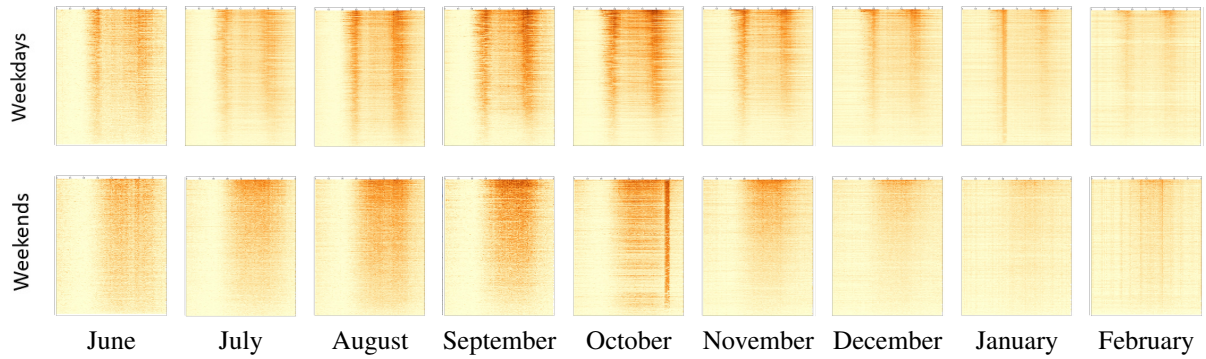
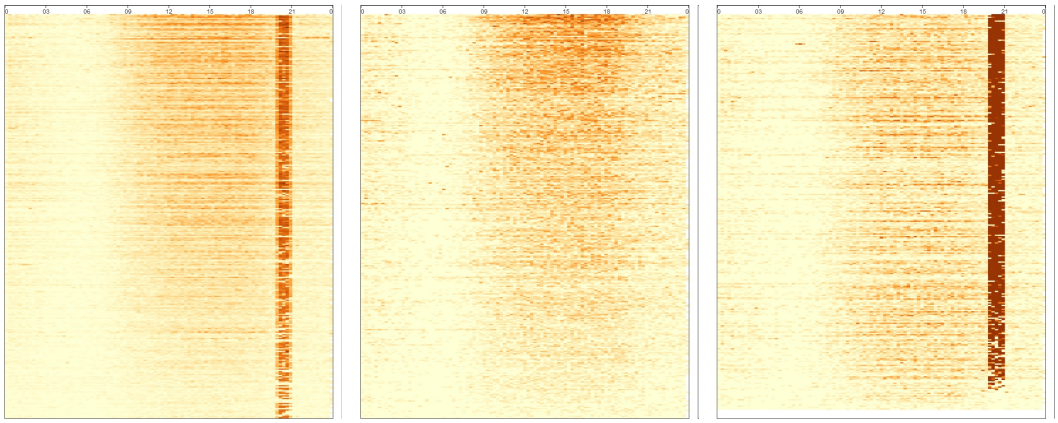


Figure 8.10: [Frequency by month] Profiles of frequency during weekdays (top row) and weekends (bottom) grouped by each of 9 months (March 2014 not shown). An increase in the system usage can be seem as the profiles get darker from June to October and them lighter during Winter months. Also, some profiles show deviations from the expectation: October weekends present a peak of use around 8 pm (more on figure 8.11), the 9 am peak on weekdays of January is more evident than the 8 pm one , and the weekends of February has two short peaks around 11 am and 3 pm.

usage in between, higher then early mornings and late nights, but lower than the rush hours. During weekends, there is a single wider, lower, and smoother peak that begins later than weekdays, at 10 am, and also ending later at 9 pm. Since the same color scale and extreme values was shared between the profiles, we can point that Fall had the most-intense activity (overall darker colors) with a strong decrease during Winter, due to harsh weather. We can now see the contrast increasing from June to October and decreasing again to February. Also, we see the duration of the weekend's peak during Winter was shorter than the other months, going from 11 am to 6 pm, probably in response to the shorter days and longer nights. The last observation is the sharp frequency peak between 9 and 10 pm that is only seen in the profile of weekends during Fall. There is an anomaly at 9 pm of weekends during October. Other anomalies can be spotted as well, like the stronger frequency peak around 9 am of January's weekdays, and the pattern of a series of short usage spikes on weekends of February.

To further inspect these outliers we drill down specific days of the week in a period (e.g.: Thursdays of July) or to a day in the calendar (e.g.: 4th July). Figure 8.11 compares the frequency of Saturdays and Sundays in October, revealing that the anomaly comes from the first. By looking into the profiles of each different Saturday of October, we found out that it only happened at 26th October. With such regularity and intensity, it is very unlikely that it was caused by commuters' activity.

Figure 8.12 exemplifies the use of the partial reordering of rows to mine temporal patterns. On the top, the brush is used to order the rows by the average balance the stations during the mornings of weekdays, on the right, and weekends, on the left. Resulting patterns are clearly different, as is the use of the system in the two periods of the week. On weekdays, we can see how the top stations begin full in the morning, get empty over working hours and then full again at night, with the bottom ones following the very opposite behavior. However, for weekends, no such pattern is visible, as the bikers have a more unpredictable behavior, riding more for leisure than for their working routines. In the lower left, we view the use frequency on weekdays and order the rows by the same property in the interval between 8 and 10 AM (first rush hour). The lower right view uses



October Saturdays | October Sundays | 26th October (Saturday)

Figure 8.11: [Outlying Frequency Peak on October] An unusual peak of frequency around 8 pm shows up on Saturdays of October, but not on Sundays. Further exploration reveals such behavior occurred only in the last Saturday of the month, October 26. The same peak can be noticed on the profile of October (figure 8.10).

the same ordering as the left one, frequency around 9 AM, but now showing the balance of the stations. The outcome is that such ordering also divide the station in the two role groups: working hours destinations on top and origins in the bottom.

8.3.3 Querying Stations

An important design requirement is to be able to identify the stations the become bottlenecks of the system, i.e. usually get full or empty. We complied to the requirement by adding interactive functions to the matrix representation, more specifically the brushing for partial reordering and the stations' ranking lists. Also, with the extensive number of combinations of the set of displayable and ordering variables and the available reduce operator, we can find not only the bottleneck stations, but also those fitting a large number of different criteria. This search can be done for any time interval in both the 10-month overview and 24-hours timelines, and, for the last one, at any specific day in calendar or day of the week of different months and seasons. In the figure 8.13 we show the results of 6 queries of stations by different criteria. In (a) and (b) the brush is defined between 9 am and 5 pm to order the stations by mean balance in that period of the day, and then, by dragging the brush area vertically we highlight in the map the stations according to their balance levels. In (a) the vertical range of the brush is limited to the top rows of the matrix, revealing in the map the stations that were usually full at that time of Summer Wednesdays. (b) shows the ones that were empty, by selecting the bottommost rows. In the maps, there is a clear division between full and empty stations in the lower and upper halves of the area covered by the program respectively. (c) the rows are ordered by range of balance level throughout the whole day, and by selecting last bottommost rows we found those that remained at an almost constant balance level. An interesting task is to find out where are the majority of bikes of the program at a given interval. (d) shows the answer by displaying the number of bikes available in the stations and ordering the rows by the average value, this way, limiting the selection to the top rows, show those with many bikes docked in the map. We can easily view the stations of higher capacity

