Filipe Laginha Pinto Correia

Nº 38663

Cartographic Representation of Spatiotemporal Phenomena

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador : João Moura Pires, Professor Auxiliar,
Universidade Nova De Lisboa

	Júri:
Presidente:	Rodrigo Seromenho Miragaia Rodrigues
Arguente:	Maribel Yasmina Santos
Vogal:	João Moura Pires

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

December, 2013

Cartographic Representation of Spatiotemporal Phenomena

Ao meu Avô Antonio

Acknowledgements

First of all, I would like to thank my supervisor João Moura Pires for his help in the completion of this thesis. I give thanks to my university, Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa, for the wonderfull years of academic life. I would also like to thank Novabase and QREN for giving me this opportunity and supporting my work, and thank my fellow colleagues of the research group, Alexandre Gonçalves, Manuel Santos, Ricardo Silva and Rui Alves, for their input and constant support throughout this year.

A big thanks to my parents and my brothers that put up with me through these times of work and stress. To all my close friends that tried very hard to take my mind of work which thankfully I was still able to finish.

This work was done with the support of the research project "GIAP - GeoInsight Analytics Platform (LISBOA-01-0202-FEDER- 024822)", funded by Comissão de Coordenação e Desenvolvimento Regional de Lisboa e Vale do Tejo (PORLisboa), included in Sistema de Incentivos à Investigação e Desenvolvimento Tecnológico (SI I&DT), through a MSc research fellowship from FCT-UNL.

Abstract

The field of geovisual analytics focuses on visualization techniques to analyze spatial data by enhancing human cognition. However, spatial data also has a temporal component that is practically disregarded when using conventional geovisual analytic tools. Some proposals have been made for techniques to analyze spatiotemporal data, but most were made for specific use cases, and are hard to abstract for other situations. There was a need to create a method to describe and compare the existing techniques.

A catalog that provides a clear description of a set of techniques that deal with spatiotemporal data is proposed. This allows the identification of the most useful techniques depending on the required criteria. The description of a technique in the catalog relies on the two frameworks proposed. The first framework is used for describing spatiotemporal datasets resorting to data scenarios, a class of datasets. Twenty three data scenarios are described using this framework. The second framework is used for describing analytical tasks on spatiotemporal data, nine different tasks are described using this framework.

Also, in this document, is the proposal of two new geovisual analytical techniques that can be applied to spatiotemporal data: the attenuation & accumulation map technique and the overlapping spatiotemporal windows technique. A prototype was developed that implements both techniques as a proof of concept.

Keywords: geovisual analytics, spatiotemporal data, spatiotemporal patterns...

Resumo

A área de geovisualização analítica foca-se em técnicas de visualização de dados georreferenciados num contexto analítico com o intuito de aumentar as capacidades cognitivas do analista. No entanto, dados espaciais também têm uma componente temporal que é praticamente descartada quando se recorre a ferramentas de geovisualização analítica convencionais. Existem algumas propostas de técnicas de análise de dados espácio-temporais na literatura, mas a maioria foi desenhada para casos de uso especifico e dificilmente são abstraídas para outras situações. Houve a necessidade de criar um método capaz de descrever e comparar as técnicas existentes.

Um catalogo que providencia uma descrição clara de um conjunto de técnicas que lidam com dados espácio-temporais é proposto. Isto possibilita a identificação das técnica mais úteis conforme os critério necessários. A descrição de uma técnica no catalogo tem como base duas *frameworks* propostas. O primeiro *framework* é usada para descrever *datasets* espácio-temporais recorrendo a classes de *datasets*, chamadas de cenários de dados. Vinte e três cenários são descritos usando este *framework*. O segundo *framework* é usado para descrever tarefas analiticas para dados espácio-temporais, nove tarefas diferentes são descritas usando este *framework*.

Adicionalmente são propostas duas novas técnicas de geovisualização analítica que podem ser aplicadas a dados espácio-temporais: *attenuation & accumulation map* e *overlapping spatiotemporal windows*. Foi desenvolvido um protótipo que implementa as duas técnicas como prova de conceito.

Palavras-chave: geovisualização analítica, dados espácio-temporais, padrões espácio-temporais. . .

Contents

List of Figures

List of Acronyms

API Application Programming Interface.

GIS Geographic Information System.

HTML Hyper Text Markup Language.

ICNF Instituto da Conservação da Natureza e das Florestas.

JSON JavaScript Object Notation.

PIIC Programa de Introdução à Investigação Científica.

QREN Quadro de Referencia Estrategico Nacional.

UI User Interface.

1

Introduction

This chapter serves as a first contact with the subject. The scope of the thesis is presented, addressing both the context of the main problem and the relevance of the subject. The main problem of this dissertation is also described as well as the goals behind a possible solution. An approach to the problem is suggested along with its contributions. Finally the general structure of the remaining document is explained, and may be used as complement to the table of contents.

1.1 Context and Motivation

The field of information visualization focuses on techniques to represent data, as a means to represent knowledge. For example, a simple flow chart can represent a business transaction, or a complex 3d visualization can represent the internet. The success of these techniques spurred its use for analytical purposes, creating visual analytics. The field of visual analytics, focuses mostly on data that is too complex to analyze due to either its size or the many relations existing in it[KPS05]. In order to break this wall of complexity, tools in this field offer an array of analytical tasks able to manipulate huge data sets with ease. The visualization itself serves as an instrument for the analysis: the user interacts with what he sees in order to see what he wants. An analyst, just by looking at the data representation, is able to see some of its properties, like its distribution or most frequent values. These tools serve as support for decision making and, as such, require high interactivity capabilities. This interactivity allows the analyst to explore the data as he sees fit. He can view the data in a different perspective, change its representation or get more detailed information. The tools also allow the analyst to explore what-if conditions or hypothesis.

With the gathering of geospatial data in different research fields, like biology, telecom, transport, social networks, etc., there is a need for analytical tools capable of dealing with spatial data. This motivated the community to create the field of Geovisual Analytics *"as the research area that looks for ways to provide computer support to solving space-related decision problems through enhancing human capabilities to analyze, envision, reason, and deliberate„*[AAJ+06].

Tools in this area deal with spatiotemporal data resorting mostly to thematic maps, which will be explained in section 2.2. Thematic maps show the information over topographic maps and very much like other visual analytics tools, users can interact with the map by zooming, panning or filtering data. There are many kinds of thematic maps, and these are great for analytical tasks in space related problems. However, only a few of these maps, like multi-maps or maps with graphics, address spatiotemporal problems, but handle them rather poorly and almost ignore the temporal aspect of data.

In the scenarios covered by geovisual analytics, besides the spatial information there is also a strong temporal component, that is almost disregarded. There is a lot of potential knowledge that can be gained by analyzing these data. Some solutions have been proposed, as will be seen in section 2.4, that rely on techniques like animation or 3D visualization.

These solutions are rather recent and are too specific to the tasks they were created for, which is why there was a need to survey the current proposals, to categorize them and abstract them. This way it will be easier to select the most appropriate technique, when given a spatiotemporal problem. This survey evaluated the solutions in terms of: kind data it supports and kind of analytical tasks it allows. Besides being able to extend the current techniques to broader contexts, the objective of this survey was to faciliate the proposal of new visual analytics spatiotemporal techniques. As a direct result of that research is the proposal of two new techniques: the attenuation & accumulation map and the overlapping spatiotemporal window technique.

This work was done with the support of a Quadro de Referencia Estrategico Nacional (QREN) research project named *Geo Insight Analytics Platform*, in association with Novabase. The objective of the project is the creation of an analytical tool to support decision making with spatial data. Due to the strong temporal presence in this kind of tools, the results from this thesis may be integrated in the platform.

1.2 Objectives

The main objective of this work is the catalog of visual analytical techniques for spatiotemporal, referenced as spatiotemporal techniques. It provides a clear description of a set of geovisual techniques. This is done with a desciription of the uses of each technique by discerning the kinds of data it can analyze, and the information that can be retained by using them.

There are many scenarios with spatiotemporal data, each with its own characteristics and complexities. In this section we describe the various scenarios in order to explain which of them are addressed by this thesis.

Geospatial data are comprised of either objects or events. Events generate single time occurrence in the data, while an object is identified by an id and can generate more than one ocurrence each with the same id. The information for each object or event can be divided into three sets of attributes: spatial, temporal and thematic[KO03]. The spatial attributes deal with all the geometric aspects of an event or an object, such as position, dimensions and shape. The temporal attributes specify the moment associated with the information, as such, it contains the time of occurrence and its duration. Thematic attributes describe non-spatiotemporal information pertaining to the event or object. This information is specif to the scope of the data and may not be present in some cases. There are many cases of object or events, but they can be distinguished mainly by the shape (point, line, area) and whether there are none, one, or more than one thematic attributes. For each of these cases there are different kinds of spatiotemporal phenomena, which are derived from three kinds of changes in objects or events: change of geometric properties (spatial change), appearance or disappearance (existential changes) and evolution of thematic attributes (thematic changes). Different geovisual techniques that deal with spatiotemporal phenomena provide different analytical tasks. However, these techniques and respective analytical tasks can only be applied to some cases and may not analyze all the phenomena present in the data.

With these problems in mind the objectives of this thesis are:

- to classify the different kinds of spatiotemporal datasets.
- to survey the existing geovisual techniques that analyze spatiotemporal phenomena.
- to abstract the analytical operations that can be made on spatiotemporal data and are relevant to decision making.
- to catalog the surveyed techniques in terms of kinds of spatiotemporal datasets it can be used for and analytical operations it can perform.

The scope of this thesis is spatiotemporal data, not just temporal or just spatial data. Concerning the possible spatiotemporal phenomena the focus is a set of basic scenarios, none of which are related to changes in position, also known as moving objects. The complexity of analyzing movement, trajectories, patterns of movement and analyzing speed of moving objects is in itself a whole different field of study [AAB+11] and is beyond the scope of this thesis.

1.3 Approach and Contributions

This section describes the approach that was taken to achieve the objectives proposed in the previous section, followed by the contributions of this document to the scientific community.

The first step was a comprehensive research of techniques already implemented in the scientific community and business world that deal with spatiotemporal data. Some of these techniques are shown in section 2.4 and have been proposed for specific scenarios, so their characteristics needed to be abstracted, in order to apply them to other scenarios. Since these techniques may be applied to the same cases, the abstraction has to facilitate the comparison between all techniques.

The second step was the conception of the catalog, mainly the proposal of two different frameworks. One framework is used to describe spaiotemporal datasets, the data scenarios framework. The other framework is used to describe spatiotemporal analytical tasks. The techniques that were surveyed are described in the catalog with the aid of the two frameworks.

The catalog shows the cases that are not covered by the current techniques, and by focusing on these gaps we were able to propose two new spatiotemporal techniques: the attenuation & accumulation map technique, and the overlapping spatiotemporal window technique.

This thesis contributes with:

- A framework for describing spatiotemporal datasets from an analytical perspective as data scenarios. The application of said framework to a small set of 23 data scenarios that can represent a range of several areas of study, as will be demonstrated in section 3.1.1.

- A framework of analytical tasks over data scenarios, used to describe 9 example analytical tasks found in the literature.

- A survey of geovisual analytical techniques currently available for the analyses of spatiotemporal data, the identification of applicable data scenarios as well as a description of usable analytical tasks.

- The proposal of geovisual analytical for spatiotemporal data: the attenuation & accumulation map technique.

- The proposal of geovisual analytical for spatiotemporal data: the overlapping spatiotemporal window technique.

- The implementation of a proof of concept that illustrates both proposed techniques, with video demos available at: `http://centria.di.fct.unl.pt/star/resources/files/AAMap.mp4` and `http://centria.di.fct.unl.pt/star/resources/files/OSTWindow.mp4`.

1.4 Document Structure

This document is divided into 6 chapters and are as follows:

- Chapter 2 presents the state of the art, it provides an exposition of some analytical techniques for temporal data and provides an explanation of the main concepts of thematic maps. The chapter finishes with a detailed analyses of some geovisual analytical techniques for spatiotemporal data that were found in the literature.
- Chapter 3 describes results of the survey of spatiotemporal techniques along with the proposed categorization of the spatiotemporal context. A formal agnostic definition is presented that is capable of describing many of the analytical situations found in the state of the art. These concepts are then applied to describe the various spatiotemporal techniques found in the literature.
- Chapter 4 introduces the two spatiotemporal techniques that are being proposed in this thesis. The basic concepts of each technique are explained, as well as the analytical capacities of both techniques.
- Chapter 5 details the prototype that was implemented to showcase both of the proposed techniques.
- Chapter 6 finishes the document by synthesizing the work done and presenting possible further work that can be done on the subjects of the thesis.

2

State of the Art

This chapter includes a state of the art for current geovisual analytical techniques, as well as other concepts that are relevant to this thesis. Firstly, time analysis techniques are presented in section 2.1, to see what can be achieved by analyzing time dependent data. Secondly, a brief description of thematic maps in section 2.2, which are the most used techniques in cartography for analyzing spatial data. Finally, a thorough analyses of current geovisual analytical techniques for spatiotemporal data, also known as spatiotemporal techniques, is presented in section 2.4.

2.1 Time Analysis

Techniques to analyze the evolution of attribute values over time (also known as time-series), may be used for extracting information on patterns, causality or trends. This information may be used to help us understand what has happened before and try to predict what will happen next.

Furthermore, due to the high interactivity allowed by today's systems, the integration of other temporal analysis techniques, with the spatiotemporal techniques being proposed, is paramount. A representation for events over time can be used as part of a chart map, as will be seen in section 2.2. It can also be a way to interact with the map by filtering certain time periods, as seen in [AAM+10], where the proposed system uses charts of multiple time-series to detect temporal patterns. When the user chooses a relevant time period the data is filtered by it and shown on the map.

According to [AMM+08], there are three aspects of time-oriented data that need to be considered when choosing a visualization.

Linear time versus cyclic time These are different ways to perceive the order of temporal data. Linear time perceives a sequential notion to the data, starting from the oldest value to the newest. Time can also be perceived as a cycle that is repeated over and over. Cyclic time can represent man-made abstractions like weeks, or financial years. They can also show natural processes like seasons or days.

Time points versus time intervals These are different ways to quantify and compare time. Time points describe instants with the same duration for which the data is valid. On the other hand, time intervals describe periods of different duration for which the data is valid.

Ordered time versus branching time versus multiple perspectives These are the different views on time. Ordered time establishes a sequential order of events. Branching time presents different chains of events to better describe and compare alternative scenarios, and it is associated with prevision or analyzing the road not taken. Multiple perspectives consider different points of view for the same temporal data(considering different time zones).

The following subsection describe various time visualization techniques that address some of the aspects described.

2.1.1 Sparklines

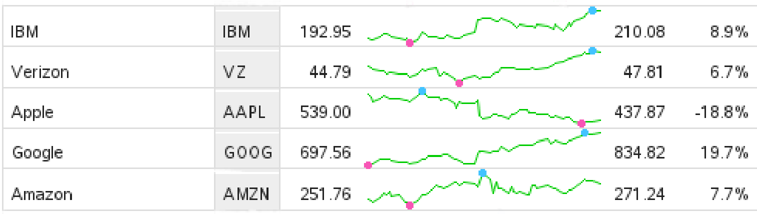
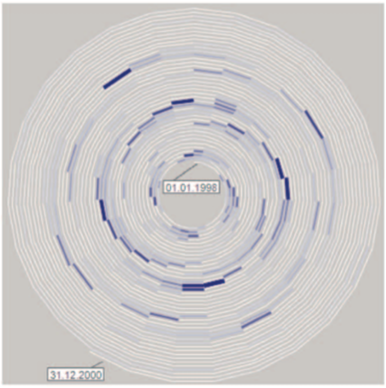


Figure 2.1: Sparklines in multiples depicting stock market indexes.

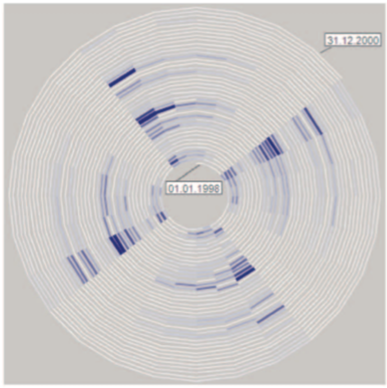
Sparklines are small high definition line charts, without axis or coordinates, that show a linear and ordered notion of time. They can show the variation of an attribute with extreme detail, without providing any quantitative information. These charts were presented by Tufte in [Tuf06] to serve as support for description of values that can vary a lot. Naturally, they are better suited for cases where the variation of a value is as important as its current value, like glucose levels, air temperature or stock market indexes, as shown in fig 2.1.

The sparkline's objective is to represent time-series while always occupying the same space. Given a year long time-series of continuous values, a sparkline can represent the evolution of a single day, week or even the whole year, while occupying the same space. They are often characterized by its creator as word-sized graphics, though the sparklines in fig 2.1 are much bigger than a word, if seen from afar they still transmit the same ideas. This aspect allows the use of several sparklines in what are called multiples, a set of small time-series graphics on the same scale and size, so as to compare the evolution of different values.

2.1.2 Time Spirals



(a) 27 days cycle



(b) 28 days cycle

Figure 2.2: Time spirals depicting the number of influenza cases over a period of three years [AMM+08].

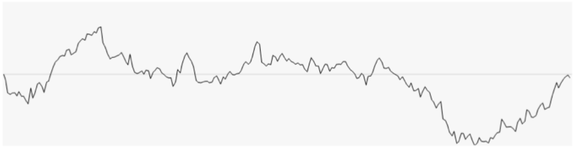
Time spirals were used in [WAM01] to detect repeating patterns that occur in cycles, therefore representing the cyclic aspect of time. The attribute being analyzed is mapped to different levels of saturation for one color. It is then displayed in a radial fashion with a set of circles. Each circle represents one cycle, and the size of the circle explains the ordering of events, the smaller the circle the older it is. As can be seen in fig 2.2(a), just presenting the data this way is not enough. The period of the cycle being viewed may be changed to detect the period of the pattern.

A pattern can be detected when a wedge is formed by similar saturation levels as can be seen in fig 2.2(b). The four dark blue areas clearly depict a pattern occurring every 28 days and each dark blue square depicts a day with a lot of reported influenza cases. It can also be seen, by counting the days between each wedge, that the pattern actually occurs every seven days. Therefore, time spirals do not require the user to guess the correct time period if there is a cyclic pattern; by interacting with the spiral, the user will reach its correct value.

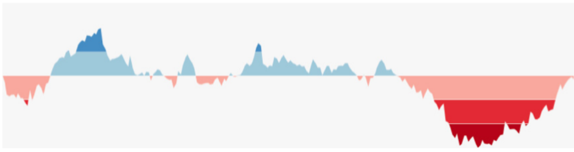
2.1.3 Horizon Graphs

Horizon graphs are compact graphs that are best suited for detecting trends, and comparing values against an objective value, the horizon. To understand horizon graphs one needs to learn how they are made[Few08].

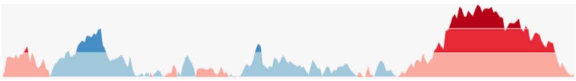
First, starting with a line graph the horizon is drawn (fig 2.3(a)). The area of the graph above the horizon is painted one color, while the area under is painted with another color.



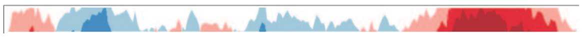
(a) Standard line graph centered around a baseline



(b) Color (blue is positive, red is negative) and layering



(c) Mirroring around the baseline



(d) Wrapping bands into a single space

Figure 2.3: An illustration of the process of creating an horizon graph [JME10].

Then, the values are divided in bands with different saturation levels, the bands further from the horizon have higher saturation (fig 2.3(b)). Next, the values below the horizon are mirrored above it (fig 2.3(c)). Finally, to make it more compact, the bands are all stacked together, showing the higher contrast bands on top fig 2.3(d)). Because of this, when looking at a certain instant in the horizon graph, only the darkest color matters, and its outline defines the attribute's value.

When analyzing horizon graphs, it is really easy to spot trends. Areas with darker color stand out more, and identify high peaks, if painted dark blue or low peaks, if painted dark red. In the same way, areas with less color identify less interesting areas. Long stable periods of above or below horizon values can be identified by continuous red or blue areas. Also, unstable time-series that change often below and above the horizon, can be spotted in graphs with intermittent patterns of red and blue.

Like sparklines, when using the same scale, horizon graphs can be presented in small multiples to compare the values from different time-series.

2.2 Thematic Maps

Although current geovisual analytic systems take advantage of modern computing capabilities, they still rely on legacy techniques from conventional cartography. One of those techniques are thematic maps, maps which describe data by focusing on its main aspects, presenting a theme. There are many kinds of thematic maps, each represents the data differently and each can be used for different analytical tasks.

A thematic map is composed of three different kinds of elements. There is the base

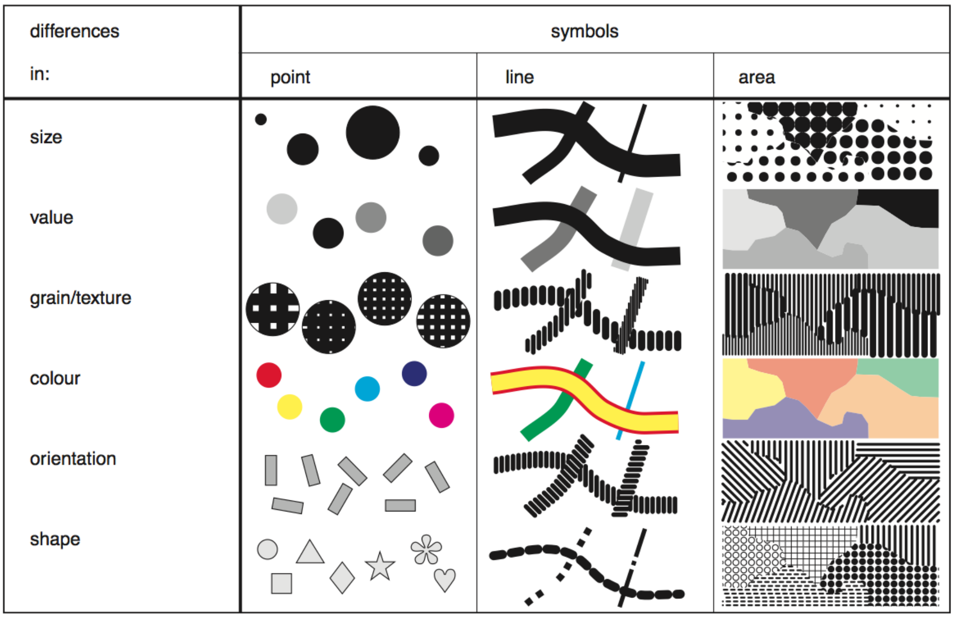


Figure 2.4: Visual variables proposed by Bertin [KO03].

map, which is the topographic representation of the area of interest and serves as a connection between the spatial attributes and the other attributes. As a way to contextualize the data in the thematic map we have the auxiliary elements, like the map legend or title. Finally, over the base map is the thematic layer. A thematic layer is what represents the thematic information, the other non-spatial attributes, and is what defines the type of thematic map. So, for each object in the dataset, its spatial properties, like position or size, are represented on the base map, while its attributes are represented on the thematic layer. Each attribute value has a visual representation for it, and by varying certain visual aspects of said representation, like color or saturation, we can discern the different values. The various visual aspects that can be used were presented by French cartographer Bertin as the visual variables of information visualization [Ber83], which are: size, saturation, color, texture, orientation and shape. The position variable is not used in thematic maps because it removes the spatial information from the map. However, as will be seen in section 2.4.3, the position can be used to identify the time position of an object. The various visual variables can be seen in fig 2.4 applied to the spatial objects that exist in maps.

The basic idea behind visual variables is to map an attribute value to a visual aspect of the object while keeping the other visual aspects constant for all objects. For instance, the thematic map in fig 2.5 shows the variation of one attribute. The attribute is mapped to saturation levels and the map legend describes the attribute values for each range of saturation levels. All the other visual aspects for each region are the same: all have the same texture, color, etc.. A thematic map's interpretation is directly related to the choice of visual variables used, with different variables completely changing the analysis of a map. The choice of visual variables may be limited to the type of spatial object (point,

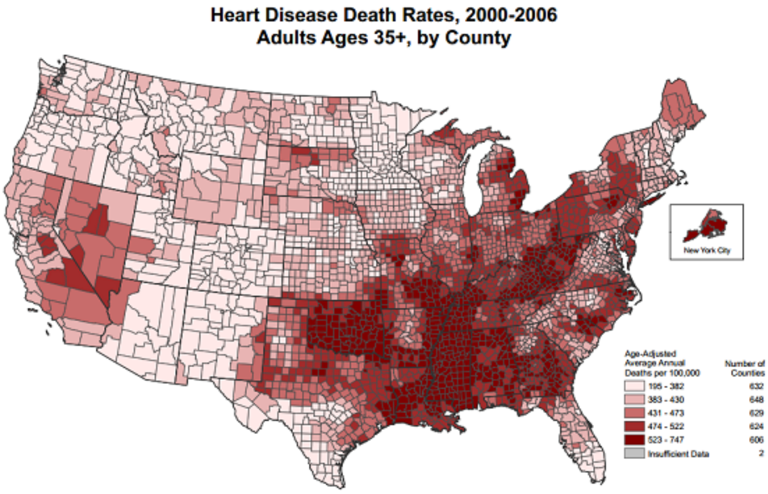


Figure 2.5: Thematic map representing heart disease death rates in the USA

line or area). For instance, if representing an area, the size or orientation variables can not be used. The choice may also be limited by the kind of attribute being mapped [KO03] and the kind of perception allowed by each variable [Car03]. After choosing the variable there are a lot of ways that it can be used, thus affecting the impact of the map.

One of the main objectives of thematic maps, and one of its strengths, is the ability to detect spatial patterns, like in fig 2.5, where the distribution of heart disease cases in the USA is presented. Certain spatial patterns can be identified in this map. For example, it can be seen that neighboring regions have very similar values. This can be due to similar conditions that are shared by neighboring regions, like the weather or local diets. The map also shows that northwestern states have less reported cases of heart disease related deaths.

There are also some kinds of thematic maps that can be used to detect spatiotemporal patterns. There are the change maps, which can be used to analyze the changes between two time periods. This is done by mapping one variable to the attribute value and the color variable to the changes, one color represents increasing values and another color represents decreasing values. This can be seen in fig 2.6, where changes are shown for the unemployment rate of Italy between 1983 and 1990. The current unemployment rate is mapped to the saturation level, while the color indicates if it decreased (marked blue) or increased (marked yellow). This technique focuses on showing changes of attributes, but does not show how they evolved. If the analytical task requires to analyze the evolution of an attribute, its variance may be mapped instead.

Another kind of thematic map for spatiotemporal patterns is the chart map. These maps represent regions and for each region there is a chart on the thematic layer to represent the value of the attributes [KO03]. Several kinds of charts may be used, but for spatiotemporal patterns it has to be charts that show the evolution of attributes over time, like line charts, or area charts. This technique may serve as a good overview of the data, but it lacks the detail required by certain analytical tasks [AAG03].

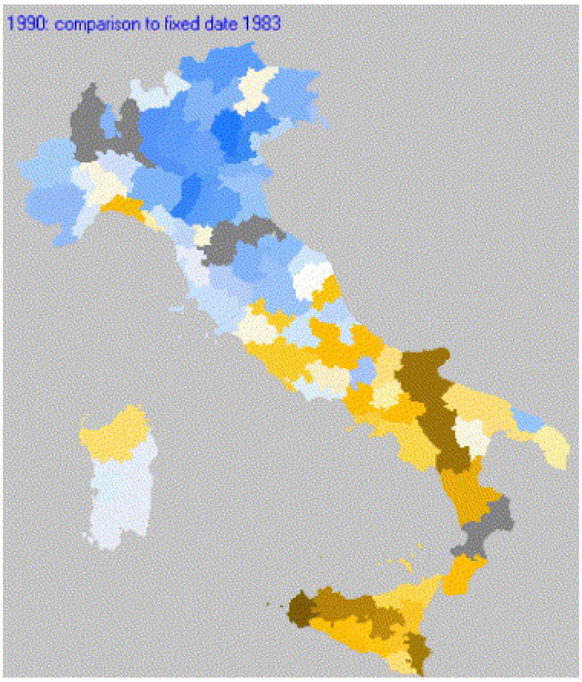


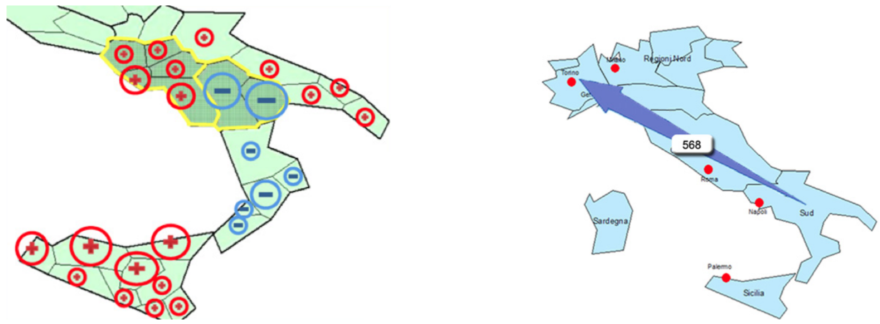
Figure 2.6: Change map for unemployment rates in Italy, between 1983 and 1990 [AAG03].

To conclude, thematic maps are useful tools for exploratory tasks when looking for spatial patterns. Still, when it comes to spatiotemporal patterns, they are limited to overview techniques, which only allow the detection of overall spatial patterns. In order to have a more detailed analysis of spatiotemporal patterns other techniques must be used.

2.3 Chorems

In order to better explain complex spatial phenomena, french cartographer Brunet created *chorems* and the *chorem* diagrams. A *chorem* is a simple representation of a complex concept, object or event. A *chorem* diagram can be drawn using just *chorems*, representing what Brunet calls a complex argument. The idea was to define a small vocabulary of *chorems* capable of presenting any phenomena. With this in mind, Brunet proposed 28 elementary *chorems* which, he argues, are enough to characterize any phenomena. These elementary *chorems* are derived from the combination of 4 spatial objects(point, line, area and network) with 7 geopsatial dynamics(like regional infrastructures or hierarchies). For instance, a line as a regional infrastructure may represent a power line.

The *chorem* approach has been used for creating *chorematic*, man-made maps, where the user has some knowledge of cartography. In [CFL+11], DeChiara et al, created a system capable of transforming a dataset into an appropriate *chorematic* map. Along with the system, they propose a *chorem* classification more appropriate for a geovisual analytical context. These new *chorems* are divided in three main categories which are geographical,



(a) Increase and decrease of population

(b) Migration

Figure 2.7: *Chorematic* map showing phenomenal *chorems* for Italy's population [CFL+11].

for regions and countries, phenomenal, for spatiotemporal phenomena, and annotation, for labels.

The geographical category is comprised of *chorems* that simplify spatial objects removing the unnecessary details of topographic maps. For instance, a detailed cartographic representation of Italy may be replaced with a simple boot shaped outline, or even a circle, depending on the detail required to present the argument. The annotation *chorems* are for labels and remarks that are added to the map. The phenomenal *chorems* describe spatial, temporal and spatiotemporal events and, of the three kinds of *chorems*, are the most relevant to the subject of this thesis. A phenomenal *chorem* can be a simple plus sign to represent increase in population as seen in fig 2.7(a), or an arrow to represent migrations as seen in fig 2.7(b). Both *chorems* are representing the same dataset, yet they represent different phenomena. The use of the phenomena *chorems* is not restricted to *chorematic* maps because the same symbols can have almost the same effect over a topographic map. By using the same symbols as the phenomena *chorems*, maybe a vocabulary of common event representations can be adopted. With these symbols, it would be easier to spot patterns in different maps, or with different datasets, since the user already knows the patterns he is looking for. On the other hand, the techniques shown in other sections rely on a combination of saturation, size and position of objects to depict patterns, which has a huge number of possible visual patterns. Even considering that the person can easily distinguish these visual patterns, the same patterns can have completely different meanings with different datasets, while a symbol is more descriptive of a phenomenon and its design conveys better information.

2.4 Spatiotemporal techniques

2.4.1 Sequence of Maps

In cartography, the simplest way to analyze evolution is by making a map for each time period, and placing them side by side. One solution adopted in current systems is based on this idea [KO03]. The system draws one thematic map for each time period, reflecting all the user's actions, like panning or zooming on all of the maps.

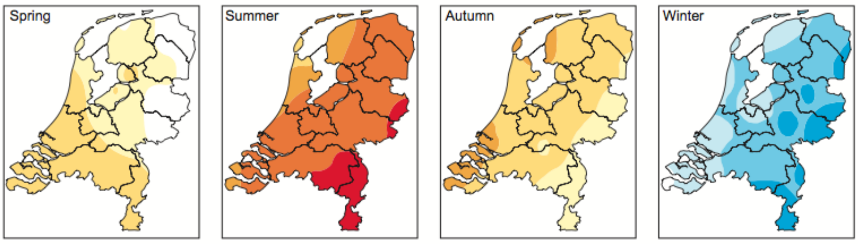


Figure 2.8: A sequence of maps displaying air temperatures for each season [KO03].

From an analytical standpoint this technique is very weak, since by showing one map per time period it can only analyze a limited set of periods. Furthermore, at low zoom levels, its impossible to compare values from the same place. On the other hand at very high zoom levels, the information for that area is easy to compare, but a simple chart would be more appropriate. Another problem with sequencing maps comes from the continuity of the data. When the data is continuous, which can happen with small time periods like days, the sequence of maps may work. However, with discrete intervals, like years or months, the other attributes can change too dramatically. This will cause the maps in the sequence to look completely different and will not give a sense of continuity and evolution between the various maps. This can be seen in 2.8, where each map represents the temperature for each season of the year, although it is easy to see which areas are warmer than others in each season, it is hard to see how the temperature changes over the year.

2.4.1.1 Animation

The use of animation as a technique for representing evolution of things over time is often used in various areas. Even in information analytics it is used in some cases, like the gapminder tool [Ros]. For classic Geographic Information System (GIS) tools, animation is adopted simply as the creation of a video [KO03]. Other solutions in geovisual analytical tools use an animation built by ordering thematic maps following a linear sequence of events. The system provides a slider so the user can play the animation as fast as he wants, or just loop it in certain intervals [AAG03].

According to Robertson et al in [RFF+08], animation is not good for analytical tasks the animation distracts the users and it is hard to detect patterns or trends. However, with the help of someone guiding the user, animation is quite good for presenting data. Unfortunately, the problem of representing discrete time periods that was present in the sequence of maps technique in section 2.4.1, gets even worse with animation. The obtained effect is an animation with a very low frame-rate which might confuse or tire the user. The solution would be in applying a smooth effect between transitions, but this may lead to the display of erroneous information to the user.

2.4.2 Growth Ring Maps

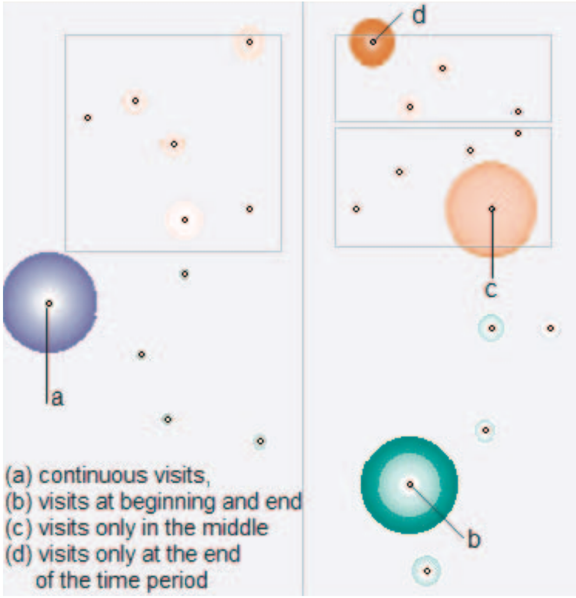


Figure 2.9: A growth ring rap depicting the sensors logs in the cage [BMJK09].

The growth ring maps were presented in [BMJK09] as a means to represent logs from sensors that were placed inside a small rat cage, over a period of several months. There were three types of sensors and each entry in the logs represents an instant a rat activated the sensor. The idea of the growth ring maps was to be able to present the number of entries in the sensor, the type of sensor and the age of each entry, while showing the position of the sensor in a 2d map.

As can be seen in fig 2.9, each circle represents a sensor, the center of the circle marks the position of the sensor. Each pixel in a circle represents an entry in the log. The entries are sorted by age and placed starting from the center in a circle layout. So, the position in the circle represents the age of the entry compared to other entries in that circle. However, to represent the age of an entry, with respect to the whole map, this technique uses the level of saturation. Lighter values represent older entries, while darker values represent younger entries. Since each entry is followed by other entries with the same age, the result is a radial gradient. The function that maps the age to the saturation level can be changed, thus giving more relevance to certain time periods. For instance, by using a linear function, all time points are given equal relevance(as shown in fig 2.9), but, if a discrete function is used, it is only possible to determine certain time periods(as shown in fig 2.10).

In the end, a growth ring can represent different trends depending on its radial gradient. A continuous gradient represents activity throughout the whole time-series, while on the other hand, a circle with a constant color shows only activity in a certain time period. On fig 2.9, four different patterns can be easily discerned and are marked with letters *a* to *d*. The sensor marked with *a* is a continuous radial gradient which symbolizes a point that had continuous visits throughout the experiment. Point *b*, on the other hand, has a gap on the gradient which denotes visits only at the end and beginning of the experiment. The faded color pattern in point *c* signifies a lot of visits in the middle of the experiment. Finally, the dark colored circle in *d* is a sign of visits only in the end of the experiment.

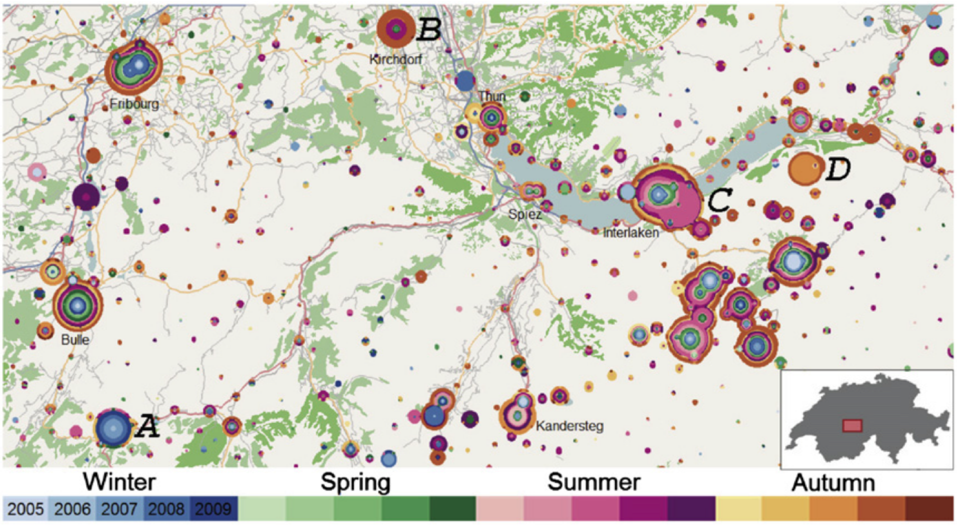


Figure 2.10: A growth ring map showing Flicker photos taken between 2006 and 2009 in Switzerland [AAB+11].

Using the growth ring maps, Andrienko et al in [AAB+11], analyzed Flicker photos taken in Switzerland over a period of five years. Instead of using continuous values for the saturation levels, they used discrete values, one for each year in the time-series. The color represents the season of the year which the photo was taken in. A growth ring is composed of photos taken close to the ring's location. Because each ring is sorted, by seasons and then by year, it is easier to detect cyclic patterns that occur in a certain season and it is also easier to find areas where there was activity in a certain season. A part of the map generated in [AAB+11] can be seen in fig 2.10.

Four different patterns, which were confirmed by collecting words from the photo titles in each area, can be discerned in this map and are marked with letters from A to D. The A ring shows a strong activity during the winter, due to a festival that occurs every January, while the activity in the B ring is due to a vintage car cemetery that was advertised on the flicker website in mid 2008. The ring C shows a large activity in the summer of 2007, due to a Red Bull air race that took place in that time. Finally, ring D's activity in autumn 2007 is because of another air show. It can be hard to identify the year by saturation level, but, since this is an analytical tool and not a static image, simply hovering over an area will identify the year.

As has been shown, a growth ring map has a lot of potential. However it still has not been used much mainly due to its parameters, the choice of colors, the function to map the gradient and the aggregation to be used. These three parameters have a direct impact on the analytical tasks.

Still, changing the parameters themselves can be considered an analytical task, as each set of possible values will denote a different aspect of the data. This technique cannot be used in datasets that are too dispersed over the map, since, with few occurrences per area, the map will not form the rings. In case there are too many events per area, the map will turn into a weird heat map, without any rings to define the points of origin.

In conclusion, the growth ring map can be used effectively with datasets that extend over a long period and that have few spatial positions. The technique focuses on showing existential changes, but it can still analyze attribute changes of one attribute for each event using color. Alternatively, the color can be used to represent a certain time period like the season the event occurred.