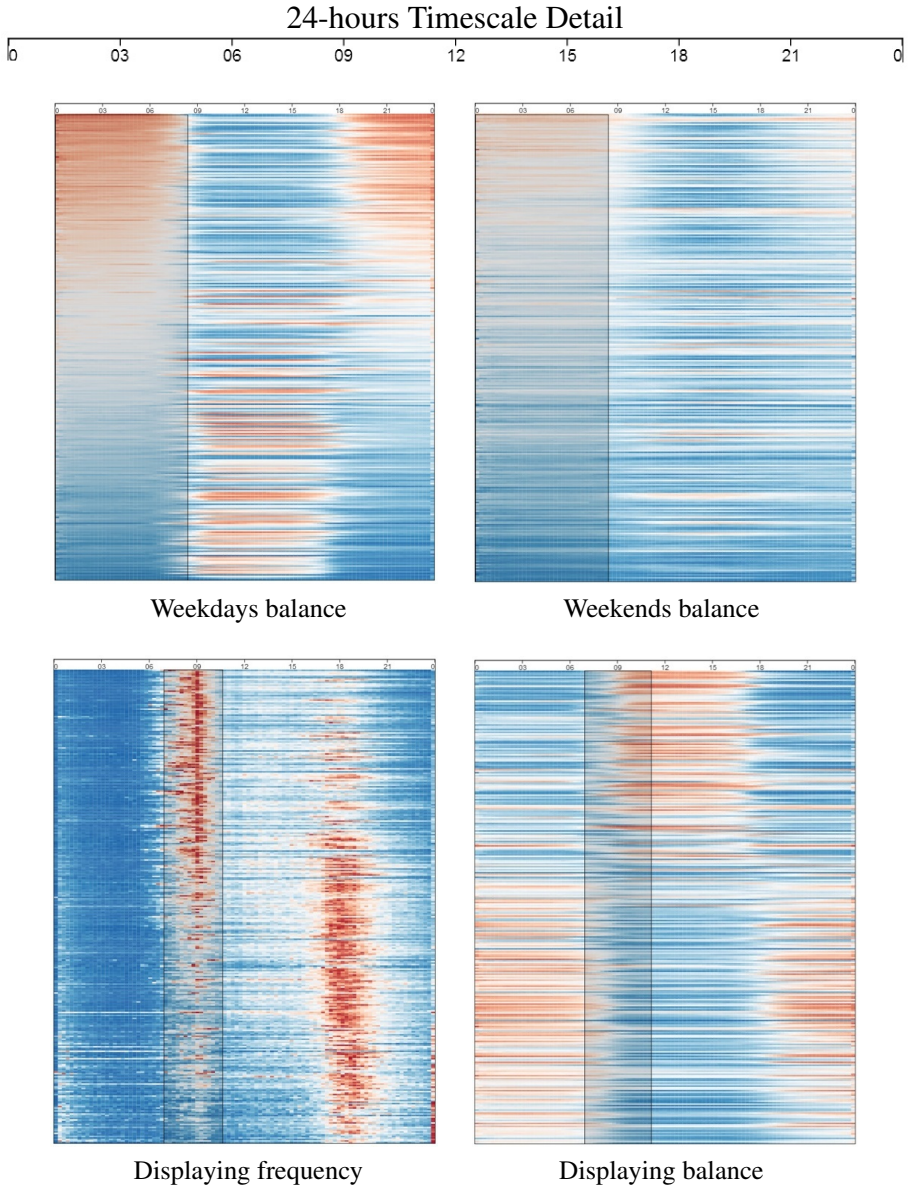
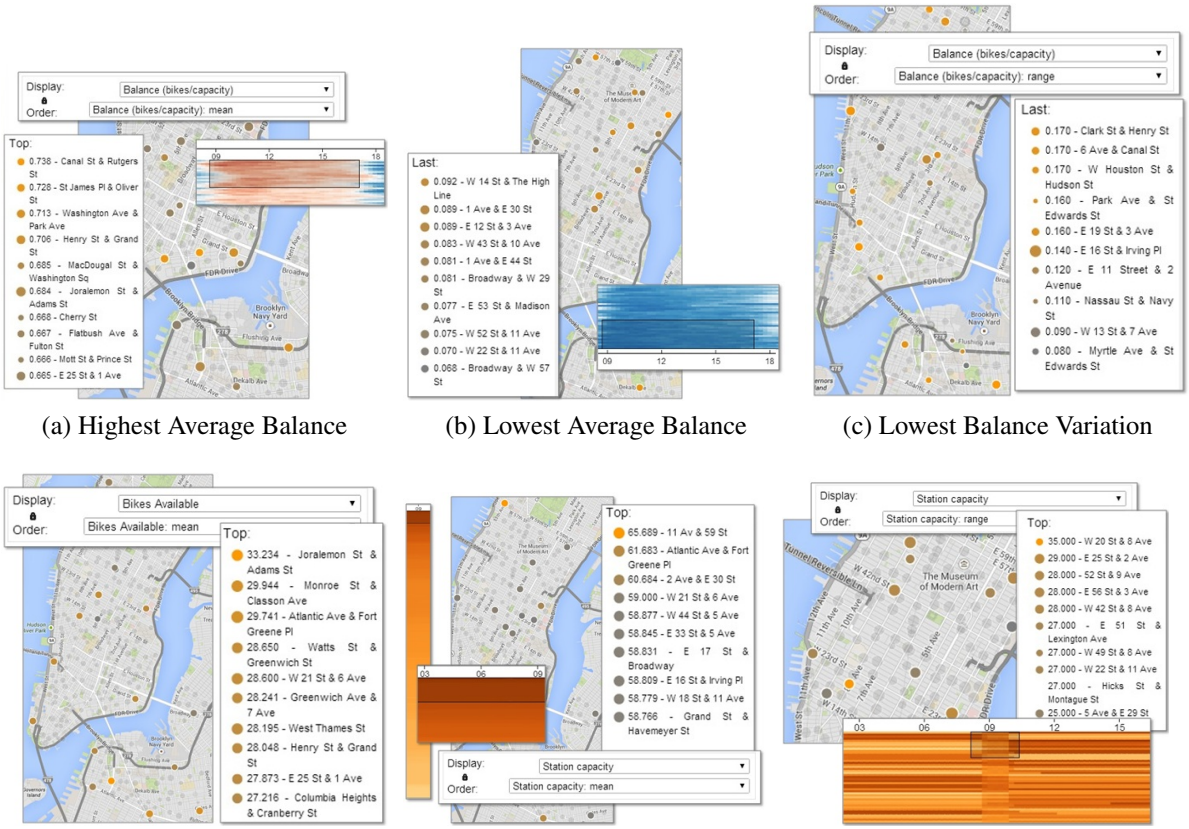


Figure 8.12: [Partial ordering and stations roles] The scale on the top is a zoomed view of the 24-hours timescale in the top of each of the four profiles below. **Top:** By ordering the stations lines in the matrix considering only balance reading from the first 1/3 of the day exposes the pattern of alternating roles (Left), switching first around 8 am and again at 6 pm. The same operation, however, shows no distinguishable visual pattern for weekends (Right), resulting in a noisy image, showing low regularity in the system usage, no clear definition of roles for the stations, and a more even distribution of balance throughout the system on those days. **Bottom:** Ranking the stations by Bikes arrival frequency around 9 am highlights the opposite behavior, as stations with low frequency of bikes arrival around 9 am have a high rate of arrivals at 6 pm. Showing how commuters travel to a set of stations by the morning and return to the stations in the complementary set at night (Left). Changing the visualized variable to balance, while still using the same ranking (bikes arrivals around 9 am), naturally groups the stations by their roles as sinks and sources (Right).



(a) Highest Average Balance (b) Lowest Average Balance (c) Lowest Balance Variation

(d) Highest Amount of Bikes Available (e) Highest Capacity (f) Highest Capacity Variation

Figure 8.13: [Finding stations by behavior] By ordering the rows of the matrix by different variables and reduce operators while using the brush to select the time period and the stations to highlight in the map, the 330 stations can be queried by their historic data. (a) and (b) show the most full and empty stations, on average, during working hours of Wednesdays during Summer. In the map, the division between most full stations bellow midtown and empty ones to the north is clear. (c) shows the stations with lowest variation of balance throughout the day. (d) Stations with the highest number of bikes during working hours. (e) and (f) show the capacity of the stations in the matrix, however, the first show the data aggregated for typical Wednesdays during Fall and select the biggest ones, the last visualizes the capacity on 01/28/2014. Data from this day shows an unusual change in capacity for a set of stations around 9 AM. Applying the brush to this interval and ordering by range of capacity points out the most affected stations.

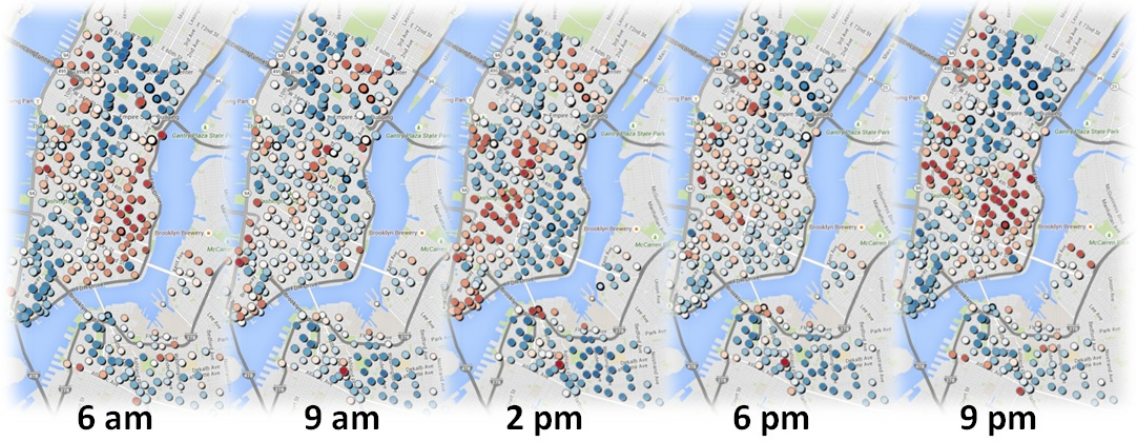


Figure 8.14: [Stations' Roles] We ordered the stations by their average balance at diferent hours in an usual weekday during Fall of 2013. As a result we can see clearly how the stations change of roles (as providers and receivers) in a day. In the early morning, most stations at the sides of Manhattan are almost full (red dots). This distribution changes as time goes by and is almost the opposite at 2 pm. Later, it comes back to the initial setting at 10 pm.

as shown in (e). As the capacities should remain constant for each station, the expected pattern is the smooth vertical gradient shown in (e), where we are visualizing the average capacity for Wednesdays during Fall. Visualizing the capacity profile for a single day sometimes reveal some discontinuities. (f) shows a special occurrence of such anomaly in the capacity profile on 28th January, when there is an alignment of capacity change for some stations right before 9 AM and then right after again. Selecting such interval and ordering the rows by capacity range we group on the top the stations when the discontinuity was sharper.