2.4.3 Space-Time Cube

A common solution in many areas that need to visualize spatiotemporal information is to use 3D visualization by mapping the position to two dimensions and reserving the third dimension for time. In geovisual analytics and cartography this technique, called the space-time cube, was proposed by Hägerstrand in [Hä70] as a graphical way to show people and the way they interact with each other in time and space. Back then, computer systems could not yet deal with those kinds of techniques, Kraak et al in [Kra03] reintroduced the concepts of the space-time cube for geovisualization systems, and in [GAA04] it was proposed the use of the space-time cube as a geovisual analytics technique.

Firstly, a brief description of the technique. At the core of this visualization is the cube, which is in fact a rectangular prism. At its base is the geographic map of the region, while the height represents the time period and the dataset is shown inside the prism. Points that are close to the base are older than the ones that are further up. Inside the prism there can be a lot of different objects depending on the way the technique is being used. If there is a point inside the cube, it represents an instant in that specific time and place as can be seen in fig 2.14. Lines can represent movement when they are downward diagonal as can be seen in fig 2.11, or can represent prolonged stays in one place, if they are vertical lines. Three dimensional solids inside the cube may represent regions in the map.

Three dimensional navigation can become quite complex and confusing for the user, so each of the space-time cube cases studied provided some features that simplify the analytical process. Each of these cases show different scenarios where the technique is applicable and how to apply it.

The first scenario focuses on showing the variation of two attributes for stationary objects. This use for the space-time cube was presented in [TH10] as part of a geovisual analytical tool. Basically, each point is represented by a column of disks, where each disk represents a time period and the width can portray one attribute, while the saturation level of the disk can be used for another. As can be seen in fig 2.12, a color scheme of red

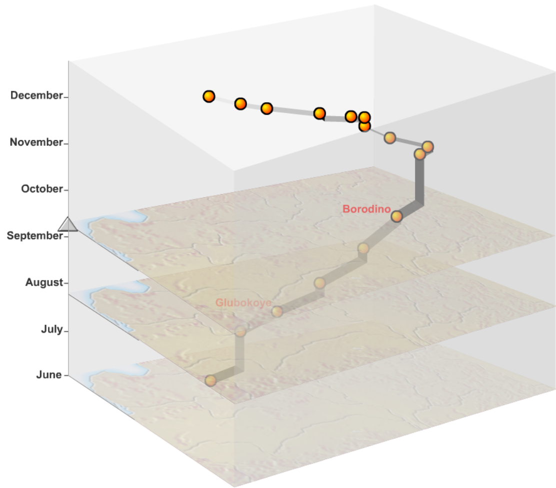


Figure 2.11: Space-time cube visualization applied to Minard's map of Napolean's russian campaign [Kra].

and blue can be adopted, relying on the same concept established by the horizon graphs. In order to analyze the data, the tool allows users to filter data by choosing values of interest from the gradient graph found in the lower left corner. This tool also allows users to do temporal aggregations by collapsing monthly data into yearly averages, which is basically shrinking the columns' height. Spatial aggregations are also possible, columns that are inside the same administrative region are collapsed into one with average values, which results in a reduced number of columns inside the cube. Both kinds of aggregation cause a loss of detail but, on the other hand they create a better overview of the data. Another way the tool simplifies the 3D visualization is by plotting the time-series data on one stacked graph per disk column. Although this technique discards the spatial aspect of the data, it makes it is easier to analyze thematic changes. A similar scenario

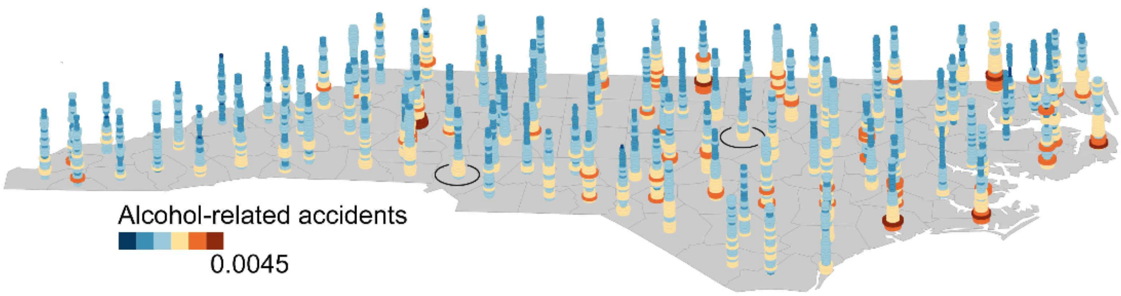


Figure 2.12: Time-series data presented in 3D columns [TH10].

was presented in [TSWS05] where the space-time cube technique was used to show the variation of six attributes for stationary objects. As can be seen in fig 2.13, this is done by mapping the time-series of the attributes to a pencil shaped object. Each side of the pencil represents one attribute and the point marks the object's position. Each attribute is identified by a different color and the saturation represents its value in that time period.

The second scenario focuses on showing event data with one attribute, by depicting each event with one point inside the cube. The point's size can be mapped to an attribute, but, since it might clutter the visualization, it is advised to use the point's color instead [GAA04]. Unlike the previous scenarios, it is hard to discern a point's location both in time and space. In order to facilitate that task, when an user hovers over a point inside the cube, a map is drawn in that level. That way the point's position is shown on the map and on the time axis, while points are outlined if they share the same time period as the hovered point. The way the points are displayed can reveal a lot of patterns, for instance, a vertical line of points will depict repeating pattern. This technique shows a rather simplistic view of the data, there is no aggregation and each point is specific to one row of data. This limits the technique when being used with big datasets. Also the camera angle may mislead the user, either by hiding patterns or by showing patterns that do not exist.

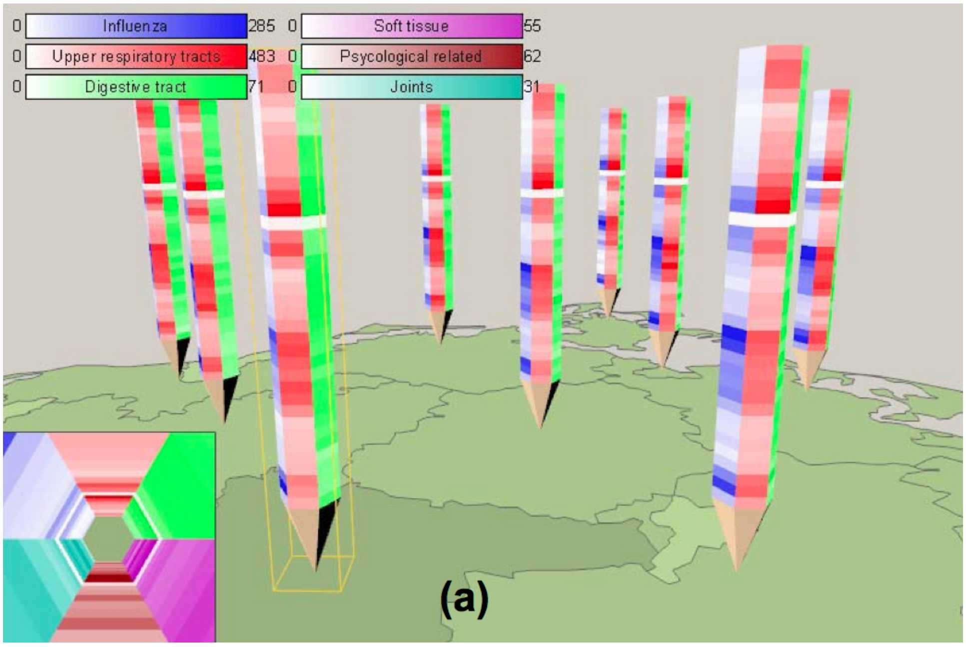


Figure 2.13: Pencil objects in a space-time cube to represent the number of documented cases of six different diseases, in Germany [TSAA12].

The final scenario focuses on presenting stationary lines with one attribute. The scenario was presented in [TSAA12], as a way to analyze trajectory data. It was used with three datasets: traffic jams in San Francisco, radiation measurements in Japan and vessel traffic in the harbor of Brest. The result of the technique, when applied to the traffic jam, is shown in fig 2.15. Each road is depicted by a sequence of segments, each with an attribute that represents the average speed. When analyzing a specific road, a vertical surface is shown representing its state over time, which is the result of stacking the information of the various time periods together. This technique shows detailed spatial information of

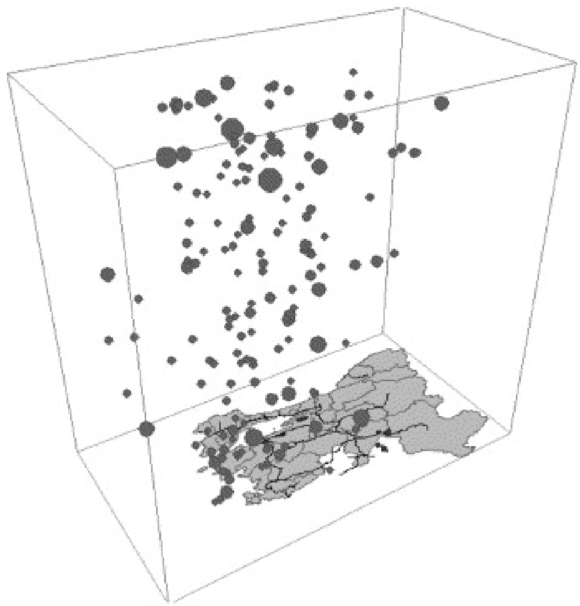


Figure 2.14: Space-time cube of events [AAG03].

the spatial object being analyzed. In the example shown it is possible to see which turns and areas are more congested, because we can see the roads shape. However, it is not possible to do any kind of comparison between this road and nearby roads. It could be an interesting analytical task to see how the various roads fare during rush hour or other interesting time periods.

To conclude, the use of the space-time cube brings some complications inherent to 3D visualization. When dealing with simple objects or events it may be a good solution, even if it does not cope well with big datasets. If analyzing more complex objects, such as lines, the space-time cube becomes too complex and confusing, being only usable with a small number of spatial objects.

2.5 Conclusion

In conclusion, this chapter presented several aspects of the state of the art for geovisual analytics. The key concepts for the visualization of temporal data and spatial data were explained along with some visual analytical techniques that can be used for both kinds of data. A set of geovisual techniques that can be used to analyze spatiotemporal data was presented. These spatiotemporal techniques have distinct basic concepts, in terms of visualizing spatiotemporal data or interpreting spatiotemporal data, these differences in the techniques is what leads to the detection of different spatiotemporal patterns.

These techniques are part of the same context and some are used for very similar situations, however, it is hard to compare them and reapply them to different situations, this is due to the fact that, when created, most techniques were applied to very specific situations. There is a need to abstract the spatiotemporal scope, and reevaluate each

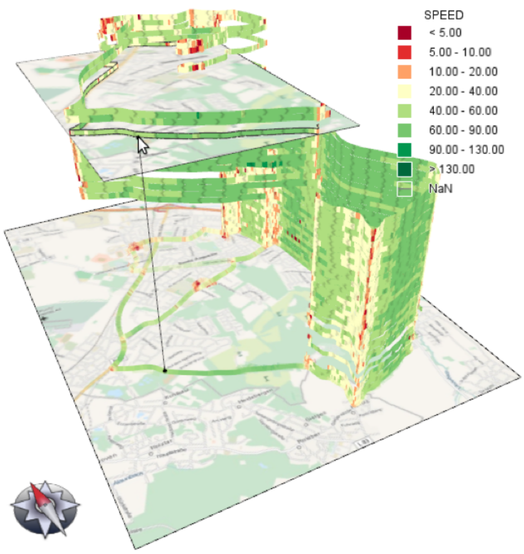


Figure 2.15: A space-time cube representation of traffic information in San Francisco [TSWS05].

technique in a broader sense, so they can be applied to other situations.

3

Catalog

The following sections of this chapter constitute a spatiotemporal catalog. There are four main components of the catalog each with its own section.

Section 3.1 describes the data scenarios component of the catalog. A data scenario is a model that can be used to describe a group of datasets. The concept is described followed by the characterization of several data scenarios for spatiotemporal datasets. The second component of the catalog, described in section 3.2, proposes a framework that can be used to formulate analytical tasks for spatiotemporal data. The framework proposal is followed by the description of some analytical tasks.

The spatiotemporal techniques component in section 3.3 describes the surveyed techniques by: (1) defining the data scenarios where they are applicable, (2) defining the analytical tasks they are suitable for. Finally, section 3.4, provides a general overview of the whole catalog by exposing the connections between the three components.

3.1 Data Scenarios

This section is about the data scenarios of the catalog. The data scenario concept and specification are explained in section 3.1.1 and in the rest of the sections of this chapter, sections 3.1.2 to 3.1.24, the various data scenarios that were considered for this catalog are described.

3.1.1 Description

A spatiotemporal data scenario is a simple specification that can be used to characterize spatiotemporal data, particularly spatiotemporal datasets. A dataset in its simplest form is a table of data, where each column represents a particular attribute, and each entry of the dataset is represented by a row. Each entry in the dataset assigns a value to each attribute. The data scenario specification focuses on a dataset's metainformation, mainly on the kinds of attributes in order to classify a dataset as one data scenario.

To explain the various aspects of spatiotemporal datasets, two examples are considered. The first example dataset is a collection of accidents that occurred in a certain country. The attributes of the dataset are: (1) the coordinates where the accident occurred; (2) the instant it occurred; (3) the instant it was reported; (4) the instant the police arrived at the scene;(5) the cause of the accident and (6) the number of victims. The second dataset is a collection of store information taken daily in a commercial area of a certain city. The attributes of this example dataset are: (1) an identifier unique to each store; (2) the area occupied by the store; (3) the day the entry information was collected; (4) the total revenue collected in that day and (5) the kind of store (clothing, food, electronics, etc. . .).

In a spatiotemporal context, a dataset's attributes can be divided in four different categories: spatial attributes, temporal attributes, id attributes and thematic attributes. A spatiotemporal dataset has at least one spatial attribute and one temporal attribute.

Spatial attributes refer to any physical property of each entry of the dataset, be it a coordinate or various coordinates forming a shape or even the height of an entry. In the accidents dataset example, the coordinates where the accident occurred are a spatial attribute. For the stores example, the area of the store, which is comprised of several coordinates to mark its boundaries, is a set of spatial attributes.

The temporal attributes represent any aspects that refer to time on the dataset like the time the entry was collected or an interval in which the entry is valid. For example, in the accidents dataset, the time the accident occurred, the time the accident was reported and the time the police reached the accident's location are all temporal attributes. Considering the stores dataset, the day the sales information refers to is a temporal attribute.

The id attributes are used to identify separate entries in the dataset that refer to the same entity. In the stores dataset example, an id identifies the store where a purchase was made. For the accident's dataset there are no id attributes as the same accident does not occur more than once.

The last group of attributes, the thematic attributes, provide information on the dataset that is domain specific and provides a *theme* to the data. A thematic attribute can be further classified as either a quantitative attribute or a classification attribute. A classification attribute is an attribute that characterizes an entry making it a part of some class of entries while a quantitative attribute is one that is measurable. For the accidents dataset, the cause of an accident is a classification attribute, it fits the accident in a class of accidents, and the number of victims is a quantitative attribute. In the stores dataset example, the daily revenue of a store is a quantitative attribute, while the kind of store is a classification attribute.

3.1.1.1 Scope

In terms of spatial attributes, only the position and shape of an entry are considered. Furthermore, the spatial attributes can be reduced to just the description of the shape of an entry as its shape implies its position. An entry's shape can be a point, a line or an area. A point is when an entry is only a point in space or the shape of the entry is not relevant for the analyses so it can be ignored, with the position representing the center of the shape. In this case, one pair of coordinates that positions the point in space describes an entry's spatial properties. A line is when an entry describes a path that connects two points, the path can have any shape but its area is ignored, so an entry's spatial properties are described by a set of coordinates that form a line. An area is when an entry has a shape with an area relevant for its analyses; in this case an entry's spatial properties are described by a set of coordinates that form the boundaries of the area.

For temporal attributes only instant attributes are considered, no intervals, and only one instant per entry. An instant has a specific length of time: it can be a day, a month, a year, etc..., depending on the dataset.

The id attribute is used to separate the data scenarios into two main categories, the event scenarios (no ids) and the object scenarios (with ids). An event is equivalent to a single time occurrence and an entry in the dataset describes that event. While an object is something that changes its state, or rather its attribute values, over time, and each entry on the dataset describes the state of the identified object at a certain time instant.

For this reason, a dataset that is described as an object scenario can be further discerned depending on the kinds of variation that its objects suffer. Considering the stores dataset example, each store does not suffer any change to its spatial attributes; its area and position do not change. The classification attribute, the kind of store does not change, however the quantitative attribute, total revenue, changes everyday. Other datasets that fit the same scenario as the stores dataset could have different attributes changing. For instance a dataset of fires, where each fire is identified by a unique id, and the state of the fire is collected every hour, the area that is on fire at a specific hour is registered, as well as, the number of firefighters assigned at that hour and the cause of the fire. This example dataset fits the same data scenario as the stores, since it has: (1) an id; (2) a time instant (hour);(3) area shaped objects (burned area); (4) a quantitative attribute (the number of assigned firefighters) and (5) a classification attribute (the cause of the fire). However, the area that is on fire changes shape and the number of assigned firefighters varies as well. This dataset is described by the same data scenario as the stores dataset but is described by a different sub-scenario. A sub-scenario of an object scenario identifies which attributes change over time. A quantitative attribute always change over time; if it does not change then it is characterizing different objects and is therefore a classification attribute. A classification attribute can vary over time or not, in case there is no variation it is used to determine different groups of objects. For point shaped scenarios, a point's position can change over time in a sub-scenario. In area shaped objects scenarios there can be a change in an object's position and its shape.

3.1.1.2 Specification

The simplest way to refer to a data scenario is through its attributes. With this in mind a convention was adopted to refer to the various data scenarios using the notation ([id] ,S ,T ,[a_i ,[a_j]]), where id denotes the id attribute, S stands for spatial attributes, T for temporal attributes and A thematic attributes, the brackets denote optional attributes.

The id is the first to be specified so it is easier to discern object scenarios from event scenarios. The temporal attribute is represented by t while the spatial attribute is represented by: A for area, P for point and L for line. For temporal attributes, as was discussed, only the instant is considered so the temporal attribute is always t. For thematic attributes there can either be one attribute (a_i), two attributes (a_i,a_j) or no attributes, where the symbol a_q is used to denote a quantitative attribute and a_c is used to denote a classification attribute. To represent the various sub-scenarios a **bold** font weight is used to represent the attribute that is changing in that sub-scenario, for instance, in the (id, P, t, a_q) scenario, the sub-scenario for a quantitative attribute changing over time is denoted by (id,P,t,$\mathbf{a_q}$). Any sub-scenario where there is a change in physical attributes over time is represented by marking the physical attribute. This means, for point shaped object scenarios, a sub-scenario where the position is changing over time uses the symbol **P**; while for area shaped objects any change either in position or shape is considered the same case and is denoted by **A**.

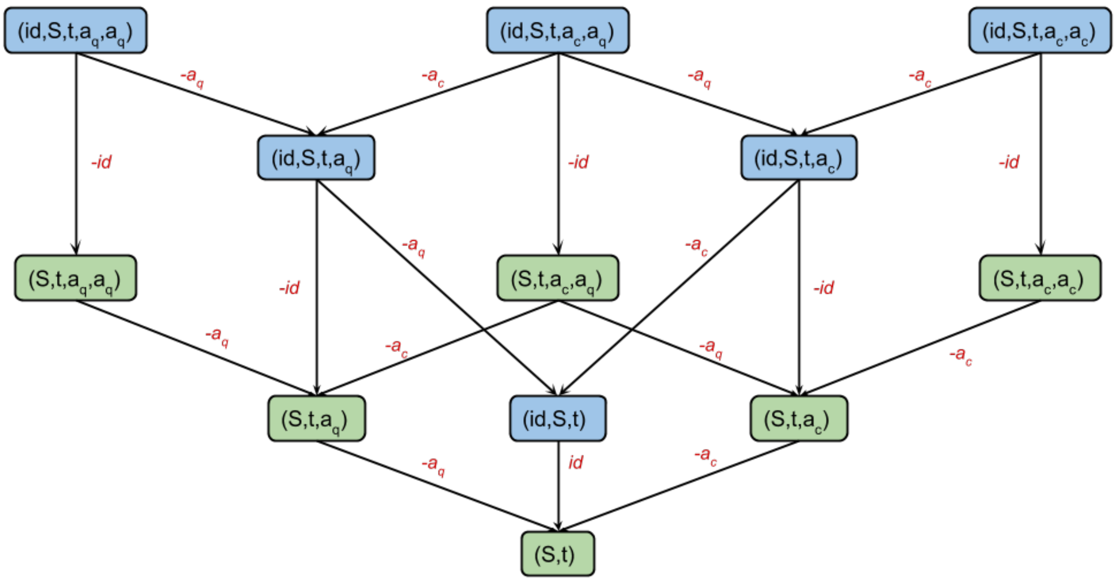


Figure 3.1: An abstraction of the various scenarios and how they are related by ignoring certain attributes.