8.3.4 Circulation Dynamics

With partial reordering, we can track the migration of bikes in the map through the day. We already saw the pattern of roles swapping between stations at 9 am and 8 pm during weekdays, now in figure 8.14 we add spatial context by creating a timelapse of the rank of balance in the map. Red dots point full stations while the blue ones are empty, thus there are more bikes in the red regions. The day begins with most bikes in riversides regions of Manhattan. At 9 am there are few red spots. Since its the rush hour there are few bike parked, bikers are commuting to work. At early afternoon, we have the opposite scene of 6 am, a concentration of full stations in the middle, with the sides and Brooklyn in blue. Later it reverses again and late night is much like early mornings, with slightly higher concentration at Williamsburg and East Village.

Another interesting perspective is given in figure 8.15, where the subject is the overall usage distribution (weekdays during Fall of 2013). The begins with little activity except for stations nerby Penn Station (actually, those stations never really stop). Frequency spreads in Manhattan towards Financial District and Midtown at 9 am, decrasing again only late in the night. Also, all the stations along the Broadway showed to be popular destinations. There is some increase in Brooklyn's downtown area also, but never as intense as in Manhattan.

As shown before, there is a great difference in the stations usage during weekdays

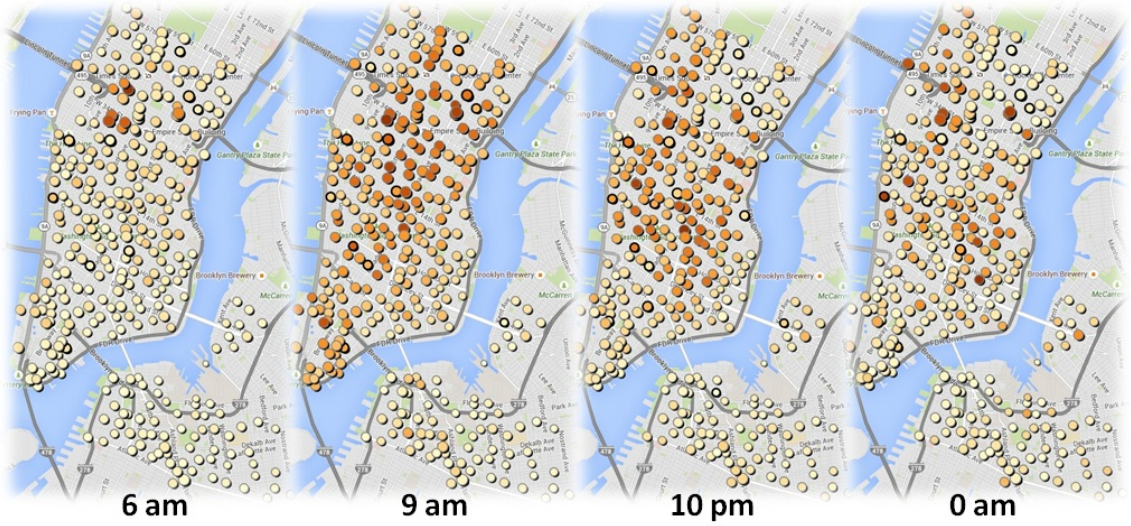


Figure 8.15: [Frequency Cycle on Weekdays] A timelapse of the frequency in typical weekdays. The stations around Penn Station are the first to show movement (6 am). Frequency increases throughout Manhattan as time goes by in a similar pattern as the one with balance in figure 8.14

and weekends. During weekends there is no clear subdivision of the city into constrasting regions. However, an evident pattern is the increase of cyclic trips. In figure 8.16 we inspect the trips from Sundays during the Summer of 2014, and rank the stations by the average amount of cyclic trips between 09 am and midnight, showing that they are more frequent around leisure spots like Central, Battery, and Brooklyn Bridge Parks, the High Line and also Williamsburg. Also, another difference between weekdays and weekends can be seem by comparing the spread of outages as in 8.17. While it is more frequent during weekdays in the north most stations in Manhattan, East Village and a region of Brooklyn, at the weekends they become unusual in the last two, but a problem in Williamsburg.

In figure 8.18 we use trip matrix view to query trips with lowest balance difference. We think this information is useful for users of the bike-sharing system as lowest balance difference values expose trips with the ideal case (full outage at the origin and empty outage at the destination). Having a full outage in the origin station, users could take a bike without disruption and eventually leave it in an empty station (empty outage). March 2014 was used for this test, showing us that during the evening, trips from the middle region to the north or south are the safest to avoid outages.

The movement of users during weekdays and weekends reveal different patterns. In figure 8.19, trips matrix for September 2013 are compared while displaying balance difference sampled at three day intervals for weekdays and weekend. We can see that during workdays more outages happen between the afternoon interval. The number of trips with full and empty outages at night is higher during weekdays than weekends, reflecting the behavior of people going back home or shopping around after working hours. The number of problematic trips is higher on weekends than over weekdays in the mornings, probably because the number of entertainment trips during Saturdays and Sundays.

Managers are also interested to know where they should increase the capacity of a station to aid the rebalancing problem. To meet this requirement, we opted for August 2014 as a sample to analyze the capacity difference between stations. In figure 8.20, the

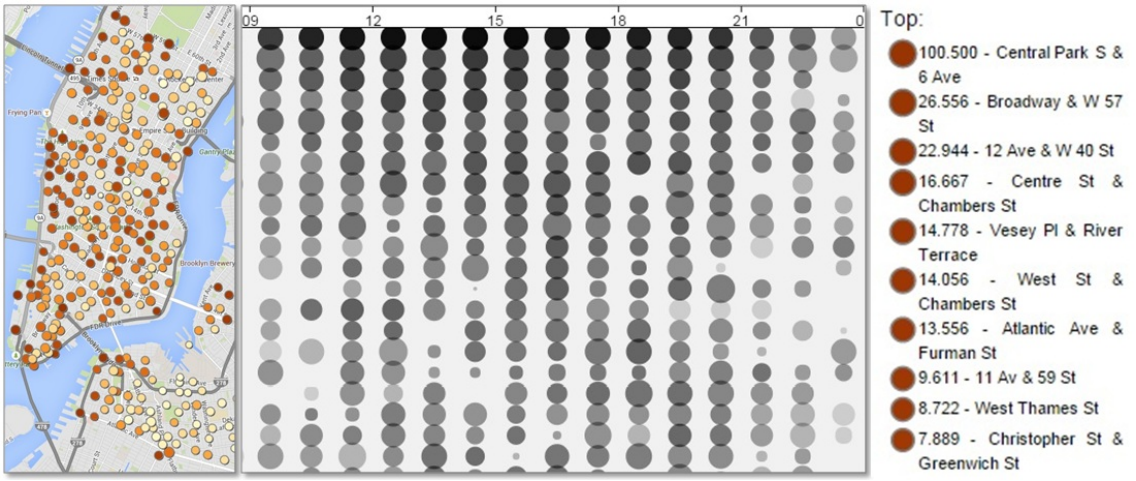


Figure 8.16: [Cyclic trips on Sundays] Stations ranked by the average number of cyclics trips on Sundays in the Summer of 2014. Cyclic trips seems to be normally distributed around 4 pm, in the interval between 09 am and 0 am. The Central Park station shows the highest number of cyclic trips per hour.