Considering the same example datasets of section 3.1.1, the accidents dataset fits in the data scenario: (P,t,a_c,a_q), because an accident's position can be represented by just one point, either the instant the accident was reported or the instant the accident occurred can be represented by t, and the cause of the accident is a classification attribute and the number of victims a quantitative attribute. If certain attributes are discarded the dataset can fit other data scenarios, for instance, if the number of victims is ignored then the dataset fits the data scenario (P, t, a_c), furthermore if the cause of the accident is also ignored the dataset fits the data scenario (P, t). This concept is sumarized by figure 3.1, showing all the relations between scenarios of a certain shape S. Considering the stores dataset it fits the data scenario (id, A, a_q, a_c), where each store is one object identified by an id, its position is an area A, and the quantitative attribute and classification attribute are represented by a_q and a_c respectively. More specifically, since a store does not move, a kind of store does not change, and considering it does not expand or retract its area; the store dataset falls in the sub-scenario $(id, A, \mathbf{a_q}, a_c)$.

3.1.1.3 Overview and Limitations

The data scenarios that are considered in this catalog were derived from a list of candidate data scenarios that was generated by making all possible combinations of a spatial shape (P,L,A), id or no id and the possible combinations of none, one or two thematic attributes. Each of these candidate data scenarios was compared with examples in the literature and assigned credible examples of dataset. Any candidate that did not fit these criteria was discarded from the catalog. The maximum of two thematic attributes was imposed because it limits the list of candidate data scenarios, besides, with more than two thematic attributes the focus shifts from spatiotemporal patterns to thematic patterns.

Any possible line or area shaped events were not considered, although some examples of datasets that fit these scenarios exist, none were found in the literature, there are no techniques that are used for these kinds of datasets. Examples like fires or radiation clouds can be perceived as area shaped events, but because these are not single time occurrences they are considered as objects. The scenario of just line objects was not considered because no usable examples were found in this scope; mainly there are no changes possible for a line's spatial properties, so no line shaped object scenarios have sub-scenarios where there is a variation of its spatial attributes.

Considering the limitations presented in the previous paragraphs the scenarios considered in the catalog can be categorized in four major groups, point shaped events, point shaped objects, line shaped objects and area shaped objects, making a total of 23 scenarios. The point shaped events are in sections from 3.1.2 to 3.1.7, which cover all scenarios pertaining to point shaped events, like phone calls, with one or two quantitative or classification attributes.

The point shaped objects are in sections from 3.1.8 to 3.1.13, which cover all scenarios pertaining to point shaped objects, like buses or stores, with one or two quantitative or classification attributes. The line shaped objects are in sections from 3.1.14 to 3.1.18, which only cover scenarios with one or two thematic attributes, like roads with number of accidents. The area shaped objects are in sections 3.1.19, 3.1.20, 3.1.21, 3.1.22, 3.1.23 and 3.1.24, these cover all scenarios concerned with area shaped objects, like countries, with one or two quantitative or classification attributes. The following sections characterize the different data scenarios using the notation specified on section 3.1.1.2. Furthermore, a description is provided for each scenario followed by examples of datasets that fit that scenario. In the case of object scenarios each of the sub-scenarios is described separately and provided an example.

3.1.2 Point shaped Events

Shorthand	Type	Shape
(P, t)	Event	Point

This scenario covers the point shaped events. For example, a dataset of fires where the coordinates of the starting point of the fire are represented by a point, and the instant when it occurred is represented by t. In addition, a dataset of crimes where the coordinates that a crime occurred in is represented by a point, and the instant when it was reported is represented by t.

3.1.3 Point shaped Events with One quantitative attribute

Shorthand	Type	Shape
(P, t, a_q)	Event	Point

This scenario covers the point shaped events with one quantitative attribute. For example, a dataset of crimes where the coordinates that a crime occurred in is represented by a point, and the instant when it was reported is represented by t. The number of victims is a quantitative attribute. Also, a dataset of fires where the coordinates of the starting point of the fire are represented by a point, and the instant when it occurred is represented by t. The number of the total assigned firefighters is a quantitative attribute.

3.1.4 Point shaped Events with One classification attribute

Shorthand	Type	Shape
(P, t, a_c)	Event	Point

This scenario covers the point shaped events with one classification attribute. For example, a dataset of phone calls where the coordinates of the callers position at the time of the call is represented by a point, and the instant the call was made is represented by t. The payment plan of the caller is a classification attribute. Also, a dataset of fires where the coordinates of the starting point of the fire are represented by a point, and the instant when it occurred is represented by t. The cause of the fire is a classification attribute.

3.1.5 Point shaped Events with Two quantitative attributes

Shorthand	Type	Shape
(P, t, a_q, a_q)	Event	Point

This scenario covers the point shaped events with two quantitative attributes. For example, a dataset of fires where the coordinates of the starting point of the fire are represented by a point, and the instant when it occurred is represented by t. The number of victims and total damage cost are both quantitative attributes.

3.1.6 Point shaped Events with Two classification attributes

Shorthand	Type	Shape
(P, t, a_c, a_c)	Event	Point

This scenario covers the point shaped events with two classification attributes. For example, a dataset of crimes where the coordinates that a crime occurred in is represented by a point, and the instant when it was reported is represented by t. The assigned police station and kind of crime are classification attributes. Another example is a dataset of phone calls where the coordinates of the callers position at the time of the call is represented by a point, and the instant the call was made is represented by t. Both the payment plan and phone model of the caller are classification attributes.

3.1.7 Point shaped Events with One classification and one quantitative attribute

Shorthand	Type	Shape
(P, t, a_c, a_q)	Event	Point

This scenario covers the point shaped events with one classification and one quantitative attribute. For example, a dataset of crimes where the coordinates that a crime occurred in is represented by a point, and the instant when it was reported is represented by t. The assigned police station is a classification attribute and the number of victims is a quantitative attribute. Also, a dataset of phone calls where the coordinates of the callers position at the time of the call is represented by a point, and the instant the call was made is represented by t. The payment plan of the caller is a classification attribute and his credit is a quantitative attribute.

3.1.8 Point shaped Objects

Shorthand	Type	Shape
(id, P, t)	Object	Point

This scenario covers the point shaped objects. This scenario has only one sub-scenario denoted $(id, \mathbf{P}, t)$, which is when the point's position can change over time. For example, a dataset of buses, where the coordinates of a bus are represented by a point and the instant the coordinates were collected is represented by t. Furthermore, a dataset of taxis, where the coordinates of a taxi are represented by a point and the instant the coordinates were collected is represented by t.

3.1.9 Point shaped Objects with One quantitative attribute

Shorthand	Type	Shape
(id, P, t, a_q)	Object	Point

This scenario covers the point shaped objects with one quantitative attribute. This scenario can be divided into 2 subs-cenarios, which are:

- $(id, P, t, \mathbf{a_q})$ - In this sub-scenario there is a variation of the quantitative attribute.
- $(id, \mathbf{P}, t, \mathbf{a_q})$ - In this sub-scenario there is a variation of both the point's position and of the quantitative attribute.

And for each sub-scenario there are the following examples:

- $(id, P, t, \mathbf{a_q})$ - stores, where the coordinates of a store are represented by a point and the instant the store information was collected is represented by t. The total value of sales at a certain t is a quantitative attribute and changes between ts.
- $(id, \mathbf{P}, t, \mathbf{a_q})$ - buses, where the coordinates of a bus are represented by a point and the instant the coordinates were collected is represented by t. The number of passengers is a quantitative attribute and varies over time.

3.1.10 Point shaped Objects with One classification attribute

Shorthand	Type	Shape
(id, P, t, a_c)	Object	Point

This scenario covers the point shaped objects with one classification attribute. This scenario can be divided into 3 subs-cenarios, which are:

- $(id, P, t, \mathbf{a_c})$ - In this sub-scenario there is a variation of the classification attribute.
- $(id, \mathbf{P}, t, a_c)$ - In this sub-scenario there is a variation of the point's position while the classification attribute does not change. The classification attribute can be used to identify specific classes of objects.
- $(id, \mathbf{P}, t, \mathbf{a_c})$ - In this sub-scenario there is a variation of both the point's position and of the classification attribute.

And for each sub-scenario there are the following examples:

- $(id, P, t, \mathbf{a_c})$ - stationary, mobile phone antennas where the coordinates of an antenna are represented by a point and the instant the antenna information was collected is represented by t. Whether an antenna is in maintenance or not is a varying classification attribute.
- $(id, \mathbf{P}, t, a_c)$ - buses, where the coordinates of a bus are represented by a point and the instant the coordinates were collected is represented by t. The model of the bus is a non varying classification attribute that can be used to identify specific buses.
- $(id, \mathbf{P}, t, \mathbf{a_c})$ - taxis, where the coordinates of a taxi are represented by a point and the instant the coordinates were collected is represented by t. Whether a taxi is occupied or not is a varying classification attribute.

3.1.11 Point shaped Objects with Two quantitative attributes

Shorthand	Type	Shape
(id, P, t, a_q, a_q)	Object	Point

This scenario covers the point shaped objects with two quantitative attributes. This scenario can be divided into 2 subs-cenarios, which are:

- $(id, P, t, \mathbf{a_q}, \mathbf{a_q})$ - In this sub-scenario both quantitative attributes vary over time.
- $(id, \mathbf{P}, t, \mathbf{a_q}, \mathbf{a_q})$ - In this sub-scenario the point's position varies along with both quantitative attributes.

And for each sub-scenario there are the following examples:

- $(id, P, t, \mathbf{a_q}, \mathbf{a_q})$ - stores, where the coordinates of a store are represented by a point and the instant the store information was collected is represented by t. A store's total revenue at instant t and its number of clients at the same instant are both quantitative attributes that vary over time.

- $(id, \mathbf{P}, t, \mathbf{a_q}, \mathbf{a_q})$ - buses, where the coordinates of a bus are represented by a point and the instant the coordinates were collected is represented by t. A bus's number of passengers and fuel deposit at instant t are both quantitative attributes that vary over time, while the buses travel their routes.

3.1.12 Point shaped Objects with Two classification attributes

Shorthand	Type	Shape
(id, P, t, a_c, a_c)	Object	Point

This scenario covers the point shaped objects with two classification attributes. This scenario can be divided into 5 subs-cenarios, which are:

- $(id, P, t, \mathbf{a_c}, a_c)$ - In this sub-scenario there is a variation of one of the classification attributes while the other classification attribute does not change. The non-changing attribute can be used to identify objects of a specific class.
- $(id, P, t, \mathbf{a_c}, \mathbf{a_c})$ - In this sub-scenario both classification attributes vary at the same time.
- $(id, \mathbf{P}, t, a_c, a_c)$ - In this sub-scenario the point's position varies while both classification attributes do not change over time. The combination of both attributes can be used to identify objects of a specific class.
- $(id, \mathbf{P}, t, \mathbf{a_c}, a_c)$ - In this sub-scenario the point's position varies while there is a variation of one classification attribute and the other classification attribute stays with the same value over time. The nonchanging classification attribute can be used to identify different kinds of objects.
- $(id, \mathbf{P}, t, \mathbf{a_c}, \mathbf{a_c})$ - In this sub-scenario the point's position varies while there is a variation of both classification attribute attributes over time.

And for each sub-scenario there are the following examples:

- $(id, P, t, \mathbf{a_c}, a_c)$ - stationary, mobile phone antennas where the coordinates of an antenna are represented by a point and the instant the antenna information was collected is represented by t. Whether an antenna is in maintenance or not is a varying classification attribute, the antennas brand is a non-varying classification attribute.
- $(id, P, t, \mathbf{a_c}, \mathbf{a_c})$ - lifeguard stations, where each station's coordinates is represented by a point, and the instant the station's information was collected is represented by t. The color of the station's flag, and whether there is a lifeguard at the station are both classification attributes that vary over time.

- $(id, \mathbf{P}, t, a_c, a_c)$ - animals, where the coordinates of an animal are represented by a point and the instant the coordinates were collected is represented by t. An animal's gender and species are both classification attributes that do not change over time.
- $(id, \mathbf{P}, t, \mathbf{a_c}, a_c)$ - buses, where the coordinates of a bus are represented by a point and the instant the coordinates were collected is represented by t. The maintenance status of a bus, whether it needs maintenance or not, or if it is under maintenance at instant t is a varying classification attribute. The brand of a bus is a non-changing classification attribute.
- $(id, \mathbf{P}, t, \mathbf{a_c}, \mathbf{a_c})$ - cellphones, where the coordinates of the cellphones at instant t are represented by a point. Whether a cellphone is making a call or not is a varying classification attribute, the cellphone data plan is a changing classification attribute as well.

3.1.13 Point shaped Objects with One classification and one quantitative attribute

Shorthand	Type	Shape
(id, P, t, a_c, a_q)	Object	Point

This scenario covers the point shaped objects with one classification and one quantitative attribute. This scenario can be divided into 4 subs-cenarios, which are:

- $(id, P, t, a_c, \mathbf{a_q})$ - In this sub-scenario the quantitative attribute varies over time and the classification attribute does not.
- $(id, P, t, \mathbf{a_c}, \mathbf{a_q})$ - In this sub-scenario both attributes can vary simultaneously.
- $(id, \mathbf{P}, t, a_c, \mathbf{a_q})$ - In this sub-scenario the quantitative attribute and the point's position varies ,while the classification attribute does not, serving as an identifier of objects of that classification.
- $(id, \mathbf{P}, t, \mathbf{a_c}, \mathbf{a_q})$ - In this sub-scenario the point's position varies along with both thematic attributes.

And for each sub-scenario there are the following examples:

- $(id, P, t, a_c, \mathbf{a_q})$ - stores, where the coordinates of a store are represented by a point and the instant the store information was collected is represented by t. The category of the store is a non varying classification attribute and the number of clients is a quantitative attribute.

- $(id, P, t, \mathbf{a_c}, \mathbf{a_q})$ - stationary, mobile phone antennas where the coordinates of an antenna are represented by a point and the instant the antenna information was collected is represented by t. The number of cellphones in range is a quantitative attribute and whether the antenna is in maintenance or not is a varying classification attribute.
- $(id, \mathbf{P}, t, a_c, \mathbf{a_q})$ - buses, where the coordinates of a bus are represented by a point and the instant the coordinates were collected is represented by t. The model of the vehicle is a non-varying classification model and the number of passengers inside is a quantitative attribute.
- $(id, \mathbf{P}, t, \mathbf{a_c}, \mathbf{a_q})$ - taxis, where the coordinates of a taxi are represented by a point and the instant the coordinates were collected is represented by t. The value of the taximeter is a quantitative attribute and whether the taxi is occupied is a varying classification attribute.

3.1.14 Line shaped Objects with One quantitative attribute

Shorthand	Type	Shape
(id, L, t, a_q)	Object	Line

This scenario only has one sub-scenario denoted $(id, L, t, \mathbf{a_q})$which is when there is a variation of the quantitative attribute. For example, a dataset of roads, where the shape of the road is represented by a line. The number of accidents that occurred at instant t on a road is a quantitative attribute.

3.1.15 Line shaped Objects with One classification attribute

Shorthand	Type	Shape
(id, L, t, a_c)	Object	Line

This scenario only has one sub-scenario denoted $(id, L, t, \mathbf{a_c})$which is when there is a variation of the classification attribute. For example, a dataset of roads, where the shape of the road is represented by a line. Whether a road is under maintenance or not is a classification attribute that changes over time.

3.1.16 Line shaped Objects with Two quantitative attributes

Shorthand	Type	Shape
(id, L, t, a_q, a_q)	Object	Line

This scenario only has one sub-scenario denoted $(id, L, t, \mathbf{a_q}, \mathbf{a_q})$which is when both quantitative attributes vary over time. For example, a dataset of migratory paths of birds, where the path between two resting places is represented by a line. The number of birds on a migratory path and its average temperature are both quantitative attributes.

3.1.17 Line shaped Objects with Two classification attributes

Shorthand	Type	Shape
(id, L, t, a_c, a_c)	Object	Line

This scenario covers the line shaped objects with two classification attributes. This scenario can be divided into 2 subs-cenarios, which are:

- $(id, L, t, \mathbf{a_c}, a_c)$ - In this sub-scenario there is a variation of one of the classification attributes while the other classification attribute does not change. The non-changing attribute can be used to identify objects of a specific class.
- $(id, L, t, \mathbf{a_c}, \mathbf{a_c})$ - In this sub-scenario both classification attributes vary at the same time.

And for each sub-scenario there are the following examples:

- $(id, L, t, \mathbf{a_c}, a_c)$ - air routes, where the route is represented by a line that connects both airports of the route, the class of planes in a route is a non changing attribute, while the major airline ocupying that route is a changing classification attribute.
- $(id, L, t, \mathbf{a_c}, \mathbf{a_c})$ - roads, where the shape of the road is represented by a line. Whether a road is under maintenance and the status of the road's traffic are both classification attributes that vary over time.

3.1.18 Line shaped Objects with One classification and one quantitative attribute

Shorthand	Type	Shape
(id, L, t, a_c, a_q)	Object	Line

This scenario covers the line shaped objects with one classification and one quantitative attribute. This scenario can be divided into 2 subs-cenarios, which are:

- $(id, L, t, a_c, \mathbf{a_q})$ - In this sub-scenario the quantitative attribute varies over time and the classification attribute does not.
- $(id, L, t, \mathbf{a_c}, \mathbf{a_q})$ - In this sub-scenario both attributes can vary simultaneously.

And for each sub-scenario there are the following examples:

- $(id, L, t, a_c, \mathbf{a_q})$ - roads, where the shape of the road is represented by a line. The number of cars on the road is a quantitative attribute, while the kind of road is a non changing of classification attribute.
- $(id, L, t, \mathbf{a_c}, \mathbf{a_q})$ - air routes, where the route is represented by a line that connects both airports of the route. The number of passengers that are at instant t in a plane over a route is a quantitative attribute, the airline with most airplanes in that route at instant t is a changing classification attribute.

3.1.19 Area shaped Objects

Shorthand	Type	Shape
(id, A, t)	Object	Area

This scenario covers the area shaped objects. This scenario has only one sub-scenario denoted $(id, \mathbf{A}, t)$, which is when the area's size and shape change over time. For example, a dataset of fires where the area represents the burned area at the instant of capture t. Also, a dataset of Radiation clouds where the area represents the cloud's area at the instant of capture t.

3.1.20 Area shaped Objects with One quantitative attribute

Shorthand	Type	Shape
(id, A, t, a_q)	Object	Area

This scenario covers the area shaped objects with one quantitative attribute. This scenario can be divided into 2 subs-cenarios, which are:

- $(id, A, t, \mathbf{a_q})$ - In this sub-scenario there is a variation of the quantitative attribute.
- $(id, \mathbf{A}, t, \mathbf{a_q})$ - In this sub-scenario there is a variation of both the quantitative attribute and the area's position or shape.

And for each sub-scenario there are the following examples:

- $(id, A, t, \mathbf{a_q})$ - countries, where each countries boundaries is represented by an area. The life expectancy of a country is a quantitative attribute.
- $(id, \mathbf{A}, t, \mathbf{a_q})$ - oil spills, where the area represents the affected area and t marks the time of the reading. The amount of litters of contaminated water is a quantitative attribute.

3.1.21 Area shaped Objects with One classification attribute

Shorthand	Type	Shape
(id, A, t, a_c)	Object	Area

This scenario covers the area shaped objects with one classification attribute. This scenario can be divided into 3 subs-cenarios, which are:

- $(id, A, t, \mathbf{a_c})$ - In this sub-scenario there is a variation of the classification attribute.
- $(id, \mathbf{A}, t, a_c)$ - In this sub-scenario there is a variation of the area's position or shape but the classification attribute does not change. The classification attribute can be used to identify specific classes of objects.
- $(id, \mathbf{A}, t, \mathbf{a_c})$ - In this sub-scenario there is a variation of both the classification attribute and the area's position or shape.

And for each sub-scenario there are the following examples:

- $(id, A, t, \mathbf{a_c})$ - countries, where each countries boundaries is represented by an area. The country's credit rating is a changing classification attribute.
- $(id, \mathbf{A}, t, a_c)$ - fires where the area represents the burned area at the instant of capture t. The cause of the fire is a non-changing classification attribute.
- $(id, \mathbf{A}, t, \mathbf{a_c})$ - volcanic clouds, where the area represents the affected area at a specific t. Whether a plane can fly through a volcanic cloud is a changing classification attribute.

3.1.22 Area shaped Objects with Two quantitative attributes

Shorthand	Type	Shape
(id, A, t, a_q, a_q)	Object	Area

This scenario covers the area shaped objects with two quantitative attributes. This scenario can be divided into 2 subs-cenarios, which are:

- $(id, A, t, \mathbf{a_q}, \mathbf{a_q})$ - In this sub-scenario both quantitative attributes vary over time.
- $(id, \mathbf{A}, t, \mathbf{a_q}, \mathbf{a_q})$ - In this sub-scenario either the area's position or shape vary along with both quantitative attributes.

And for each sub-scenario there are the following examples:

- $(id, A, t, \mathbf{a_q}, \mathbf{a_q})$ - countries, where each countries boundaries is represented by an area. A country's life expectancy and children mortality rate are both quantitative attributes.

- $(id, \mathbf{A}, t, \mathbf{a_q}, \mathbf{a_q})$ - fires where the area represents the burned area at the instant of capture t. The number of assigned firefighters and affected buildings at instant t are both quantitative attributes.

3.1.23 Area shaped Objects with Two classification attributes

Shorthand	Type	Shape
(id, A, t, a_c, a_c)	Object	Area

This scenario covers the area shaped objects with two classification attributes. This scenario can be divided into 5 subs-cenarios, which are:

- $(id, A, t, \mathbf{a_c}, a_c)$ - In this sub-scenario there is a variation of one of the classification attributes while the other classification attribute does not change. The non-changing attribute can be used to identify objects of a specific class.

- $(id, A, t, \mathbf{a_c}, \mathbf{a_c})$ - In this sub-scenario both classification attributes vary at the same time.

- $(id, \mathbf{A}, t, a_c, a_c)$ - In this sub-scenario either the area's position or shape vary while both classification attributes do not change over time. The combination of both attributes can be used to identify objects of a specific class.

- $(id, \mathbf{A}, t, \mathbf{a_c}, a_c)$ - In this sub-scenario either the area's position or shape vary while there is a variation of one classification attribute and the other classification attribute stays with the same value over time. The nonchanging classification attribute can be used to identify different kinds of objects.

- $(id, \mathbf{A}, t, \mathbf{a_c}, \mathbf{a_c})$ - In this sub-scenario either the area's position or shape vary while there is a variation of both classification attribute attributes over time.

And for each sub-scenario there are the following examples:

- $(id, A, t, \mathbf{a_c}, a_c)$ - of public infrastructure (police stations, fire stations, hospitals, etc...), where each building is represented by an area with the shape of the building. The kind of building is a non changing classification attribute, while the rating of each building taken by public questionaires is a classification attribute.

- $(id, A, t, \mathbf{a_c}, \mathbf{a_c})$ - countries, where each countries boundaries is represented by an area. A country's credit rating and governing political party are changing classification attributes.

- $(id, \mathbf{A}, t, a_c, a_c)$ - oil spills, where the area represents the affected area and t marks the time of the reading. The cause of the spill and the company responsible are both non-changing classification attributes.

- $(id, \mathbf{A}, t, \mathbf{a_c}, a_c)$ - fires where the area represents the burned area at the instant of capture t. The cause of the fire is a non-changing classification attribute and whether the fire has been contained or not is a changing classification attribute.

- $(id, \mathbf{A}, t, \mathbf{a_c}, \mathbf{a_c})$ - volcanic clouds, where the area represents the affected area at a specific t. Whether a plane can fly through the cloud or not is a changing classification attribute, if the cloud is toxic or not is also a changing classification attribute.

3.1.24 Area shaped Objects with One classification and one quantitative attribute

Shorthand	Type	Shape
(id, A, t, a_c, a_q)	Object	Area

This scenario covers the area shaped objects with one classification and one quantitative attribute. This scenario can be divided into 4 subs-cenarios, which are:

- $(id, A, t, a_c, \mathbf{a_q})$ - In this sub-scenario the quantitative attribute varies over time and the classification attribute does not.

- $(id, A, t, \mathbf{a_c}, \mathbf{a_q})$ - In this sub-scenario both attributes can vary simultaneously.

- $(id, \mathbf{A}, t, a_c, \mathbf{a_q})$ - In this sub-scenario the quantitative attribute and either the area's position or shape vary ,while the classification attribute does not, serving as an identifier of objects of that classification.

- $(id, \mathbf{A}, t, \mathbf{a_c}, \mathbf{a_q})$ - In this sub-scenario either the area's position or shape vary along with both thematic attributes.

And for each sub-scenario there are the following examples:

- $(id, A, t, a_c, \mathbf{a_q})$ - of public infrastructure (police stations, fire stations, hospitals, etc...), where each building is represented by an area with the shape of the building. The kind of building is a classification attribute that does not change, while the number of people inside a building is a quantitative attribute.

- $(id, A, t, \mathbf{a_c}, \mathbf{a_q})$ - countries, where each countries boundaries is represented by an area. The country's population is a quantitative attribute while the countries credit rating is a classification attribute.

- $(id, \mathbf{A}, t, a_c, \mathbf{a_q})$ - fires where the area represents the burned area at the instant of capture t. The cause of the fire is a non changing classification attribute, and the number of assigned firefighters at instant t is a quantitative attribute.

- $(id, \mathbf{A}, t, \mathbf{a_c}, \mathbf{a_q})$ - volcanic clouds, where the area represents the affected area at a specific t. The concentration of volcanic gases is a quantitative attribute, and if a plane can fly through it or not is a classification attribute.

3.2 Analytical Tasks

This section of the catalog introduces the analytical tasks. Analytical tasks are operations that can be performed on datasets with the objective of gathering pertinent information about the data itself and are the main instrument an analyst has at his disposal. More specifically, for the context of this thesis, an analytical task is used to extract patterns from a dataset.

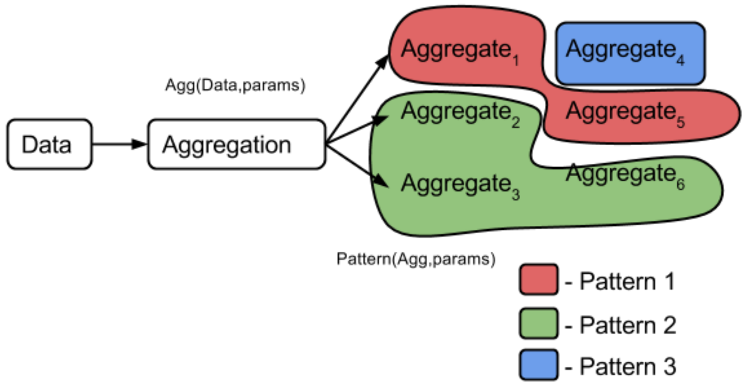


Figure 3.2: A flow diagram of the process of an analytical task for extracting patterns, separated in two steps an aggregation step and a patterning step.

An analytical task can be divided into two steps, as seen depicted by the diagram in figure 3.2. In the first step an aggregation is performed on the data according to some criteria and parameters and forms aggregates from the dataset. In the second step patterns are detected on single or multiple aggregates, depending on what kind of patterns the analyst wants to find.

An aggregation is a process that can be done on datasets to summarize the data; it is done by specifying attributes of the dataset that will be grouped and specifying functions to summarize the other attributes. In the end a smaller dataset is obtained where every combination of the grouped attributes is unique and has a summarization of the other attributes relating to it.

Considering the scope of spatiotemporal data, there are 3 kinds of patterns that can be extracted:

1. temporal patterns, which are patterns that occur over time, and can be extracted by using the tasks in section 3.2.1.
2. spatial patterns, which are patterns that occur over space, and can be extracted by using the tasks in section 3.2.2.
3. spatiotemporal patterns, which are patterns that occur over space and time, and can be extracted by using the tasks in section 3.2.3.

3.2.1 Analytical tasks for temporal patterns

For detecting temporal patterns in spatiotemporal datasets, the first step is to completely discard the spatial attributes, so any dataset is reduced to a time instant attribute, an id attribute and thematic attributes. In this family of tasks the dataset is summarized by the time instant attribute, and patterns are detected on this summarized data, called temporal patterns. So these patterns are a sequence of time units, where the summarized data associated to them satisfy some specified conditions. Using the established model for analytical tasks, a task for temporal patterns is divided in a step of temporal aggregation and a step of temporal patterning.

Temporal attributes have an implicit granularity and hierarchy: a day is part of a month which in turn is part of a year, also a day has a smaller grain than a month and a month a smaller grain than a year. This hierarchy can be used in these analytical tasks, by specifying a grain higher than the one of a datasets' instants; data can be summarized in smaller groups of values. For instance, a dataset with daily data can be grouped by month, which generates fewer values making it simpler to detect patterns.

In the case of a (P, t) scenario, when discarding the spatial attributes there are only temporal attributes. The only aggregation that is possible in this case is to count the number of entries for each instant t or count the number of entries for a higher grain of that time instant. In other words, this aggregation returns the number of events that occurred at each time unit, where a time unit can be a dataset instant or a higher grain. In figure 3.3, the bar chart N, shows the concentration of events from the table aggregated with different granularities.

In a data scenario with one quantitative attribute, for instance (P, t, a_q), the data is grouped by the specified time unit and the quantitative attribute is aggregated using an aggregation function. Several functions can be used in this case and only some are given as example which are: (1) *min*, returns the minimum of a_q values, (2) *max*, returns the maximum value of a_q, (3) *avg*, returns the average value of a_q and (4) *sum*, returns the sum of all the quantitative attributes. So, for each time unit, this aggregation assigns a metric of the quantitative attribute to it, a value that describes the quantitative attribute. The aggregation of the quantitative attribute is illustrated in figure 3.3 as the A_Q bar chart, showing again the possible aggregation at two different granularities, the one on the left uses the *sum* aggregation function at the lowest granularity while the bar on the right uses the *max* aggregation function at a higher granularity.

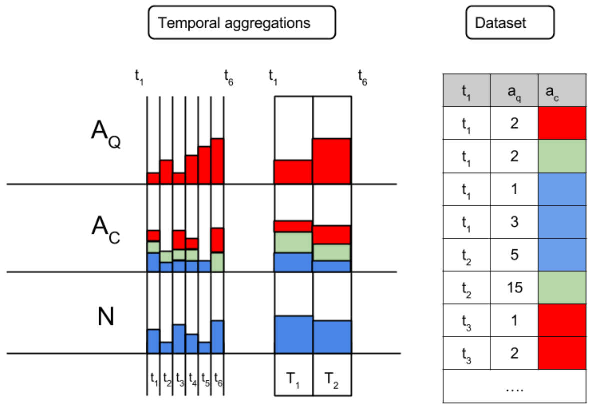


Figure 3.3: An illustration of some temporal aggregations as bar charts, with data from the table shown.

Furthermore, in object scenarios with quantitative attributes, for instance a (id, P, t, a_q) data scenario, the grouping can be done using the id and the temporal attribute at the same time. The aggregation functions are the same as the previous case, but in the end the aggregations describe for each time unit a metric of the quantitative attribute of each object.

When there is a classification attribute, for example in a (id, A, t, a_c) data scenario, there are two possible aggregations. The first is to aggregate by just grouping the temporal attribute, then the aggregation function over the classification attribute may be one of the following: (1) mode, the most frequent value is returned; (2) a function that returns the frequency of the most frequent value; (3) a function that returns the least frequent value. These aggregations return a metric of the classification attribute for each time unit. As was the case with scenarios that have a quantitative attribute, if there is an id, then the grouping can be done by the temporal attribute and the id, using the same aggregation functions. This kind of aggregation can only be applied to the sub-scenarios where there is a variation in the classification attribute and associates a metric that describes the classification attribute to each pair: object id, time unit.

In classification attribute data scenarios, the second possible aggregation is by grouping the aggregation with pairs time unit and classification attribute. The only possible aggregation function is to count the various pairs of classification values and time units. This aggregation returns the frequency of each classification value for each time unit. This aggregation is illustrated in figure 3.3 as the A_C bar charts, showing aggregations at two different granularities. Each bar is marked with the color of a classification value and its height represents the frequency of that value at that time. As in the previous aggregations, it is possible to group by id, when the dataset to be analyzed contains objects. This way, the aggregation gives the frequency of each classification value for each object at each time. The table in figure 3.4 describes the various aggregations explained, defining

Group by	Aggregate	Function	Requirements
t	none	Count	Only applicable to events
t	A_q	Sum, Avg, Max, Min	Only to scenarios with quantitative attributes
t	A_c	Mode	Only to scenarios with classification attributes
t, A_c	none	Count	Only to scenarios with classification attributes
t, id	A_q	Sum, Avg, Max, Min	Only to object scenarios with quantitative attributes
t, id	A_c	Mode	Only to object sub-scenarios with varying classification attributes
t, id, A_c	none	Count	Only to object sub-scenarios with varying classification attributes

Figure 3.4: The various temporal aggregations.

the requirements of a data scenario and listing example aggregation functions.

An aggregation by temporal attributes produces a sequence of temporal aggregates that is used in the patterning step. A temporal aggregate has some properties that are useful when defining the pattern step; it has temporal properties inherent from the time unit it is associated to. There is a defined time size that is given by the grain used in the aggregation. It is possible to define order between the aggregates and to calculate a temporal distance between two different aggregates. For example, in a dataset that has yearly data for 3 years and this data is aggregated by year, then there are 3 different aggregates, one for each year: $[A_{t1}, A_{t2}, A_{t3}]$. Assuming that A_{t1} is the aggregate for the first year and A_{t3} the aggregate of the last year, then there is an ordering A_{t1}, A_{t2}, A_{t3}, where A_{t1} is followed by A_{t2} and A_{t2} is followed by A_{t3}. Also it is possible to say that A_{t1} and A_{t3} are a year apart and A_{t2} and A_{t3} are a continuous interval of 2 years.

The pattern step in temporal analytical tasks is a two-part operation, the first part turns the sequence of aggregates in a sequence of sequences of aggregates using a sequencing function, the second part chooses the sequences of aggregates that form a pattern according to a specific predicate.

Three different sequencing functions were considered that are enough to cover all the discovered analytical tasks. For the description of each sequencing function an example sequence of aggregates is used: $[a_{t1}, a_{t2}, a_{t3}, a_{t4}]$, where t is the chosen granularity and a_{ti} is the aggregate at the time $i * t$. The first sequencing function, the single sequencing, returns a sequence per aggregate containing just that aggregate. So applying this sequencing function to the example introduced before, would result in: $[[a_{t1}], [a_{t2}], [a_{t3}], [a_{t4}]]$. The second sequencing function is the window; this function requires one parameter to be specified. This sequencing combines the various aggregates in continuous groups of a size specified by the parameter. In other words, applying this function to the previous example sequence, with a parameter value of 2, results in the following sequences:

Name	Sequence	Parameters	Description
Max	singular	none	Returns the aggregate with the maximum value of all aggregates.
Min	singular	none	Returns the aggregate with the minimum value of all aggregates.
Above	singular	x	Returns the aggregates that value above x.
Below	singular	x	Returns the aggregates that value below x.
Increasing	cyclic, window	none	Returns the sequences of aggregates that have values increasing over time.
Decreasing	cyclic, window	none	Returns the sequences of aggregates that have values decreasing over time.
Similar	cyclic, window	x	Returns the sequences of aggregates that have similar values considering a margin of x.

Figure 3.5: Some example predicates for temporal patterns

$[[a_{t1}, a_{t2}], [a_{t2}, a_{t3}], [a_{t3}, a_{t4}]]$.

The last sequencing function considered is the cycle sequencing, which also requires the use of a specified parameter. This sequencing function will be used to detect cyclic patterns, so the aggregates are grouped according to a cycle of time units, which is the value of the parameter. For instance, for a sequence of aggregates $[a_{t1}, a_{t2}, a_{t3}, a_{t4}]$ a cycle of size 2 will result in the following sequence of sequences: $[[a_{t1}, a_{t3}][a_{t2}, a_{t4}]]$. Both the window and cycle parameters are expressed as integers and have the same time granularity as the aggregation. This means that for a sequence of aggregates with a granularity of month a parameter would have to be at least one month.

The predicate definition is the final step in detecting temporal patterns; it is applied to each sequence of aggregates that resulted from the sequencing function. If a sequence of aggregates passes the predicate it is considered a pattern. There are several possible predicates, only some are presented as example in figure 3.5. The pattern predicate is limited to the sequencing that was used. For instance, the *max* predicate chooses the maximum of all the aggregates, and is only applicable when the single sequencing was applied. Conversely, the *increasing* predicate chooses the groups where the value of the aggregate has been increasing over several time units, so it can only be applied when the window or cycle sequencing were used. Some of the predicates may have parameters that help refine the patterns collected; for instance, the *above* criterion can be used to select sequence of aggregates where its values are above a specified parameter.

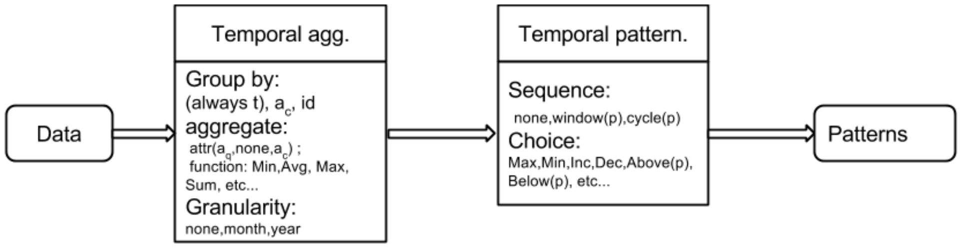


Figure 3.6: The various components of an analytical task for temporal patterns.

In conclusion, a temporal analytical task is made by specifying:

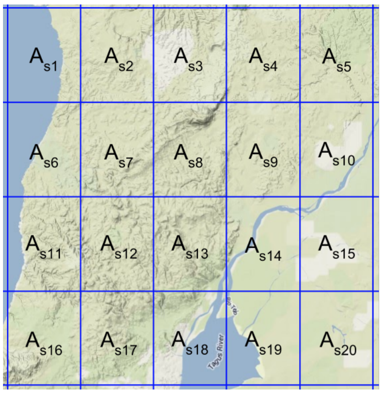
- the attributes that are grouped by, with the specification of a temporal granularity.
- the aggregation function to be used on the other attributes, where its choice is limited to the kind of the attribute being aggregated, according to figure 3.4.
- a temporal sequencing function (singular, window, cycle).
- a pattern predicate, that is dependent on the choice of a sequencing function, for example one from figure 3.5.

The specified analytical task will have one or more parameters depending on the chosen components, one parameter will be the granularity for the aggregation, depending on the sequencing function there can be one parameter or not, and the pattern predicate can have one or more parameters as well. A pattern derived from these tasks is defined by the whole process, in the end it is a sequence of one or more aggregates that can be identified by its grouped by attributes that the sequence of aggregated values formed and ordered according to its time attributes that finally verify a certain temporal predicate.

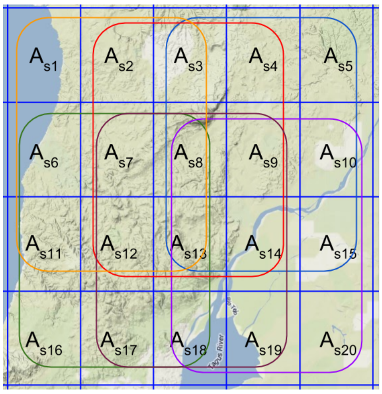
3.2.2 Analytical tasks for spatial patterns

This kind of analytical tasks start with the discarding of the temporal attributes and focuses on patterns that emerge from spatial factors. Patterns like similarities in events that occur close to each other or similarities between objects that are placed near natural landmarks are some of the possible patterns that can be extracted with these tasks.

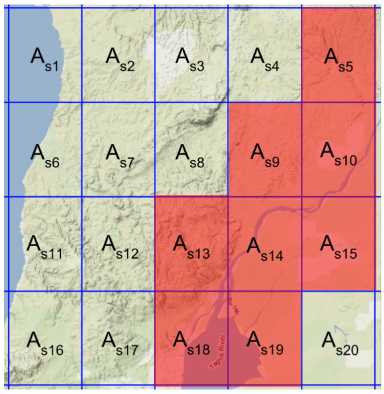
Procedurally, these tasks follow the same model as the tasks for temporal patterns. There is an aggregation that is performed according to spatial attributes and results in what are dubbed spatial aggregates. These aggregates are then sequenced using a spatial sequencing function and a pattern predicate is applied to those sequences.