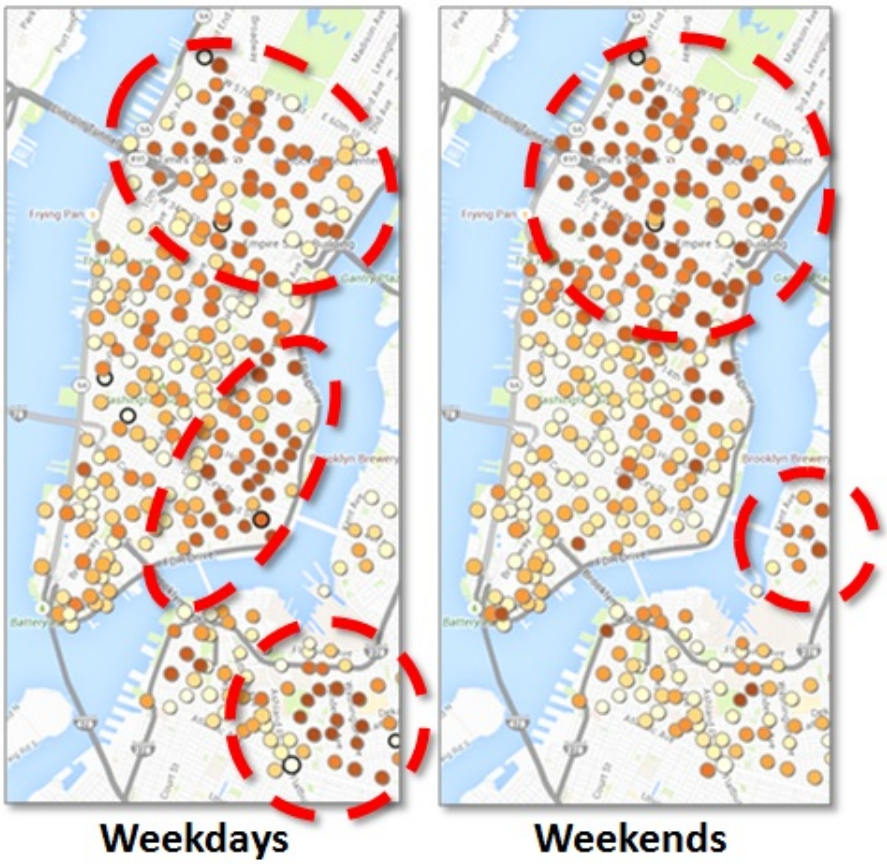
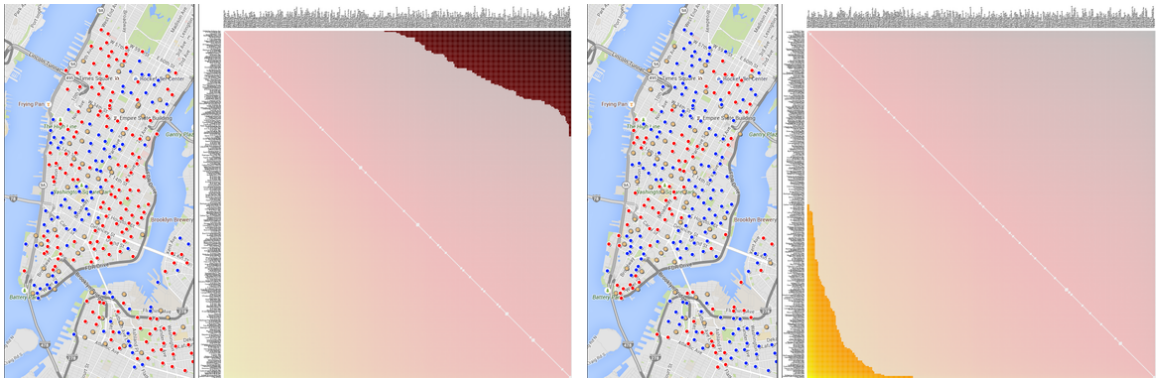
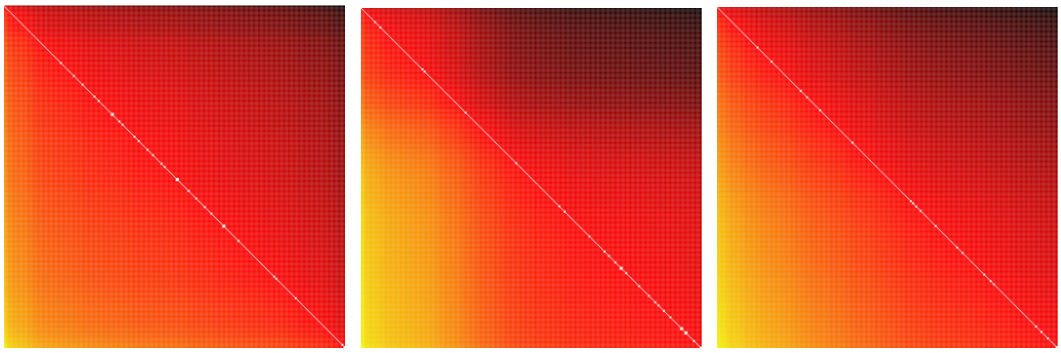


Figure 8.17: [Outages] Ranking the stations by the number of outages (both empty and full types) in weekdays and weekends. During weekdays we can spot three areas with higher concentration of stations constantly suffering from outages: Midtown and East Village in Manhattan, and between Fort Greene Park and the Pratt Institute in Brooklyn. In the weekends, there are more outages around Midtown and they also become more frequent in Williamsburg.



(a) Lowest 20% in the afternoon (b) Highest 20% in the afternoon

Figure 8.18: [Trips Matrix - Balance difference]. High and low selection for weekdays of March 2014, matrix ordered by station balance. As a result, we can see station's roles inverted. High balance difference (yellow selection) identifies trips between stations with outages. Lowest balance difference shows trips without problems.

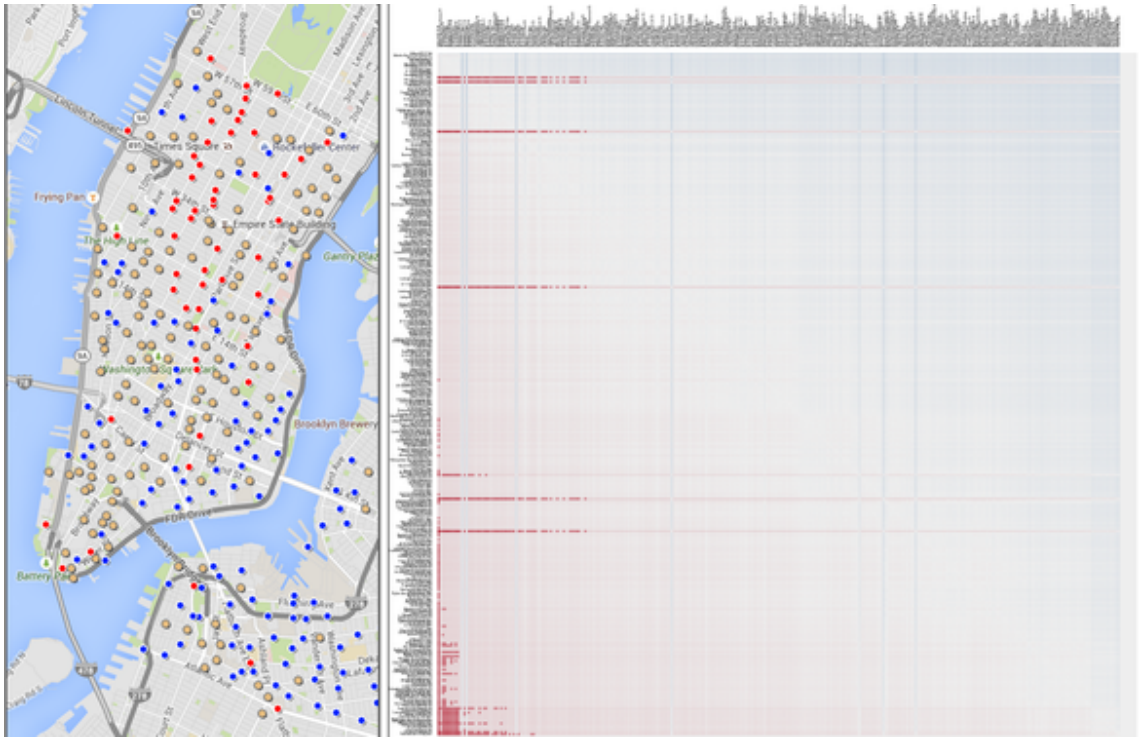


(a) Morning (b) Afternoon (c) Night

Figure 8.19: [Trips Matrix - September weekdays] Comparison of trips balance difference at three day intervals of September 2013. Aggregated by weekdays. Yellow color is mapped to highest values (outages in both stations of the trip). Red shows stations with similar balance and black identifies trips with lowest balance difference.

trips matrix uses a diverging color scale, mapping high capacity difference values to red colors and low values to blue colors. To analyze the geographical behavior of incoming and outgoing stations we chose to do a selection of trips with the highest 10% capacity difference. This selection reports trips from smaller stations to bigger ones, generally located in the bottom left corner of the trips matrix. The visualization allows us to detect outliers with the measurement of capacity. Stations with incorrect capacity value are: E 33 St & 1 Ave, W 43 St & 6 Ave, W 13 St & 5 Ave and W 49 St & 5 Ave. We attribute this to an incorrect measure due to a system failure.