(a) Initial Aggregation



(b) Neighborhood sequencing



(c) Around a shape sequencing

Figure 3.7: An example of two different spatial sequencings.

The process of a spatial aggregation is similar to the process of a temporal aggregation. The entries of the dataset are grouped by spatial attributes and summarized using the same aggregation functions as specified before. The difference in spatial aggregation is the need for specifying a spatial partitioning. This parameter of the spatial aggregation describes a division of the space occupied by the dataset into partitions. This partitioning can be as simple as a square grid where all squares have the same width and height as seen in figure 3.7(a). Or the space can be partitioned using real world divisions like: administrative regions, neighborhoods, countries, provinces, lots, parcels, etc... Furthermore, a partitioning of the space using a clustering algorithm derived from the same or other datasets can be used for more complex analyses. The spatial division is essential for analytical tasks on event scenarios, and is a powerful parameter overall for finding spatial patterns. It is not important to define the various possible spatial divisions, since they do not limit the other properties or steps of an analytical task. The defined aggregation functions, the sequencing functions and patterns can be applied using any spatial division. For object scenarios, the spatial division can be ignored, since each object has specific spatial properties.

The attributes that can be aggregated and the functions that can be used are the presented in figure 3.8 which are similar to the ones presented in figure 3.4, but instead of

Group by	Aggregate	Function	Requirements
s	none	Count	Only applicable to events
s	A_q	Sum, Avg, Max, Min	Only to scenarios with quantitative attributes
s	A_c	Mode	Only to scenarios with classification attributes
s, A_c	none	Count	Only to scenarios with classification attributes
id	A_q	Sum, Avg, Max, Min	Only to object scenarios with quantitative attributes
id	A_c	Mode	Only to object sub-scenarios with varying classification attributes
id, A_c	none	Count	Only to object sub-scenarios with varying classification attributes

Figure 3.8: The various spatial aggregations.

the temporal attribute *t* in the group by column a spatial division *s* is used. Also, in the particular cases of object scenarios only the *id* is used to aggregate instead of a spatial division, this is because an *id* implies a spatial entity that is associated to the object.

Spatial properties of an aggregate are what is used in a spatial sequencing function. These properties define an aggregate's position in space and can be used: (1) to calculate a distance between aggregates, (2) to define the position of an aggregate in order of another, for instance to say that an aggregate is located to the north of another, (3) to determine proximity between aggregates. The definition of a spatial sequencing function builds on these spatial properties to create sequences of aggregates. There are several possible spatial sequencing functions and only a few are explained as examples.

Considering the grid spatial division in figure 3.7(a), and that there is an aggregate associated with each spatial division that is identified by A_{sn} where n stands for the respective spatial aggregate.

The first spatial sequencing is the neighborhood sequencing, which returns all the neighbors of all the spatial aggregates with the used spatial aggregate included. A neighbor of a spatial aggregate A_{si} is defined as a spatial aggregate that shares a border with A_{si}. Figure 3.7(b) depicts this sequencing by identifying some of the sequences with colored rounded rectangles in the example division of figure 3.7(a). For example, the neighborhood for A_{s1} is $[A_{s1}, A_{s2}, A_{s6}, A_{s7}]$, and the neighborhood for A_{s14} is the sequence of aggregates: $[A_{s8}, A_{s9}, A_{s10}, A_{s13}, A_{s14}, A_{s15}, A_{s18}, A_{s19}, A_{s20}]$

The second example sequencing is to sequence due to the proximity to certain natural landmarks. The aggregates are grouped by landmark, and only if they are within a certain margin from it, the margin is a parameter of this spatial sequencing. The example in figure 3.7(c) depicts a sequencing made around the river Tagus. The spatial aggregates that are near the river are: $[A_{s5}, A_{s9}, A_{s10}, A_{s13}, A_{s14}, A_{s15}, A_{s18}, A_{s19}]$, the aggregates A_{s5}

and A_{s9} are at a close distance from the river, while the other aggregates from the sequence actually have the river passing through.

Finally, the spatial predicates that are applied to the spatial sequences are the same that are used in the previous section with the exception of the *increasing* and *decreasing* predicates which require a temporal ordering of the sequences. The process is the same, a predicate that may have one or more parameters, is applied to a sequence; if the sequence verifies the predicate it is considered a pattern. This may be the same operation, but because of the previous spatial aggregations and spatial sequences the semantic meaning of the patterns is very different from the meaning of temporal patterns. Furthermore, predicates that rely more on the spatial properties of the sequence can be used, for instance a predicate that tests if the aggregates on the sequence increase in value from east to west or from north to south, as in previous parts of this sections the predicates presented are examples that may be used later, but a task is not limited to these patterns.

In conclusion, a spatial analytical task is made by specifying:

- the attributes that are grouped by, with the possible aid of a spatial division.
- the aggregation function to be used on the other attributes, where its choice is limited to the kind of the attribute being aggregated, according to figure 3.8.
- a spatial sequencing function (neighborhood, inside a shape, around a natural landmark).
- a pattern predicate, that is dependent on the choice of a sequencing function.

The specified analytical task will have one or more parameters depending on the chosen components, one parameter may be the spatial division to use for the aggregation, depending on the sequencing function there can be one parameter or not, and the pattern predicate can have one or more parameters as well. A pattern derived from these tasks is defined by the whole process, in the end it is a sequence of one or more aggregates that can be identified by its grouped by attributes that the sequence of aggregated values formed and organized according to its spatial position, which ultimately verifies a spatial predicate.

3.2.3 Analytical tasks for spatiotemporal patterns

There are three approaches to finding spatiotemporal patterns using the proposed models. The whole concept of analyzing relies on an iterative analyses process, where several analytical tasks are chained in order to find a specific analysis. The first two approaches rely on this principle, and use the already established analytical tasks for spatial patterns and temporal patterns to find spatiotemporal patterns.

An analytical task can be used to detect a temporal pattern with the result being sequences of temporal aggregates, in other words a array of summarized data that are related to a specific time interval. The initial dataset can be filtered by the time interval

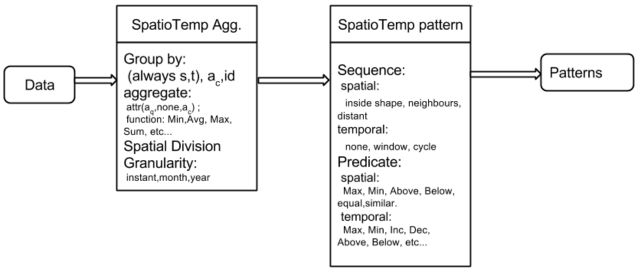


Figure 3.9: A diagram depicting a description for analytical tasks for detecting spatiotemporal patterns

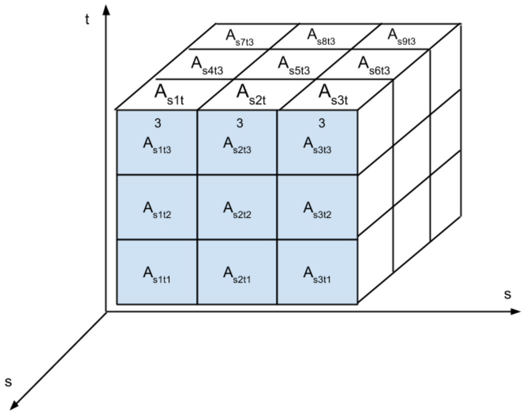


Figure 3.10: A 3d visual representation of spatiotemporal aggregates using a grid spatial division and an unspecified temporal granularity.

of one of the patterns and then used in an analytical task for spatial patterns, if a spatial pattern is found then it is also a spatiotemporal pattern.

The reverse process will also result in detecting spatiotemporal patterns, meaning that, if a spatial pattern is found on a dataset using one of the tasks from section 3.2.2, the dataset can be filtered by the obtained spatial locations of the patterns. The filtered dataset can then be analyzed with a technique for temporal patterns from section 3.2.1 and the resulting temporal patterns will also be spatiotemporal patterns.

These two processes are good utilities for detecting spatiotemporal patterns but do not specify a single analytical task for spatiotemporal patterns. The last approach is to mix both models for tasks of spatial patterns and temporal patterns into a model that describes tasks that find spatiotemporal patterns. This results in the model of figure 3.9. An analytical task for detecting spatiotemporal patterns is divided in a process of spatiotemporal aggregation and a process of spatiotemporal patterning. Spatiotemporal aggregation is done by specifying a spatial division and a temporal granularity. The attributes to group by and the attributes to aggregate follow the same pairings as before but with the spatial and temporal attributes always in the group by clause. The spatiotemporal aggregates have both the properties from temporal aggregates and spatial aggregates; so each aggregate has associated to it a pair position in time, position in space, or (t, s), that is unique to each aggregate.

In the patterning step the sequences are generated using both a temporal sequence function and a spatial sequence function. As an example, for the spatiotemporal aggregation in figure 3.10 that has been aggregated with a certain temporal granularity of 4 instants and a grid spatial division. A sequencing done by using a temporal sequencing of window of size 2 and a spatial sequencing of neighborhood, one of the many derived sequences would be the small cube: [A_{s1t1}, A_{s1t2}, A_{s2t1}, A_{s2t2}, A_{s4t1}, A_{s4t2}, A_{s5t1}, A_{s5t2}].

Then two predicates are applied to the sequence of aggregates, one predicate, the temporal predicate is applied separately to all the aggregates of the sequence that pertain to the same spatial division but to different time units. The second predicate, the spatial predicate is applied separately to all the aggregates of the sequence that pertain to the same time unit but different spatial division. Considering the previous example sequence [A_{s1t1}, A_{s1t2}, A_{s2t1}, A_{s2t2}, A_{s4t1}, A_{s4t2}, A_{s5t1}, A_{s5t2}], taken from figure 3.10, the temporal predicate would be applied to the aggregates [A_{s1t1}, A_{s1_t2}], [A_{s2t1}, A_{s2_t2}], [A_{s4t1}, A_{s4_t2}] and to [A_{s5t1}, A_{s5_t2}], while the spatial predicate would be applied to [A_{s1t1}, A_{s2_t1}, A_{s4t1}, A_{s5_t1}] and [A_{s1t2}, A_{s2_t2}, A_{s4t2}, A_{s5_t2}]. This distinction between spatial predicates and time predicates is a simplification of the model that helped in easily specifying several tasks for spatiotemporal patterns. However, more complex predicates that verify the whole spatiotemporal cubes of aggregates could be used if needed.

In conclusion, a spatiotemporal analytical task is made by specifying:

- the attributes that are grouped by, with the aide of a spatial division and a temporal granularity.
- the aggregation function to be used on the other attributes, where its choice is limited to the kind of the attribute being aggregated.
- a temporal sequencing function(singular, window, cycle), and a spatial sequencing function (neighborhood, inside a shape, around a natural landmark).
- a pattern predicate, that is dependent on the choice of a sequencing function, this predicate can be made by applying a temporal predicate and a spatial predicate.

The specified analytical task will have one or more parameters depending on the chosen components, one parameter will be the granularity for the aggregation, another will be a spatial division. Depending on the sequencing functions there can be one parameter or more parameters, and the pattern predicates can have one or more parameters as well. A pattern derived from these tasks is defined by the whole process, in the end it is a sequence of one or more aggregates that can be identified by its grouped by attributes that the sequence of aggregated values formed. This aggregates have a certain temporal order associated as well as a spatial distribution, which ultimately verify a spatiotemporal pattern.

This model is used to describe the analytical tasks in the following sections of this chapter. Those tasks were extracted from the papers that introduced all the spatiotemporal techniques that were described in section 2.4, and will be later explained as part of the catalog in section 3.3.

3.2.4 Find spatiotemporal areas with interesting concentration of events

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s, count(*)	Window	Neighborhood	Above	Above

The dataset, which is an event data scenario, is aggregated by counting the number of events per spatial division(p_2) and time unit with a specified granularity (p_1). The spatiotemporal aggregates are sequenced temporally by time window with a specified size (p_3) and sequenced spatially by neighborhood. A sequence of aggregates is a spatiotemporal pattern if all the aggregates are above a parameter value (p_4).

This task has 4 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time window and (p_4) the value which all aggregates have to be above of.

For example, considering a dataset of phone calls over a city during a week, registered per minute. By analyzing the number of calls made per hour (p_1) distributed over a grid with 1 km squares (p_2). A user may find a specific neighborhood, a group of 1km squares, where in an interval between 7PM and 9PM ($p_3 = 1hour$) on Friday the concentration of phone calls was significantly high, meaning its above a certain value (p_4) that the analyst deems interesting.

3.2.5 Analyze events occurrence rate in a certain area

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s, count(*)	Window	Neighborhood	Increasing	Similar

The dataset, which is an event data scenario, is aggregated by counting the number of events per spatial division(p_2) and time unit with a specified granularity (p_1). The spatiotemporal aggregates are sequenced temporally by time window with a specified size (p_3) and sequenced spatially by neighborhood. A sequence of aggregates is a spatiotemporal pattern if: (1) each spatial division has an increase in number of events over time units, (2) for each time unit all spatial divisions have similar concentration of events within a certain margin of similarity (p_4).

This task has 4 parameters, which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time window and (p_4) the margin of similarity between aggregates.

Considering a dataset of fires over a country during the summer period, registered daily. By aggregating the number of fires by 3 day periods (p_1), and distributing the data spatially over a grid of parcels with an of 100 km each (p_2). This task allows the user to find a specific group of neighboring parcels, where in a 9 day period at the end of July the number of fires for the first 3 days is around 0 to 2 ($p_4 = 1$) fires per parcel, for the second 3 days is around 1 to 3, and for the last three days the number of fires that occurred was around 3 to 5 fires.

3.2.6 Detect cyclic patterns of occurrence

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s, count(*)	Cycle	None	Similar	None

The dataset, which is an event data scenario, is aggregated by counting the number of events per spatial division(p_2) and time unit with a specified granularity (p_1). The spatiotemporal aggregates are sequenced temporally by cycles of a specified size (p_3) and without any spatial sequencing. A sequence of aggregates is a spatiotemporal pattern if each spatial division has the same number of events over time, within a certain margin of similarity (p_4).

This task has 4 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time cycle and (p_4) the margin of similarity between aggregates.

For example, in a dataset of photos taken daily over a certain region during a period of 5 years, while aggregating the data per season (p_2) of the year and dividing it by administrative regions (p_1). This task allows the user to find a specific region where every summer ($p_3 = 4$) the concentration of photos taken inside it was between 190 and 210 ($p_4 = 10$).

3.2.7 Finding peak attribute values in objects

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s,id,Min(a_q)	Window	Neighborhood	Above	Above

The dataset, which is an object data scenario with at least one quantitative attribute, is aggregated by minimum a_q value that occurred per defined time unit (p_1) and over each specified spatial division(p_2). The spatiotemporal aggregates are sequenced temporally by time window with a specified size (p_3) and sequenced spatially by neighborhood. A sequence of aggregates is a spatiotemporal pattern if all the aggregates, the minimum a_q values, are above a parameter value (p_4).

This task has 4 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time window and (p_4) the value which all aggregates have to be above of.

Considering a dataset of European countries with daily GDP information over a period of 10 years. Using no spatial division (p_3) and aggregating the minimum value per month. This task allows the user to find groups of neighboring countries where in a 3 year period (p_3) all countries add their GDP above 5.000 US$ (p_4).

3.2.8 Finding peak attribute values in events

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s,Max(a_q)	Single	None	Max	Max

The dataset, which is an event data scenario with at least one quantitative attribute, is aggregated by the maximum a_q value that occurred over a spatial division (p_2) using a specified temporal granularity (p_1). The spatiotemporal aggregates are temporally sequenced as single time units and without any spatial sequencing. A spatiotemporal pattern is found in the aggregate that is the maximum value of all the aggregates.

This task has 2 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division.

Considering a dataset of crimes with respective number of victims over a city during a yearlong period, registered every hour and on the location it occurred. By aggregating it to daily information (p_1) with the maximum of victims of the crimes using a 1km square grid over the city (p_2). This task allows an user to find when (time unit) and where (spatial division) the crime with the most victims occurred.

3.2.9 Finding regions of objects with variation trends

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s, Avg(a_q)	Window	Near Object	Similar	Outer Decrease

The dataset, which is an object scenario with one quantitative attribute, is aggregated using the average of the quantitative attribute using a specified temporal granularity (p_1)

and a spatial division (p_2). These spatiotemporal aggregates are sequenced temporally according to a window of a certain size (p_3) and spatially by neighboring aggregates. A spatiotemporal is found if for a sequence of spatiotemporal aggregates, each spatial aggregate has similar values within a margin (p_4) over time, and for each time unit the attribute value decreases depending on the aggregates position to the center of the sequence.

This task has 4 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time window and (p_4) the margin of similarity between temporal aggregates.

Considering a dataset of shops in shopping districts of a city with total sales of working days during a yearlong period, with data collected every minute per store only at lunch times. Using this task the sales are aggregated to its average per hour (p_1) using the buildings as a spatial division (p_2), they are sequenced temporally as 5 day (p_3) time windows, and spatially as proximity to natural parks. Using this sequences a user can find natural parks where the sales over 5 day periods during the spring are the same per store (p_4), but the sales are much higher on stores that are closer to the park.

3.2.10 Finding regions of events with variation trends

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s,id,Avg(a_q)	Window	Neighbourhood	Trends towards	None

The dataset, which is an event data scenario with one quantitative attribute, is aggregated using the average of the quantitative attributes using a specified temporal granularity (p_1) and a spatial division (p_2). This tasks sequences the spatiotemporal aggregates temporally using a window of a specific size(p_3) and spatially using neighboring aggregates. A pattern is found when a sequence of aggregates has each spatial division trending towards the same quantitative value (p_4) over the time period.

This task has 4 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time window and (p_4) the a_q value trend.

Considering a dataset of phone calls over a city during a weekly period, registered every minute with each call's duration. Using this task, by aggregating the data with the average of every hour (p_1) and 1km square(p_2), a user can then sequence the data by windows of 3 hours (p_3) and by neighboring squares. Then the user can find said sequences of 3 hour neighboring 1km squares where, in that time period, the average of phone call durations trended towards 5 minutes (p_4).

3.2.11 Finding trending regions of events

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s,Mode(a_c)	Window	Neighborhood	Trends towards	None

The dataset, which is an event scenario with one classification attribute, is aggregated by the most occurring classification attribute using a specified spatial division (p_2) with a certain temporal granularity (p_1). The spatiotemporal aggregates are sequenced by a time window of a specified size (p_3) and sequenced spatially in neighborhoods. A spatiotemporal aggregate forms a pattern if, for each spatial division its value over time changed to the specified classification value (p_4) and stayed that value for the rest of the time window.

This task has 4 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time window and (p_4) the trend classification value.

For example a dataset of houses with tv channel being watched over a city during a one month period, registered every minute. This data can be aggregated by hours (p_1), and divided it into a spatial grid of 1km sided squares (p_2), a user can use this task to identify a specific group of neighboring aggregates where everyone changed to the same tv channel (p_4) at a time period between 1PM and 3PM.

3.2.12 Finding trending regions of objects

Aggregation	T. Seq.	S. Seq	T.Predicate	S. Predicate
t,s,id,Mode(a_c)	Window	Near Object	Trends towards	None

The dataset, which is an object scenario with at least one varying classification attribute, is aggregated by the most occurring classification attribute using a specified spatial division (p_2) with a certain temporal granularity (p_1). The spatiotemporal aggregates are sequenced by a time window of a specified size (p_3) and sequenced spatially by proximity to a spatial object. A spatiotemporal aggregate forms a pattern if, for each spatial division its value over time changed to the specified classification value (p_4).

This task has 4 parameters which are: (p_1) the temporal granularity, (p_2) the spatial division, (p_3) the size of the time window and (p_4) the trend classification value.

Considering a dataset of political districts with polling data over a city during a year-long period, registered weekly. A user by aggregating this data in monthly data (p_1) and using the same districts as a spatial division (p_2), can use this task to identify a specific group of districts around a newly renovated area, that have in a period of 3 months (p_3), all switched to the same political party (p_4).

3.3 Techniques

This section of the catalog categorizes techniques that were presented in chapter 2. It is one of the main objectives of the catalog to be able to more clearly expose each technique in terms of what situations it can be used in and what can be done with them. This way it is easier to see the differences and similarities between the various techniques. The characterization of each technique starts with a small textual description that focuses on the key aspects of the technique. The rest of the classification is done using the previous elements of the catalog: the scenarios and the analytical tasks.