The perspectives drawn with the 10-months timeline (figure 8.9) showed its potential to trace the trends of how the behavior of the commuters change as the program matures (design requirement **R4**) and the city goes under climate changes. The same trends can be found by navigating through the different periods available in the 24-hours view while also showing the cyclical patterns of the weeks (figure 8.10). A straightforward analysis that followed was the comparison between the different days of the week, between the



(a) Highest 30% values

Figure 8.20: [Trips Matrix - Capacity difference] Time period: August 2014 aggregated by weekdays. Trips selection by highest 10% capacity difference. A highest capacity difference represents trips from smaller stations to bigger ones. We can see these kind of trips were usually from the south region to the north region of New York.

same days of the week in different periods, and between weekdays and weekend usage. Anomalies turned out to be easily identified, as specific days and hours that present clear discordance from the expectation. These anomalies are perceived as a harsh change of color that show up for many stations in the matrix at a curiously well-defined time period. One example was detailed in figure 8.11. Facing the strong regularity of these anomalies and the fact that we could neither find any spatial correlation between the affected stations (proximity), nor some special event in the city at the given period of time we believe those outliers are related to operational activities in the program or malfunctioning issues in the stations' state tracking and feeding systems. The last option is even more likely regarding those cases when the number of stations affected is too high, so it is unlikely that the operational staff could be able to operate in so many places with such coordination.

Another analysis depicted the identification of those stations that fit a given criteria, like being empty or full (design requirement **R1**). We did show how the reordering of the matrix timeline by different variables can intuitively help spotting in the map those that exhibit the behavior we are looking for. We presented the distribution of the most full and empty stations during work hours in the map (figures 8.13(a) and (b)); identifyied those stations that show almost no balance change in usual days (figure 8.13(c)); showed the stations that keep the highest amount of bikes in such hours (figure 8.13(d)) and found unexpected variations in the capacity of the stations (comparing figures figures 8.13(e) and (d))). Apart from the search of patterns and stations, the reordering brush was also essential to create timelapses that show the progression of the ranks in the time. Figures 8.14 and 8.15 showed how the bikes move between the different regions (related to the

classification by roles **R3**), and also the evolving rate of use of a typical working day. In figure 8.17, by ranking stations by the frequency of outages in both weekdays and weekends we saw the changes of popularity between regions of Manhattan, Brooklyn and Williamsburg.

The seasonal trend was clearly present in the results, with great decrease in the rate of commuting during cold months (even though it was still surprisingly high given the harsh weather). We could find no explanation for the anomalies identifyied by relating them to unusual events in the city (design requirement **R2**), resting in the assumption that those where caused by operational issues of the Citi Bike program. However, the several results presented validate the hypothesis that data from bike-sharing can be used to provide many cues of the city life style, and that the method we proposed is fit for its purpose.

Operating a bike-sharing system deployed in a big city is a challenging task due to the intense commuting dynamics and its complexity. In such scenario, the number of outages increases, rebalancing requires more effort as the system is usually larger (more stations and bikes) and the rebalancing fleet is subject to traffic jams specially when the system provides 24-hour service and rebalance must be done on the fly. Expecting that a deeper and clearer understanding of the system dynamics may help in the operational efforts to provide a better service, we introduced a visualization design to support the exploration of a dataset with a long history of stations' usage footprint and user trips of a bike-sharing system. Data can be visualized as it was registered at each specific day, and also aggregated over different time periods to represent the expectation back then. Using New York City's Citi Bike program as a case study, our designs lead to a substantial variety of insights, presented and discussed to validate the applicability of the proposed solution. Our results showed the changes in the activity of the bikers over a 10-months period from different perspectives. These changes are related to both the adoption of the bike-sharing as a new transportation mode in NYC and the weather influence over cycling. A number of anomalies were spotted in the different overview perspectives, and further exploration revealed the respective days, hours and stations with abnormal trace. Those events shared a steep and synchronized change in the trace values of several stations at once, and due to this odd regularity are expected to be reflections of operational activity or system malfunction.

9 FINAL CONSIDERATIONS

In the work on running races, the designs we created proved to be of use in the comparison of different races, allowing, for instance, to assess the physical condition of the studied population. We expect such designs to be useful for elaborating new running strategies, or to help organizers to better plan running events by understanding how the race course affects the runners. The analysis of related datasets, like similar groups of runners on the same and different races, can allow the comparison between those groups, establishing training goals for groups of lesser performance, for example. Also, the use of more variables and the analysis of other exercise modes, like cycling, are also examples of interesting topics for further research. This work lead to a collaboration with a researcher in physiology at *IPA - Centro Universitário Metodista*, Professor Maristela Padilha, and a publication at the *Brazilian Symposium on Computer Graphics and Image Processing - SIBGRAPI*. Also, we began another work in the same domain in collaboration with Professor Gustavo Nonato at *USP São Carlos*. This work focused in the analysis of history of exercises of a single runner, and also the comparison of different runners by their histories of exercises. It was interrupted with the beginning of a period of internship with the visualization group at the Polytechnique Institute of the New York University.

In the domain of the bike-sharing systems, our prototype lead us to a substantial number of insights regarding the usage of the Citi Bike program in 10-months, proving the applicability of the scheme of ranked mapping of time series in the context of bike-sharing systems. This work resulted from a collaboration with the visualization group of Professor Claudio Silva at *NYU-Poly* and was recently submitted to the *IEEE Transactions on Visualization and Computer Graphics - TVCG*.

We brought the view of time series represented visually compressed to the context of exploratory visual analysis of spatio-temporal data, as a component of coordinate views to give full temporal context of several series at once in a compact fashion. By applying the general idea of viewing the set of series as a ordered stack of thin lines to the two major works developed in this thesis, we showed how such representation can be used with different facets of spatio-temporal data. We believe the results presented in both works serve as evidence to support our claim on the urge for views of sets of time series in the exploration of spatio-temporal datasets, and the value of the ordered stacks of series as a useful component to support such analysis.

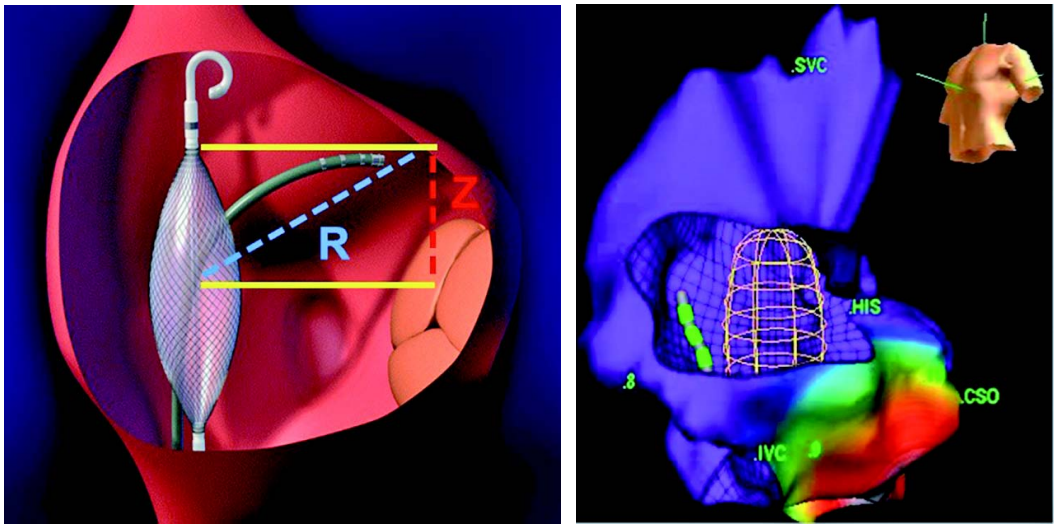
APPENDIX A ATRIAL FIBRILLATION

This work was the final result of a collaboration that begun with the visit of professor Fernando Schlindwein to our Computer Graphics and Interaction group during 2013. Prof. Schlindwein and his student Joao Salinet have been working in the Bioengineering Research Group of the University of Leicester, in the UK, with methods to analyse electrocardiograms' data. Knowing bout the expertise of our group in the field of GPU computing, prof. Schlindwein came to us looking for a collaboration in one of their works. The collaboration resulted in a paper titled *Visualizing intracardiac atrial fibrillation electrograms using spectral analysis*, which was published in the Computing in Science & Engineering journal. In this section, we summarise the outcome of this collaboration.

Atrial fibrillation (AF) is the most common cardiac arrhythmia, and it is associated with increased risk of stroke, heart failure, and mortality [70]. Prof. Schlindwein and Salinet developed a method that uses spectral analysis techniques to aid the identification of sources of atrial fibrillation. They expected that the use GPU computing and visualization would be an affordable approach to reduce the time to process and analyse the electrocardiogram data, thus leading to improvements in the treatment of the medical condition.

A.1 Related Works

Measuring and modeling the genesis and propagation of the electrical activity in the heart in quantitative terms is a very important area of research that will help understand and treat heart arrhythmias. Atrial fibrillation (AF) is a heart rhythm disturbance characterized by uncoordinated and rapid electrical atrial activation which takes over from normal sinus rhythm, with consequent deterioration of the mechanical ability of the atria to pump blood effectively. The ventricles will beat irregularly and rapidly during AF when conduction is intact. On the ECG, the wave of depolarization that spreads throughout the atria, called P waves, are replaced by rapid, small amplitude oscillations which vary in amplitude, shape, and timing between QRS complexes (Fuster *et al.*[**?**]), which corresponds to the three graphical deflections (Q, R and S waves) seen on a typical electrocardiogram. AF is a heart rhythm disturbance characterized by uncoordinated and rapid electrical atrial activation that takes over, with consequent deterioration of the mechanical ability of the atria to pump blood effectively. This malfunction can cause serious problems like stroke. It is the most common cardiac arrhythmia encountered in clinical practice with a prevalence of 1-2% of the general population [**?**]. The symptoms of AF include palpitations, tiredness, shortness of breath, dizziness and chest pain. As the mechanical pumping ability of the atria is compromised, the resulting pooling of blood



(a) Sensing (b) Electrocardiogram Visualization

Figure A.1: **Electrocardiogram and Ablation.** (a) The ablation catheter and the EnSite 3000 balloon array are first used to map the inner surface of the hearts chamber using the ablation catheter as a probe and the balloon as reference. Then the array collects atrial electrograms (potentials) at 1,200 Hz. (b) The system can display the potentials as a color map. (figure from [70])

in the atria increases the long-term risk of stroke fivefold [**?**]. AF is a public health problem with approximately 3,700/year per patient being spent in Europe [**?**]. The cost of a catheter-based ablation procedure is about 12,500. Any advances in the understanding of this condition, especially advances that might lead to more effective treatment are, therefore, of great importance.

In 1913 Mines [**?**], studied the vulnerability of an excitable circle of cells in the heart. It follows from that original idea that if the concept of *reentry* is to be applied to atrial fibrillation, there would be a preferred range of *dominant frequencies* associated with the circuits. If reentry circuits are formed, knowing the velocity of propagation of the electrical activation (typically slower than 20 cms^{-1}), the size of the atrium (up to 6 cm) and, most importantly, the duration of the refractory period (about 200-240 ms in normal cardiac cells, but reduced down to about 80-85 ms in AF [**?**] then the DF range associated with AF would be between 4.2 Hz and 12.5 Hz. In recent years, a noncontact multielectrode array catheter has been developed to assist with the mapping of intracardiac electrical signals in complex arrhythmia cases. This innovation allows the 3D reconstruction of the hearts chambers, and projection of the recorded electrical activity onto its geometry as a simultaneous high density of electrograms. Figure A.1 shows a schematic of this process and a 3D representation of the reconstructed surface. Instead of using the sampled electrocardiogram signals, Salinet and Schlindwein apply spectral analysis on them to extract the dominant frequencies (DF). These DFs are then mapped on the 3D surface as colors, and used as indicators of regions that can be causing AF. Figure A.2 shows the initial electrocardiogram signals mapped in the 3D surface. They are divided into consecutive time windows, and goes through spectral analysis, to evaluate the DF at each point in the surface.

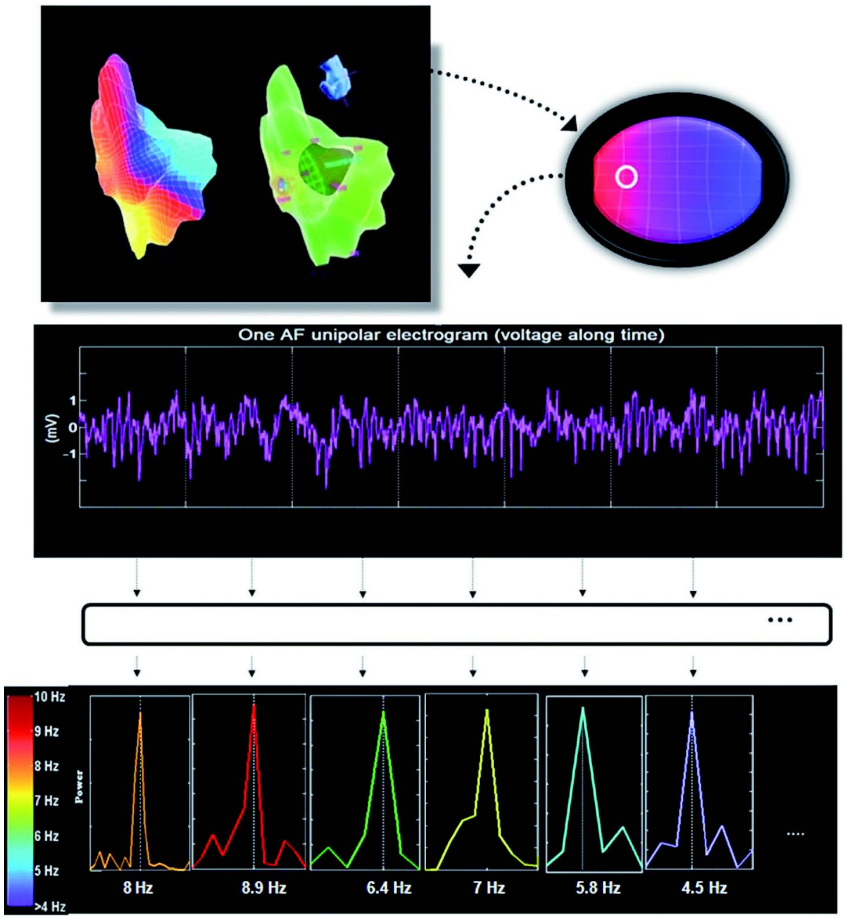


Figure A.2: **Spectral analysis of the dominant frequency (DF).** The data collected by the EnSite 3000 system allows manipulation of the atrium on the screen and displays next to it the torsos orientation. For each of the 2,048 points of the 3D surface, we obtain the atrial electrograms DF for each segment along time and then color code the surface according to the DFs frequency. (figure from [70])

A.2 Results

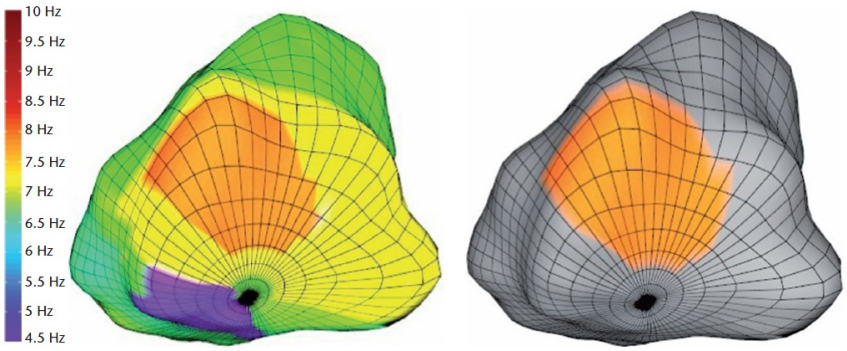


Figure A.3: **3D DF mapping and highest DF identification.**(a) The left atriums 3D representation, including the mapping of the DFs. The DFs (represented in a color scale) will help doctors visualize the behavior of the atriums electrical activation in the frequency domain in real time. (b) The system can also automatically identify the region corresponding to the highest DF area. (figure from [70])

We implemented a prototype to extract the DFs and visualize how they change over time. Figure A.3 shows the mapping of the DFs of a single time window to the atrium surface, with the colormap used. By repeating the process for each consecutive time window, we create an animation that shows the movement of regions with different DF over the surface (see figure A.4).

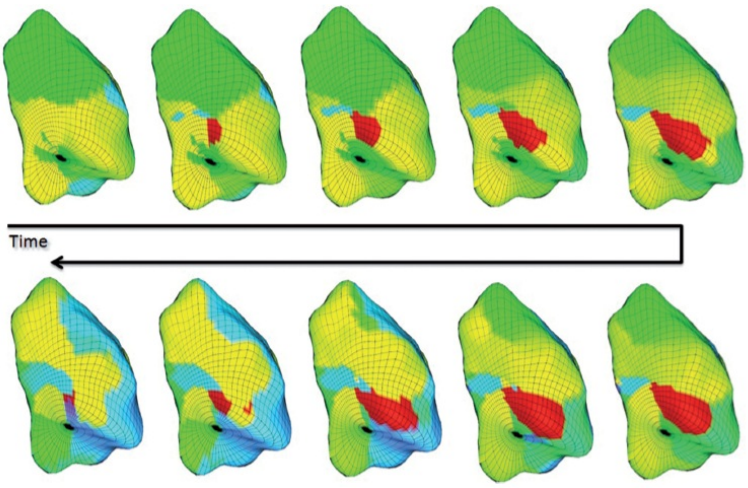
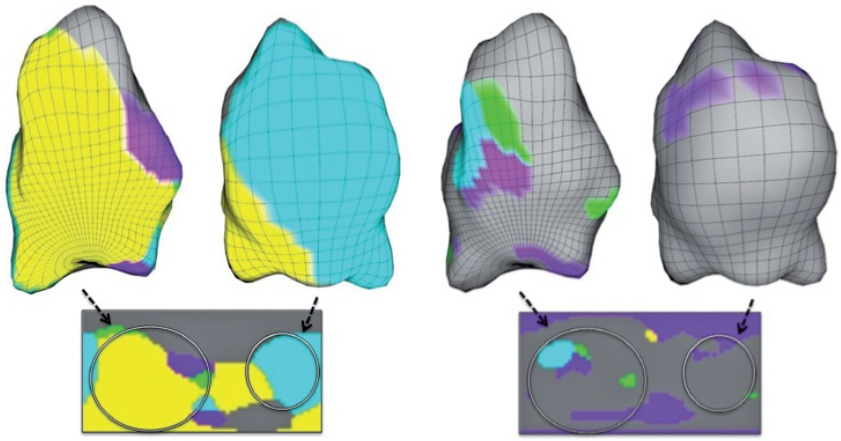