Each technique has a set of scenarios it can be used on and a set of tasks it can perform on those scenarios. However, this section is not just assigning scenarios, and tasks to the techniques. A scenario's representation in one technique is different then its representation in another technique. Different techniques have different ways of representing events and object scenarios, different ways of representing a scenario's shape and attributes. So all scenarios that can be analyzed by a technique are grouped by shape, and kind, and are described using that technique's vocabulary and details. Also, a list of bibliographical references and scenarios is provided to show where that technique was applied to that scenario.

The analytical task characterization of a technique is divided in two parts. The first part is a general description of the various concepts of the framework of analytical tasks and how these translate to the technique's concepts. Spatial divisions, temporal aggregations, predicates and sequencing, all may be represented differently depending on the technique. The second part of the analytical tasks characterization, is a list of some of the tasks from section 3.2, that can be performed using the specified technique. Each element of said list is described using the already established concepts of the technique. Also, provided for each task of the list, is a list of scenarios where the technique can perform that task.

3.3.1 Animation

Shorthand A

Description For a dataset comprised of spatiotemporal data between two dates d_1, d_2 in a certain region, an animation is built between d_1 and d_2 showing for each frame of the animation a thematic map of that region.

Scenarios

Total Covered (id, A, t) and (id, A, t, a_c) in [AAG03]

[(id, A, t) and (id, A, t, a_c)]- The area shaped objects are represented by their geographical projection on the map showing their shape and position. The classification attribute is mapped to each area's color.

Analytical Tasks

In term of temporal granularity, this technique needs to use a fine grain; otherwise the animation may suffer drastic changes between frames, which can cofound the user.

For temporal sequencing the animation technique has the temporal window and single. Patterns that occur in temporal windows are spotted when playing through the animation and a pattern emerges visually during the interval of the time window, by replaying the animation at that interval the user can pinpoint the exact time window. The singles sequencing can only be used when the user pauses the animation, and is used to detect spatial patterns that occurred at certain instants of the desired time grain.

Concerning spatial divisions, this technique can perform any of the divisions that are part of the scope of this catalogue. All the examples of spatial sequences that were given are also possible, it is only limited by the zoom level, since it affects what the user sees, high zoom levels may hide parts of a spatial sequence. .

3.2.12 - [(id, A, t, a_c) **]** A thematic map is made for each instant of the specified temporal granularity, and any spatial division can be used, each division is painted with a color symbolizing the most occurring classification value. The view of the map is placed over the object and at a zoom level that can show all the divisions close to that object. Playing the animation in consecutive time windows of the desired size, a pattern will emerge when all the divisions close to the object turn the same color.

3.3.2 Chorems

Shorthand C

Description For a dataset comprised of spatiotemporal data between two dates d_1, d_2 in a certain region, a thematic map is made by placing chorems on a map of that region. The chorems describe what happened between d_1 and d_2 at its location.

Scenarios

Total Covered (id, P, t), (id, P, t, a_c) and (id, P, t, a_q) in [CFL+11]

[(id, P, t)**,** (id, P, t, a_c) **and** (id, P, t, a_q) **]-** The point shaped objects are represented as chorems in their respective geographical location. Both classification and quantitative attributes are mapped to different chorem symbols.

Analytical Tasks

Considering time granularity, since the chorems technique relies on icons that symbolize effects over a time period, only coarse temporal granularities can be used with this technique. In other words, only granularities that reduce the dataset to a few time periods can be used. In terms of time sequencing the chorem representation is limited to showing one single time sequence per map, be it single sequencing, a single temporal window, or a cycle sequence. In terms of windows and cycles, the bigger the cycle or window, the more complex each chorem will be.

The spatial divisions used with this technique have to be relatively big in relation to the zoom level that is used. Since each chorem is placed at the center of each division, if these divisions are not bigger than each chorem they will collide with each other, and clutter the visualization in a useless mixture of visual aides. As with all techniques that are base in maps the spatial sequencing is directly related to the level of zoom, and all the considered spatial sequences are possible.

3.2.9 - [(id, P, t, a_q) **]** each point object is represented by a chorem that describes what happened to the quantitative attribute at a certain time window, with the size of the chorem translating to the attribute value at the last instant of the time window. A chorem with an $=$ symbol can be used to mean that the quantitative attribute stayed the same for the given time window. In a group of chorems around a natural landmark the pattern will emerge when all the chorems are $=$ and the size of a chorem is bigger for the ones closer to the landmark.

3.3.3 Growth Ring Maps

Shorthand GRM

Description For a dataset comprised of spatiotemporal data between two dates d_1, d_2 in a certain region, a growth ring map is made by placing growth rings on a map of the region. A growth ring describes what happened between d_1 and d_2 at the position of the ring's center or the ring's neighboring region.

Scenarios

Total Covered (id, P, t, a_q) in [BMJK09] and covered (P, t) and (P, t, a_c) in [AAB+11]

[(id, P, t, a_q) **]**- Each point shaped object is represented by a growth ring projected on the map at the point's geographical position. Each attribute value is represented by a point in the ring and for quantitative attributes its value is mapped to the points.

[(P, t) **and** (P, t, a_c) **]**- Each point shaped event is represented as a point in the growth ring closest to where the event occurred. The classification attribute is mapped to the events color.

Analytical Tasks

With respect to temporal granularity, for this technique the finer the grain used the bigger each ring will be. Also each ring will have a lot more different brightness levels with a fine grain than with a coarse grain. So a coarser grain reduces the complexity of the visualization, by showing less colors and smaller rings. In terms of temporal sequencing, this technique is able to represent all the cycles of a specific size present in the dataset. A time window is represented by a group of consecutive rings, with no interrupting break in brightness between the rings. The data pertaining to a single time unit of the specified temporal granularity is represented by a ring of a single color and brightness in all the rings, therefore it is quite hard to detect spatial patterns at single time periods, temporal patterns in single growth rings are possible by comparing the several rings o on growth ring.

The growth ring maps technique can use spatial divisions that are relatively big in accordance to the used zoom level, so that the growth rings do not collide with neighboring rings. The various spatial sequences considered can be used in this technique.

3.2.4 - [(P, t) **and** (P, t, a_c) **]** The space is divided using relatively big spatial division and at the center of it a growth ring is placed. Each pixel in the ring represents an event; and each possible color is associated to a single time unit, depending on the grain. The time window can be represented as a collection of consecutive rings. The pattern will emerge in neighboring growth rings that have thick consecutive rings in the desired time window. The thickness of the ring determines the number of events in that time period.

3.2.5 - [(P, t) **and** (P, t, a_c) **]** The space is divided using relatively big spatial division and at the center of it a growth ring is placed. Each pixel in the ring represents an event; and a single color is associated to a all events, but a brightness level of that color is associated to a single time unit, depending on the defined temporal granularity. The time window can be represented as a collection of consecutive rings. The pattern will emerge in neighboring growth rings that have consecutive rings that increase in width for the desired time window. Of this time period all rings of the same color, in other words that refer to the same time period, have similar widths.

3.2.6 - [(P, t) **and** (P, t, a_c) **]** The space is divided using relatively big spatial division and at the center of it a growth ring is placed. Each pixel in the ring represents an event; and each possible color is associated to a single cycle, depending on the defined temporal granularity. The brightness of each ring defines the order in each cycle, with a brighter ring representing older cycles and dark rings younger cycles. The pattern will emerge in neighboring growth rings where the rings with the same color, same cycle, have similar widths for every across all possible brightness levels.

3.3.4 Space Time Cube

Shorthand STC

Description For a dataset comprised of spatiotemporal data between two dates d_1, d_2 in a certain region, each entry of the dataset is placed inside a space time cube. A space time cube is a 3d rectangular prism that has a map of the region as its base and the interval between d_1 and d_2 as its height.

Scenarios

Total Covered (id, A, t) without references, covered (id, A, t, a_q, a_q) in [TSWS05], covered (P, t), (P, t, a_c), (P, t, a_q) and (P, t, a_c, a_q) in [GAA04], covered and in [Hä70], covered (id, A, t, a_c) and (id, A, t, a_q) in [TH10], covered (id, P, t) in [Kra03], covered (id, L, t, a_q) and (id, L, t, a_c) in [TSAA12] and covered (P, t) in [AAM+10]

[(id, A, t), (id, A, t, a_q, a_q), (id, A, t, a_c) and (id, A, t, a_q)]- A straight column of disks along the time axis represents each area shaped object, the column's base position is on the area's center. Each disk represents an attribute value, quantitative attributes are mapped to the disks area or the disks saturation level. A classification attribute can be mapped to the disks color.

[(P, t), (P, t, a_c), (P, t, a_q) and (P, t, a_c, a_q)]- Each point shaped event is represented by a sphere inside the cube, with its geographical and time position as its coordinates. A quantitative attribute can be mapped to the the sphere's saturation level or volume while a classification attribute can be mapped to its color.

[(id, P, t)]- Each point shaped object is represented by a sphere inside the cube, with its geographical and time position as its coordinates. A line can be used to link the different spheres that.

[(id, L, t, a_q) and (id, L, t, a_c)]- Each line shaped object is represented by a wall over the time axis and with the lines shape and position as its base. The lines segments' can have different attribute values by mapping each segment's color to classification attributes and each segment's saturation level to quantitative attributes..

Analytical Tasks

In terms of granularity, for the specific case of event scenarios aggregating by number of events, this technique cannot change granularities since it will increase the number of events per time unit, and will clutter the visualization. With aggregations of event scenarios and attribute values do not have the same effect. For static object scenarios it is possible, reducing the number of disks per column.

Relating to temporal sequencing, the cycle sequencing has been used in this technique but very poorly, with no real analytical capacities. The time window sequencing is easily represented in static object scenarios, a sub-column of n disks identifies a window of size n. In single sequencing, for object scenarios, each disk identifies a single time unit and only two kinds of patterns can be found that rely on this sequencing. The first are temporal patterns that can be found on single columns, because it is easy to compare the various disks in a single column. The second patterns are spatial patterns for a single time unit, since when a user selects a single disk a base map is drawn at that level showing all the disks for that instant or interval.

For event scenarios, single sequencing is possible to identify single events that stand out from the rest, for example, events with maximum attribute values. It is also used to identify spatial patterns at specific instants using the base map as was explained before. Time window sequencing is less obvious in event scenarios since the camera angle an analyst uses may affect the view he has on certain time windows.

Spatial divisions are directly related with the zoom level of the base map and the camera angle used. Most spatial divisions boundaries will be represented on the base map and will have a column of disks sprouting from the center of the division. If the divisions are to small and too close at the used zoom level, the columns will be to close and clutter the visualization, also certain camera angles may hide some of the columns of disks. The various example spatial sequences that were considered are possible, but it is another facet of this technique that is limited by the camera angle. Some angles may give the idea that some aggregates are further away or close then they actually are.

3.2.4 - [(P,t)**,** (P,t,a_c)**,** (P,t,a_q) **and** (P,t,a_c,a_q) **]** Each event is marked as small sphere inside a space time cube at the desired coordinates, looking over the time axis a user can define a time window of any desired size. The areas with most concentration will be identified as cubes filed with spheres, where the height of the cube is the size of the window, and the base will show the spatial divisions that are involved.

3.2.7 - [(id,A,t,a_q,a_q)**,** (id,A,t,a_q) **and** (id,L,t,a_q) **]** Static objects are represented as columns of disks where each disk represents the attribute value at the desired temporal granularity; quantitative attributes are mapped to the width of each disk. A spatial division aggregates the multiple columns of each division into one at its center. Neighborhoods of aggregates are identified by the proximity of columns; a time window is identified by a continuous sequence of disks. The pattern is identified by a group of continuous disks as thick as the desired value, for neighboring columns.

3.2.8 - [(P,t,a_q) **and** (P,t,a_c,a_q) **]** events are represented as spheres with the quantitative attribute mapped to its radius. The event with the peak attribute value is the biggest sphere.

3.2.10 - [(P,t,a_q) **and** (P,t,a_c,a_q) **]** A straight column of disks along the time axis represents each spatial division, the column's base position is on the division's center. Each disk's width represents the average of the a_q attribute value for the desired temporal grain. The pattern shows in neighboring columns for sequences of disks of each column that start and end at the same time. Each of this column's top disks have a width that represents the trend value.

3.2.11 - [(P,t,a_c) **and** (P,t,a_c,a_q) **]** Each event is marked as a small sphere inside a space time cube at the desired coordinates, with the color of the sphere defining the a_c value. Looking over the time axis a user can define a time window of any desired size. The desired pattern will be identified as a parallelepiped where the height is the size of the time window, and the base is made of neighboring parcels. The color of the events near the top parallelepiped is basically the color of the trend classification attribute.

3.4 Conclusion

In this catalogue's last section an overview is presented of how its 3 components, scenarios, analytical tasks and techniques, are related. This is done by a one to one association in figures 3.11, 3.12 and 3.13. Figure 3.11 shows the relation between scenarios and tasks, meaning for each scenario it shows which analytical task can be done on it. This way, when a dataset is conformed to one scenario its easy to find which kind of spatiotemporal patterns can be extracted from it. The tasks in that table are ordered so it's easier to find groups of tasks that are applied to a group of scenarios, for example, the first 6 scenarios are associated only to the first 6 tasks.

Next, figure 3.12, shows the relation between scenarios and techniques, this table shows for each scenario what techniques can be used to represent and analyze them. As can be seen by the last 10 empty rows there are still a lot of scenarios which have no way to be represented using todays techniques, namely most scenarios with two attributes have no techniques that can be used.

The last overview table, in figure 3.13, associates the analytical tasks with the spatiotemporal techniques. With this table we can see that every task has at least one technique that can be used. Also, it can be seen that the space time cube stands out as the technique with most analytical tasks, with a total of 5 tasks.

As could be seen in figure 3.11, tasks can be grouped in accordance with its applicable scenarios. So, instead of presenting a table that shows the relation between scenarios, analytical tasks and techniques, the tasks were split into groups according to applicable scenarios. In figures 3.14, 3.15, 3.16 and 3.17 is a more detailed description of the association between the three components, each cell in one of these tables shows which techniques can be used to analyze a specific scenario using a specific analytical tasks. The **-** symbol means there are no techniques, while a **NA** means that the task is **n**ot **a**pplicable to that scenario.

Scenarios	Tasks								
	3.2.4	3.2.5	3.2.6	3.2.8	3.2.10	3.2.11	3.2.7	3.2.9	3.2.12
(P, t)	X	X	X	-	-	-	-	-	-
(P, t, a_q)	X	X	X	X	X	-	-	-	-
(P, t, a_c)	X	X	X	-	-	X	-	-	-
(P, t, a_q, a_q)	X	X	X	X	X	-	-	-	-
(P, t, a_c, a_c)	X	X	X	-	-	X	-	-	-
(P, t, a_c, a_q)	X	X	X	X	X	X	-	-	-
(id, P, t)	-	-	-	-	-	-	-	-	-
(id, P, t, a_q)	-	-	-	-	-	-	X	X	-
(id, P, t, a_c)	-	-	-	-	-	-	-	-	X
(id, P, t, a_q, a_q)	-	-	-	-	-	-	X	X	-
(id, P, t, a_c, a_c)	-	-	-	-	-	-	-	-	X
(id, P, t, a_c, a_q)	-	-	-	-	-	-	X	X	X
(id, L, t, a_q)	-	-	-	-	-	-	X	X	-
(id, L, t, a_c)	-	-	-	-	-	-	-	-	X
(id, L, t, a_q, a_q)	-	-	-	-	-	-	X	X	-
(id, L, t, a_c, a_c)	-	-	-	-	-	-	-	-	X
(id, L, t, a_c, a_q)	-	-	-	-	-	-	X	X	X
(id, A, t)	-	-	-	-	-	-	-	-	-
(id, A, t, a_q)	-	-	-	-	-	-	X	X	-
(id, A, t, a_c)	-	-	-	-	-	-	-	-	X
(id, A, t, a_q, a_q)	-	-	-	-	-	-	X	X	-
(id, A, t, a_c, a_c)	-	-	-	-	-	-	-	-	X
(id, A, t, a_c, a_q)	-	-	-	-	-	-	X	X	X

Figure 3.11: Which tasks can be applied to which scenarios

Scenarios	Techniques			
	A	C	GRM	STC
(id, A, t)	X	-	-	X
(id, A, t, a_c)	X	-	-	X
(id, P, t, a_c)	-	X	-	-
(id, P, t)	-	X	-	X
(id, P, t, a_q)	-	X	X	-
(P, t)	-	-	X	X
(P, t, a_c)	-	-	X	X
(P, t, a_q)	-	-	-	X
(P, t, a_c, a_q)	-	-	-	X
(id, A, t, a_q)	-	-	-	X
(id, L, t, a_q)	-	-	-	X
(id, L, t, a_c)	-	-	-	X
(id, A, t, a_q, a_q)	-	-	-	X
(id, A, t, a_c, a_c)	-	-	-	-
(id, L, t, a_q, a_q)	-	-	-	-
(id, L, t, a_c, a_c)	-	-	-	-
(P, t, a_q, a_q)	-	-	-	-
(id, L, t, a_c, a_q)	-	-	-	-
(P, t, a_c, a_c)	-	-	-	-
(id, P, t, a_q, a_q)	-	-	-	-
(id, P, t, a_c, a_c)	-	-	-	-
(id, P, t, a_c, a_q)	-	-	-	-
(id, A, t, a_c, a_q)	-	-	-	-

Figure 3.12: What scenarios each technique can represent

Tasks	Techniques			
	A	C	GRM	STC
Task 3.2.12	X	-	-	-
Task 3.2.9	-	X	-	-
Task 3.2.5	-	-	X	-
Task 3.2.6	-	-	X	-
Task 3.2.4	-	-	X	X
Task 3.2.7	-	-	-	X
Task 3.2.10	-	-	-	X
Task 3.2.11	-	-	-	X
Task 3.2.8	-	-	-	X

Figure 3.13: What analytical tasks and each technique can do

Scenarios	Tasks		
	3.2.4	3.2.5	3.2.6
(P, t)	STC and GRM	GRM	GRM
(P, t, a_q)	STC	-	-
(P, t, a_c)	STC and GRM	GRM	GRM
(P, t, a_q, a_q)	-	-	-
(P, t, a_c, a_c)	-	-	-
(P, t, a_c, a_q)	STC	-	-

Figure 3.14: Relation between scenarios, techniques and tasks 3.2.4, 3.2.5 and 3.2.6

Scenarios	Tasks
	3.2.8
(P, t, a_q)	STC
(P, t, a_q, a_q)	-
(P, t, a_c, a_q)	STC

Figure 3.15: Relation between scenarios, techniques and task 3.2.8

Scenarios	Tasks	
	3.2.10	3.2.11
(P, t, a_q)	STC	NA
(P, t, a_c)	NA	STC
(P, t, a_q, a_q)	-	NA
(P, t, a_c, a_c)	NA	-
(P, t, a_c, a_q)	STC	STC

Figure 3.16: Relation between scenarios, techniques and tasks 3.2.10 and 3.2.11

Scenarios	Tasks
	3.2.7
(id, P, t, a_q)	-
(id, P, t, a_q, a_q)	-
(id, P, t, a_c, a_q)	-
(id, L, t, a_q)	STC
(id, L, t, a_q, a_q)	-
(id, L, t, a_c, a_q)	-
(id, A, t, a_q)	STC
(id, A, t, a_q, a_q)	STC
(id, A, t, a_c, a_q)	-

Figure 3.17: Relation between scenarios, techniques and task 3.2.7

Scenarios	Tasks	
	3.2.9	3.2.12
(id, P, t, a_q)	C	NA
(id, P, t, a_c)	NA	-
(id, P, t, a_q, a_q)	-	NA
(id, P, t, a_c, a_c)	NA	-
(id, P, t, a_c, a_q)	-	-
(id, L, t, a_q)	-	NA
(id, L, t, a_c)	NA	-
(id, L, t, a_q, a_q)	-	NA
(id, L, t, a_c, a_c)	NA	-
(id, L, t, a_c, a_q)	-	-
(id, A, t, a_q)	-	NA
(id, A, t, a_c)	NA	A
(id, A, t, a_q, a_q)	-	NA
(id, A, t, a_c, a_c)	NA	-
(id, A, t, a_c, a_q)	-	-

Figure 3.18: Relation between scenarios, techniques and tasks 3.2.9 and 3.2.12

4

Pixel Based Techniques

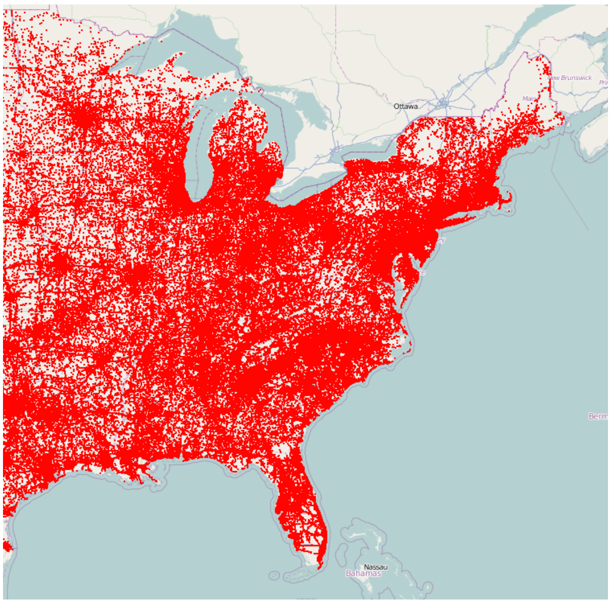


Figure 4.1: A map depicting all the traffic deaths on the east coast of the U.S for the period 2002 and 2012.