Figure A.4: **Consecutive DF maps.** Consecutive DF maps using 92 percent of overlapping between windows. Using more overlapping creates more frames and a smoother animation that helps to understand how the different DF zones evolve over the surface along time. (figure from [70])

The treatment of AF is the ablation (burning) of a point in the atrium surface. With the visual representation of the regions of DFs in the atrium surface, doctors reason on the best point to ablate, that will hopefully fix condition. Figure A.5 shows a visualization of the DFs before and after an ablation is performed. We implemented the spectral analysis stage using a single CPU core, 4 CPU cores, and a GPU, and compared the processing times (see figure A.6). The comparison shows that using the processing power of modern GPUs, it is feasible to implement a pipeline in which data acquisition, computation and visualization can be done in real-time. The 3D representations used, can be displayed with the same equipment and manipulated in exactly the same way as they are by cardiologists who perform the catheterization, and ablation procedures, but now with immediate feedback about the ablation impact.



(a) Before Ablation (b) After Ablation

Figure A.5: **3D DF mapping before and after ablation.** 3D DF mapping of the left atrium of a patient with persistent atrial fibrillation(AF). (a) The baseline DF map. (b) The DF immediately after the standard pulmonary veins isolation (PVI) procedure. This figure demonstrates a general reduction of the size of the DF areas, a reduction of the DF values, and a reduction in the complexity of the DF areas after the PVI procedure. (figure from [70])

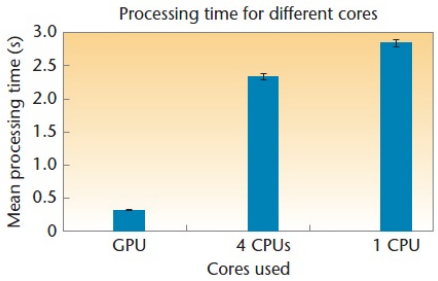


Figure A.6: **Comparison of processing times.** Comparison of processing times between single CPU core, multiple CPU cores and the GPU. (figure from [70])

REFERENCES

[1] Google directions. https://developers.google.com/maps/documentation/directions/.

[2] Openstreetmap. http://www.openstreetmap.org/.

[3] Routino : Router for openstreetmap data. http://www.routino.org/.

[4] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski. Visualizing time-oriented dataâA systematic view. *Computers & Graphics*, 31(3):401–409, June 2007.

[5] W. Aigner, S. Miksch, W. Muller, H. Schumann, and C. Tominski. *Visualization of Time-Oriented Data*. Springer, 2011.

[6] D. Albers, C. Dewey, and M. Gleicher. Sequence Surveyor: Leveraging Overview for Scalable Genomic Alignment Visualization. *IEEE Trans. Vis. Comput. Graph.*, 2011.

[7] G. Andrienko and N. Andrienko. *Exploratory Analysis Of Spatial And Temporal Data*. Springer, 2005.

[8] N. Andrienko and G. Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005.

[9] N. Andrienko and G. Andrienko. A visual analytics framework for spatio-temporal analysis and modelling. *Data Mining and Knowledge Discovery*, 27(1), 2013.

[10] N. Andrienko, G. Andrienko, and P. Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *Journal of Visual Languages Computing*, 14(6):503 – 541, 2003. Visual Data Mining.

[11] R. Beecham, J. Wood, and A. Bowerman. A visual analytics approach to understanding cycling behaviour. In *Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on*, 2012.

[12] W. J. Beecham, R. and A. Bowerman. Identifying and explaining interpeak cycling behaviours within the london cycle hire scheme. In *Workshop on Progress in Movement Analysis: Experiences with Real Data*, 2012.

[13] L. Beis, M. Wright-Whyte, B. Fudge, T. Noakes, and Y. Pitsiladis. Drinking behaviors of elite male runners during marathon competition. *Clinical Journal of Sport Medicine*, 22(3), 2012.

[14] R. Benson and D. Connolly. *Heart Rate Training*. Human Kinetics, 1st edition, 2011.

[15] J. D. Berry, B. Willis, S. Gupta, C. E. Barlow, S. G. Lakoski, A. Khera, A. Rohatgi, J. A. De Lemos, W. Haskell, and D. M. Lloyd-Jones. Lifetime risks for cardiovascular disease mortality by cardiorespiratory fitness levels measured at ages 45, 55, and 65 years in men. The Cooper Center Longitudinal Study. *Journal of the American College of Cardiology*, 57, 2011.

[16] J. Bertin. *Semiology of Graphics*. University of Wisconsin Press, 1983.

[17] S. N. Blair, H. W. K. III, C. E. Barlow, J. Ralph S. Paffenbarger, L. W. Gibbons, and C. A. Macera. Changes in Physical Fitness and All-Cause Mortality. A Prospective Study of Healthy and Unhealthy Men. *Journal of The American Medical Association (JAMA)*, 273, 1995.

[18] S. N. Blair, H. W. K. III, J. Ralph S. Paffenbarger, D. G. Clark, K. H. Cooper, and L. W. Gibbons. Physical Fitness and All-Cause Mortality. A Prospective Study of Healthy Men and Women. *Journal of The American Medical Association (JAMA)*, 262, 1989.

[19] P. Borgnat, P. Abry, P. Flandri, C. Robardet, J.-B. Rouquier, and E. Fleury. Shared bicycles in a city: A signal processing and data analysis perspective. *Advances in Complex Systems*, 2011.

[20] E. R. Burke. *Precision Heart Rate Training*. Human Kinetics, 1998.

[21] M. R. Carnethon, M. Gulati, and P. Greenland. Prevalence and Cardiovascular Disease Correlates of Low Cardiorespiratory Fitness in Adolescents and Adults. *Journal of The American Medical Association (JAMA)*, 294.23, 2005.

[22] V. Chiraphadhanakul. Large-scale analytics and optimization in urban transportation: improving public transit and its integration with vehicle-sharing services. Master's thesis, Massachusetts Institute of Technology, Sloan School of Management, Operations Research Center, 2013.

[23] L. L. Constantine. Canonical abstract prototypes for abstract visual and interaction design. *Components*, 1(978):1–15, 2003.

[24] J. Dill and J. Gliebe. Understanding and measuring bicycling behavior: A focus on travel time and route choice. Technical report, 2008.

[25] N. Elmqvist and P. Tsigas. A taxonomy of 3d occlusion management techniques. *2007 IEEE Virtual Reality Conference*, pages 51–58, 2007.

[26] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva. Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2149–2158, Dec. 2013.

[27] J. Ferzoco. How new yorkers and tourists use citi bike on two nice days. http://ny.curbed.com/tags/jeff-ferzoco.

[28] M. Flegenheimer. The balancing act that bike-share riders just watch. http://www.nytimes.com/2013/08/15/nyregion/the-balancing-act-that-bike-share-riders-just-watch.html.

[29] M. Flegenheimer. Bike-share effort draws riders and hits snags. http://www.nytimes.com/2013/06/12/nyregion/two-weeks-in-riders-and-errors-for-bike-share-effort.html?pagewanted=all.

[30] A. U. Frank. Different types of "times" in gis. *Spatial and temporal reasoning in geographic information systems*, pages 40–62, 1998.

[31] J. Friel. *Total Heart Rate Training*. Ulysses, 2006.

[32] M. Friendly. The history of the cluster heat map. *The American Statistician*, 2009.

[33] J. Froehlich, J. Neumann, and N. Oliver. Measuring the Pulse of the City through Shared Bicycle Programs. In *Proceedings of the International Workshop on Urban, Community, and Social Applications of Networked Sensing Systems (UrbanSense08)*, 2008.

[34] J. Froehlich, J. Neumann, and N. Oliver. Sensing and predicting the pulse of the city through shared bicycling. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, 2009.

[35] Garmin. Garminconnect. http://connect.garmin.com/.

[36] M. Guenther and J. Bradley. Journey data based arrival forecasting for bicycle hire schemes. In A. Dudin and K. De Turck, editors, *Analytical and Stochastic Modeling Techniques and Applications*, volume 7984 of *Lecture Notes in Computer Science*, pages 214–231. Springer Berlin Heidelberg, 2013.

[37] D. Guo, J. Chen, A. M. MacEachren, and K. Liao. A visualization system for space-time and multivariate patterns (vis-stamp). *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1461–1474, Nov. 2006.

[38] S. Gupta, A. Rohatgi, C. R. Ayers, B. L. Willis, W. L. Haskell, A. Khera, M. H. Drazner, J. A. De Lemos, and J. D. Berry. Cardiorespiratory fitness and classification of risk of cardiovascular disease mortality. *Circulation*, 112, 2005.

[39] E. Hajnicz. *Time Structures: Formal Description and Algorithmic Representation*. Springer, 1996.

[40] M. Hao, U. Dayal, D. Keim, and T. Schreck. Multi-Resolution Techniques for Visual Exploration of Large Time-Series Data. *Symposium A Quarterly Journal In Modern Foreign Literatures*, pages 1–8, 2007.

[41] W. Haskell, M. E. Nelson, R. Dishman, E. Howley, W. Kort, W. Kraus, I. Lee, A. McTiernan, R. Pate, K. Powell, J. Regensteiner, J. Rimmer, and A. Yancey. Physical Activity Guidelines Advisory Committee Report, 2008. June 18, 2008 2008.

[42] W. Javed, B. McDonnel, and N. Elmqvist. Graphical perception of multiple time series. *IEEE transactions on visualization and computer graphics*, 16(6):927–34, 2010.

[43] M. J. Joyner, J. R. Ruiz, and A. Lucia. The Two-Hour Marathon: Who and When? *Journal of Applied Physiology*.

[44] S. Kaufman. Citi bike and gender. http://wagner.nyu.edu/rudincenter/2014/05/citi-bike-and-gender/.

[45] S. Kaufman. Citi bike and "reactionary biking". http://wagner.nyu.edu/rudincenter/2014/03/citi-bike-and-reactionary-biking/.

[46] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov.*, 7(4):349–371, Oct. 2003.

[47] J. H. Kim, R. Malhotra, G. Chiampas, P. d'Hemecourt, C. Troyanos, J. Cianca, R. N. Smith, T. J. Wang, W. O. Roberts, P. D. Thompson, and A. L. Baggish. Cardiac Arrest during Long-Distance Running Races. *New England Journal of Medicine*, 366(2), 2012.

[48] R. Kincaid and H. Lam. Line Graph Explorer : Scalable Display of Line Graphs Using Focus + Context. *Main*, 2006.

[49] M. Krstajic, E. Bertini, and D. Keim. CloudLines: Compact Display of Event Episodes in Multiple Time-Series. *IEEE transactions on visualization and computer graphics*, 17(12):2432–9, Dec. 2011.

[50] J. Larsen. Bike-sharing programs hit the streets in over 500 cities worldwide. http://www.earth-policy.org/plan$_{bu}pdates/2013/update112$.

[51] J. Larsen. Dozens of u.s. cities board the bike-sharing bandwagon. http://www.earth-policy.org/plan$_{bu}pdates/2013/update113$.

[52] N. Lathia, S. Ahmed, and L. Capra. Measuring the impact of opening the london shared bicycle scheme to casual users. *Transportation Research Part C: Emerging Technologies*, 22, 2012.

[53] A. M. MacEachren. *How Maps Work: Representation, Visualization and Design*. Guilford Press, 1995.

[54] N. B. Maps. Citi bike 2013 summary. http://www.nycbikemaps.com/spokes/citi-bike-2013-summary/.

[55] S. C. Mathews, D. L. Narotsky, D. L. Bernholt, M. Vogt, Y.-H. Hsieh, P. J. Pronovost, and J. C. Pham. Mortality Among Marathon Runners in the United States, 2000-2009. *Am J Sports Med*, 2012.

[56] P. Mclachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC - interactive visual exploration of system management time-series data. In *In Proc. ACM Conf. Human Factors in Computing Systems (CHI*, 2008.

[57] A. Murray and R. J. S. Costa. Born to run. Studying the limits of human performance. *BMC Med*, 10(1), 2012.

[58] O. O'Brien. 5.5 million journeys at nyc bike share. http://oobrien.com/2014/04/5-5-million-journeys-at-nyc-bike-share/.

[59] O. O'Brien. Bike share map update â 6 new cities, weather, stats. http://oobrien.com/2011/12/bike-share-map-update-6-new-cities-weather-stats/.

[60] O. O'Brien. A glimpse of bike share geographies around the world. http://oobrien.com/2012/01/a-glimpse-of-bike-share-geographies-around-with-world/.

[61] O. OâBrien, J. Cheshire, and M. Batty. Mining bicycle sharing data for generating insights into sustainable transport systems. *Journal of Transport Geography*, 34(0), 2014.

[62] O. O'Brien and P. DeMaio. The bike-sharing world map. www.bikesharingworld.com, 2007.

[63] F. Ogilvie and A. Goodman. Inequalities in usage of a public bicycle sharing scheme: Socio-demographic predictors of uptake and usage of the london (uk) cycle hire scheme. *Preventive Medicine*, 55(1), 2012.

[64] P. Papazek, G. Raidl, M. Rainer-Harbach, and B. Hu. A pilot/vnd/grasp hybrid for the static balancing of public bicycle sharing systems. In *Computer Aided Systems Theory - EUROCAST 2013*, volume 8111 of *Lecture Notes in Computer Science*. 2013.

[65] F. Paternò, C. Mancini, and S. Meniconi. *ConcurTaskTrees: A Diagrammatic Notation for Specifying Task Models*, volume 96, pages 362–369. Citeseer, 1997.

[66] T. Petersen and M. Robert. *Optimising Bike Sharing in European Cities*. 2011.

[67] G. Raidl, B. Hu, M. Rainer-Harbach, and P. Papazek. Balancing bicycle sharing systems: Improving a vns by efficiently determining optimal loading operations. In *Hybrid Metaheuristics*, volume 7919 of *Lecture Notes in Computer Science*. 2013.

[68] M. Rainer-Harbach, P. Papazek, B. Hu, and G. R. Raidl. Balancing bicycle sharing systems: A variable neighborhood search approach. In *Proceedings of the 13th European Conference on Evolutionary Computation in Combinatorial Optimization*, EvoCOP'13, 2013.

[69] H. Reijner. The Development of the Horizon Graph. *Electronic Proceedings of the VisWeek Workshop From Theory to Practice: Design, Vision and Visualization*, 2008.

[70] J. L. Salinet Jr, G. N. Oliveira, F. J. Vanheusden, J. L. D. Comba, G. A. Ng, and F. S. Schlindwein. Visualizing intracardiac atrial fibrillation electrograms using spectral analysis. *Computing in Science & Engineering*, 15(2):79–87, 2013.

[71] J. Schuijbroek and R. Hampshire. Inventory rebalancing and vehicle routing in bike sharing systems. *Working Paper at Carnegie Mellon University Research Showcase*, 2013.

[72] E. Tam, H. Rossi, C. Moia, C. Berardelli, G. Rosa, C. Capelli, and G. Ferretti. Energetics of running in top-level marathon runners from Kenya. *Eur J Appl Physiol*, 2012.

[73] C. Tominski and J. Abello. Axes-Based Visualizations with Radial Layouts. *Proceedings of the ACM Symposium on Applied Computing*, 2004.

[74] T. Urli. Balancing bike sharing systems (bbss): instance generation from the citibike nyc data. *CoRR*, 2013.

[75] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh. Experimental comparison of representation methods and distance measures for time series data. *Data Mining and Knowledge Discovery*, 26(2), 2013.

[76] Z. Wang, M. Lu, X. Yuan, J. Zhang, and H. v. d. Wetering. Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2159–2168, Dec. 2013.

[77] B. Wellington. Mapping citi bikeâs riders, not just rides. http://iquantny.tumblr.com/post/81465368612/mapping-citi-bikes-riders-not-just-rides.

[78] B. Wellington. Meet the busiest citi bike in all of new york city. http://gizmodo.com/meet-the-busiest-citi-bike-in-all-of-new-york-city-1580746440.

[79] M. Wilhelm, L. Roten, H. Tanner, J. Schmid, I. Wilhelm, and H. Saner. Long-term cardiac remodeling and arrhythmias in nonelite marathon runners. *Am J Cardiol*, 110(1), 2012.

[80] J. Wood, R. Beecham, and J. Dykes. Moving beyond sequential design: Reflections on a rich multi-channel approach to data visualization. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), 2014.

[81] J. Wood, A. Slingsby, and J. Dykes. Visualizing the dynamics of london's bicycle hire scheme. *Cartographica*, 2011.

[82] J. Woodcock, M. Tainio, J. Cheshire, O. O'Brien, and A. Goodman. Health effects of the london bicycle sharing system: health impact modelling study. *BMJ*, 348, 2014.

[83] M. Zaltz Austwick, O. OâBrien, E. Strano, and M. Viana. The structure of spatial networks and communities in bicycle sharing systems. *PLoS ONE*, 2013.

[84] C. Zhong, T. Wang, W. Zeng, and S. Müller Arisona. Spatiotemporal visualisation: A survey and outlook. In S. Arisona, G. Aschwanden, J. Halatsch, and P. Wonka, editors, *Digital Urban Modeling and Simulation*, volume 242 of *Communications in Computer and Information Science*, pages 299–317. Springer Berlin Heidelberg, 2012.