This chapter defines the proposal of two pixel based spatiotemporal techniques, meaning that both techniques determine each pixel individually, depending on the context and data. The pixels in turn result in a visualization whose analytical meaning is determined by the way the pixels were created. The first technique is the attenuation & accumulation technique, referred as AA-Map for short and is explained in section 4.1. The second technique is the overlapping spatiotemporal windows technique, shorthanded as OST-Windows and is described in section 4.2. Both techniques were applied to the (P, t) data scenario using two different datasets. The first dataset consists of vehicular fatalities in the US between 2002 and 2010, registered hourly, taken from [Gov]. The second dataset was taken from Instituto da Conservação da Natureza e das Florestas (ICNF) [ICN] and consists of all the fires in Portugal that occurred between 2001 and 2012, registered hourly. The result of painting a map with each event of the dataset results in figures 4.1 and 4.2 respectively. As can be seen there is not much information that can be gathered from both images. Both techniques showed in this chapter will be able extract more knowledge of the datasets than what can be obtained with these images.

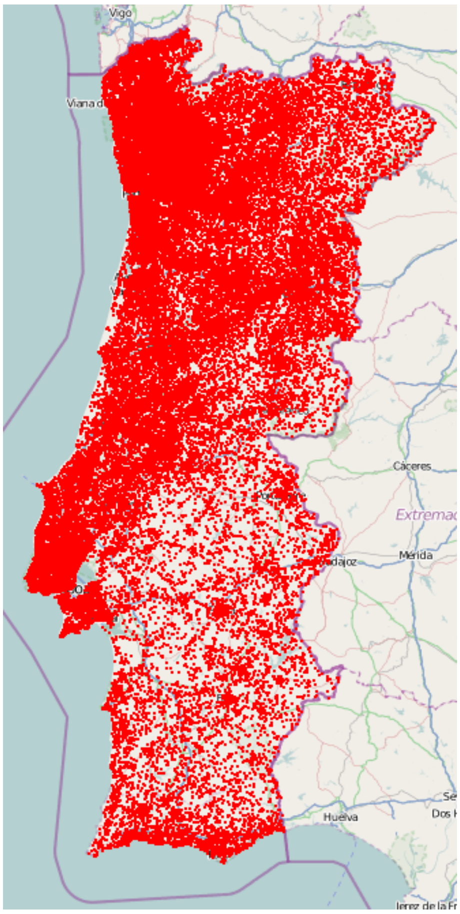


Figure 4.2: A map depicting all the fires in Portugal for the period 2002 and 2012.

4.1 Attenuation & Accumulation Map

Most visual techniques have conceptual, almost metaphorical, ideas behind them. These ideas define the basis of a technique and shape what can be viewed and analyzed when using them. The technique presented in this section is called attenuation & accumulation map technique, refered as AA-Map, and was developed as part of a Programa de

Introdução à Investigação Científica (PIIC) project by Catarina Albino.

Initially the idea behind this technique was to show the lasting effects that past events have in current times. At its core, the idea for this technique can be compared to the following situation: imagine a field of grass, over time people have walked over this field leaving marks on the grass, footprints. When looking at each patch of grass on this field, three different things can be seen:

- First, patches of dirt where grass doesn't grow anymore. These are places that people have walked on, all the time and the high accumulation of people walking over it as lead to this situation.
- Second, patches of tall, unpressed grass, these are places where people have walked on a long time ago, and their impact on the grass as attenuated to almost nothing.
- Third, patches of pressed grass, these are places where some people have walked over recently, or where a lot of people walked a long time ago, due to a mix of accumulation and attenuation of the people's presence.

Taking this concept over to a map, considering the (P, t) data scenario, each pixel in the map will be our patches of grass, with each pixel having a series of events(footprints) that occurred on that position. To replicate the footprint on the grass effect we will paint each pixel with the same color but with different opacities. An empty pixel is equivalent to a patch of tall unpressed grass, a half opaque pixel is the same as a patch of pressed grass and finally an opaque pixel has the same meaning as a patch of dirt.

In figure 4.3 is a map about deadly accidents in the east coast of the U.S, painting the events green using the AA-Map technique. In contrast with figure 4.1, this image can be used to reach some conclusions. For instance, high concentrations of accidents easily stand out as big green clusters, with most coinciding with city centers. More dangerous roads stand out as well and can be identified by the green lines uniting the clusters, thankfully some lines are faded meaning that they had accidents a while ago but not anymore.

Basically, a pixel's opacity will depend on the number of events associated to the pixel's coordinates and the age of those events. In this case a pixel, if opaque meant a lot of events, if light meant a lot of old events or some recent events. Clearly this opacity function is the key to the analysis that the map in figure 4.3 can provide. With this in mind we came up with a method of describing this function formally that makes it easier to define the impact that both attenuation and accumulation have on the conception of the map.

The opacity function, which will be called F, for a set of events $E_1, .., E_k$, that occurred are associated with one pixel and occurred in a time interval $[t_{min}, t_{max}]$, is given by the equation 4.1.

$$F_{[t_1,t_n]}(E_1, .., E_K) = M(A_c(A_t(E_1), .., A_t(E_K)))) \tag{4.1}$$

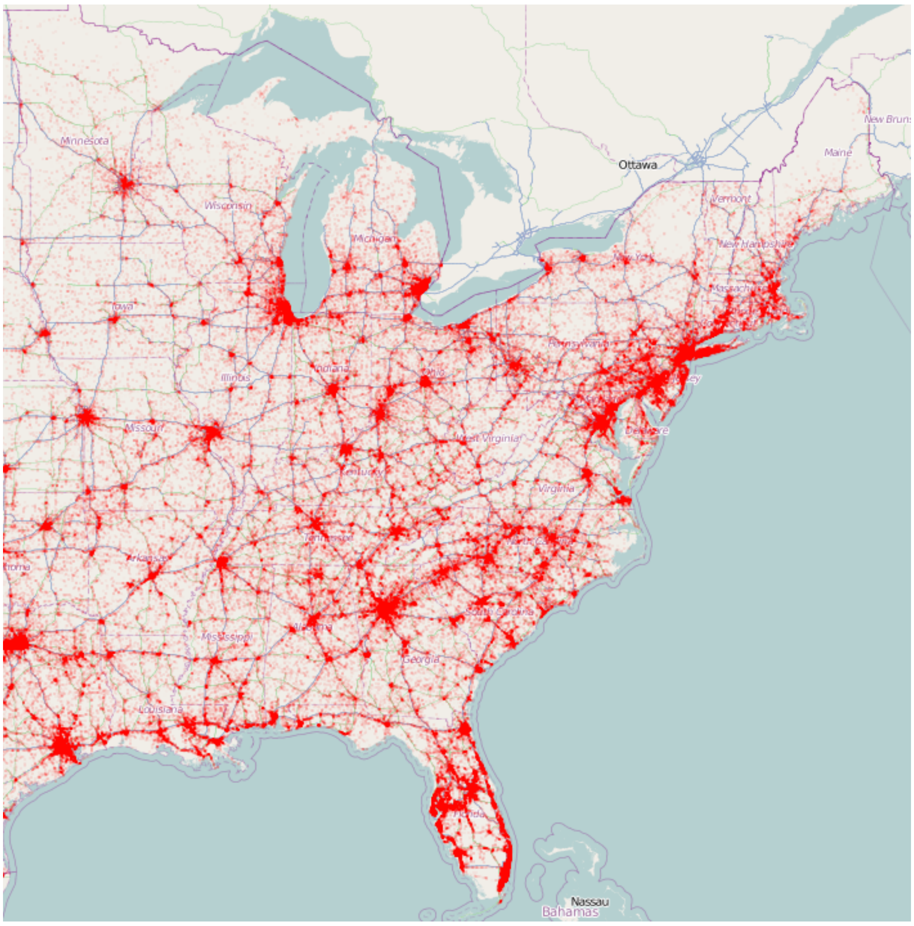


Figure 4.3: An AA-Map showing deadly vehicular accidents on the east coast of the U.S.

The function is composed in three separate functions: an attenuation function (A_t), an accumulation function(A_c) and a mapping(M). An attenuation function receives an event and returns an attenuated value, its the attenuation function that decides which events will have more impact to the analysis. An accumulation function receives a series of attenuated values and returns an accumulated value, this function decides what kinds of concentration are more relevant. Finally, a mapping function receives the accumulated value and translates it to an opacity value. Each of these functions can have one or more parameters, these parameters have a direct impact on a map's analytical context. Fundamentally, the analyses that an AA-Map can provide comes down to the choice of a tuple (A_t, A_c, M), which is fine tuned using each functions parameters.

For the context of this thesis we have considered three different attenuation function, four accumulation functions and one mapping function. In section 4.1.1, the restrictions that an attenuation function has to conform to are explained, followed by a description of each attenuation function that was considered. Section 4.1.2 describes the particularities of an accumulation function and describes each of the accumulation functions we have considered. The mapping function is described in 4.1.3, and finally section 4.1.4 describes the opacity functions, the tuples (A_t, A_c, M), that were considered.

4.1.1 Attenuation Functions

An attenuation function decides how much an event is worth to the analysis, it evaluates an event. Since we're considering just the (P, t) data scenario only the instant t is considered as a function's input, but for other scenarios there could be more input, for instance, in a (P, t, a_q) scenario, the quantitative attribute could also be used to evaluate an event. To facilitate the conception of an attenuation function and its semantic meaning, a few restrictions were imposed on these functions. An attenuation function is applied to one event that occurred in a time interval $[t_{min}, t_{max}]$, and returns attenuated values that are bound to the range $[at_{min}, at_{max}]$.

Each of the three attenuation functions we used is characterized in sections 4.1.1.1 to 4.1.1.3. A characterization consists of a small description that includes: the rational behind the function, its mathematical properties, its impact in an analytical context and the mathematical formula for the attenuation function. A function's formula has one argument, the event's instant and is calculated using the following parameters:

t_{min} - The first instant of the time interval.

t_{max} - The last instant of the time interval.

at_{min} - The minimum attenuated value possible by the attenuation function.

at_{max} - The maximum attenuated value possible by the attenuation function.

4.1.1.1 Linear

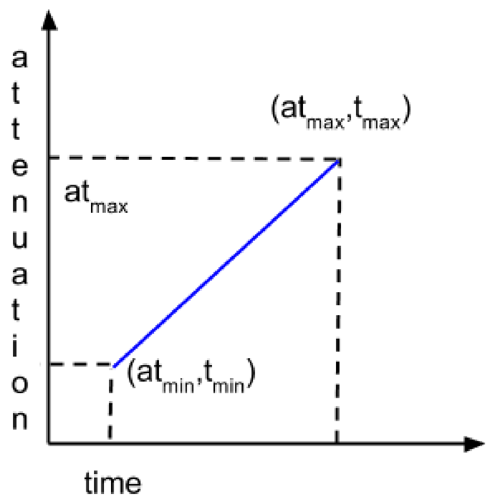


Figure 4.4: A graph describing an example linear attenuation function.

The attenuation is calculated using a linear function. This attenuation function gives a direct proportion from an events time to attenuation value, with younger events having a higher attenuated value then older values. A symmetric linear function can also be done which gives higher attenuated values to older events.

The attenuation is done by mapping t_{min} to the minimum value possible at_{min} and mapping t_{max} to the maximum value possible at_{min}. A symmetric of this function can be done by mapping instead the t_{min} to at_{max} and t_{max} to at_{min}. The attenuation is given by a line defined in the interval $[t_{min}, t_{max}]$ described by the equation 4.2, where m is the slope of the line and b is the y-intercept. The slope is given by the equation 4.3, while the y-intercept is given by the equation 4.4. For the symmetric function the slope is given by the equation 4.5 and the y-intercept is given by the equation 4.6.

$$At(t) = m * t + b \tag{4.2}$$

$$m = \frac{t_{max} - t_{min}}{at_{max} - at_{min}} \tag{4.3}$$

$$b = \frac{at_{min}t_{max} - t_{min}at_{max}}{t_{max} - t_{min}} \tag{4.4}$$

$$m = \frac{t_{max} - t_{min}}{at_{min} - at_{max}} \tag{4.5}$$

$$b = \frac{at_{max}t_{max} - t_{min}at_{min}}{t_{max} - t_{min}} \tag{4.6}$$

4.1.1.2 Constant

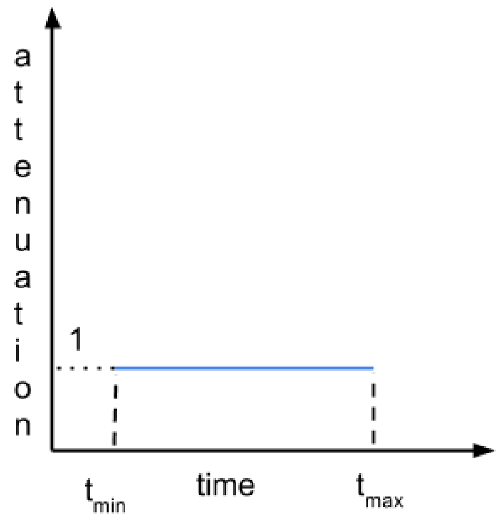


Figure 4.5: A graph describing the constant attenuation function.

The opacity is always the same value for all inputs, returning always 1. So $at_{max} =$

$at_{min} = 1$, making the formula for this function. This attenuation function ignores the time component and concedes the analysis to the accumulation and mapping functions.

$$At(t) = 1 \tag{4.7}$$

4.1.1.3 Exponential

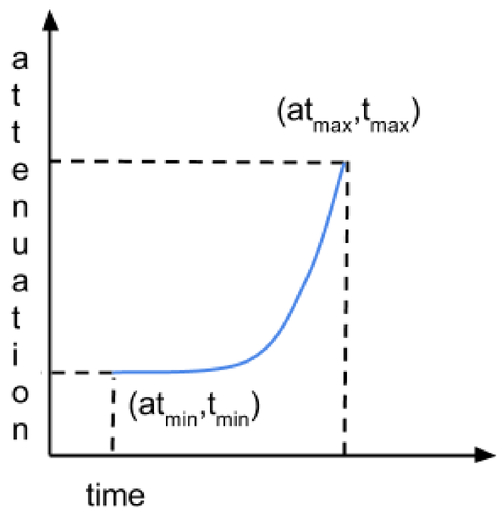


Figure 4.6: A graph describing an example exponential attenuation function.

The opacity is calculated using an exponential function. The result is quite similar to the linear attenuation function, but, because of the characteristics of the exponential function, it gives high attenuation values to a bigger interval of younger events than the linear function. Also like the linear function, a symmetric exponential can be used, giving more attenuation to older events.

The attenuation is done by mapping t_{min} to at_{min} and t_{max} to at_{max}. The exponential attenuation function is given by the equation 4.8 where exponent component r is given by the equation. The symmetric exponential attenuation function is given by the equation 4.10 where its exponent component r is given by the equation 4.11.

$$At(t) = at_{min}e^{(t-t_{min})r} \tag{4.8}$$

$$r = \frac{log(at_{max}) - log(at_{min})}{t_{max}} \tag{4.9}$$

$$At(t) = at_{max}e^{(t-t_{max})r} \tag{4.10}$$

$$r = \frac{log(at_{min} - log(at_{max}))}{t_{max}} \tag{4.11}$$

4.1.2 Accumulation Functions

An accumulation function evaluates a series of attenuated values, and decides which ones are interesting and which ones are not, by assigning high values to interesting series, and low values to uninteresting. Each attenuated value in the series has its values in the range $[at_{min}, at_{max}]$, as defined in section 4.1.1. The returned accumulated value will also be bounded to an interval $[ac_{min}, ac_{max}]$. Four accumulation functions were considered: minimum, maximum, average and sum.

Minimum - returns the minimum value of the attenuated values, so in terms of the attenuated function, returns the least important event in the series. For example, by using the linear attenuation function, the minimum returns the oldest event in the series. The ac_{min} is the same as at_{min}, and ac_{max} is the same as at_{max}.

Maximum - returns the maximum value of the series, it returns the most important event as considered by the attenuated function. For instance, with the exponential attenuation function, the maximum returns the youngest event in the series. The output range $[ac_{min}, ac_{max}]$ is $[at_{min}, at_{max}]$.

Average - returns the average of the series. In the long run, this accumulation function evaluates the series using the same criteria as the attenuation function, by applying it to the average of the series. Considering for example the symmetric linear function, if the average of the events is old, the accumulated value will be high, and if the average is young the accumulated value will be low. The output range $[ac_{min}, ac_{max}]$ is $[at_{min}, at_{max}]$.

Sum - returns the sum of all attenuated values in the series. This accumulation function focuses on concentration of events discriminated by the attenuated function. For instance, consider three different series: A,B and C. Series A has one low value, series B has 10 low values and series C has 10 high values. Applying this function, C would have an accumulated value higher than B's and A's, and B's accumulated value would be higher value A's. Output range is $[at_{min}, +\infty]$, so ac_{min} is at_{min} and ac_{max} is $+\infty$.

4.1.3 The Mapping function

The mapping function is what ultimately decides the importance of each pixel to the analyses, it receives an accumulated value and maps it to the range of possible opacity values, making it the last step in the calculation of a pixel's opacity. The input of a mapping function belongs to the range of values $[ac_{min}, ac_{max}]$, and outputs an opacity value that is in a range of possible opacity values. This range of opacity values ranges from 0, which defines an empty pixel, to op_{max}, which defines a completely opaque pixel, this op_{max} is a constant that depends on the color space that is considered.

The mapping function is a straightforward linear mapping from the range $[ac_{min}, ac_{max}]$ to the range $[0, op_{max}]$, assigning 0 to ac_{min} and op_{max} to ac_{max}, and linearly mapping the values between. However, some accumulated values might not be interesting enough, have too low an accumulated value to even appear in a map. Also, some accumulated value may be exceptionally high, outliers, that by mapping to the maximum opacity will disappear all the other events. So a range of values $[k_{min}, k_{max}]$ is used instead of the range $[ac_{min}, ac_{max}]$ to map the accumulated values to opacity values. This way k_{min} is mapped to 0 and k_{max} is mapped to op_{max}

The bounds of the output range,k_{min} and k_{max}, are the parameters of the mapping function and changing them has the following effects. Figure 4.7 shows visually the effects of the k parameters on the opacity attributed to a pixel. This parameters allow the user to define some thresholds where information is not interesting to him. Setting the k_{min} to a value higher than ac_{min} will remove any accumulated that are too low from the map. On the other hand, setting the $kmax$ value to a lower value than ac_{max} allows the user to ignore the effects that any outliers may have on the visualization. In other words, the user may not be interested in the pixel with the highest accumulated value, but a group of pixels with high accumulated values.

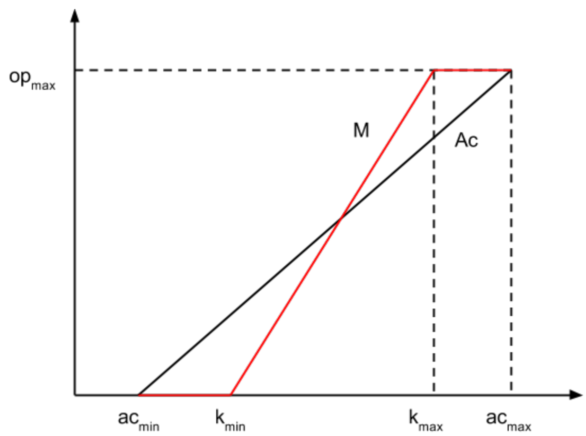


Figure 4.7: A visual representation of the relation between the mapping function and the accumulation function.

Finally, the mapping function is given by the equation 4.12

$$f(ac) = \begin{cases} 0 & \text{if } ac < k_{min} \\ \dfrac{k_{max} - k_{min}}{op_{max}} \times ac + \dfrac{-k_{min}op_{max}}{k_{max} - k_{min}} & \text{if } ac \in [k_{min}, k_{max}] \\ op_{max} & \text{if } ac > k_{max} \end{cases} \quad (4.12)$$

4.1.4 Analytical Settings

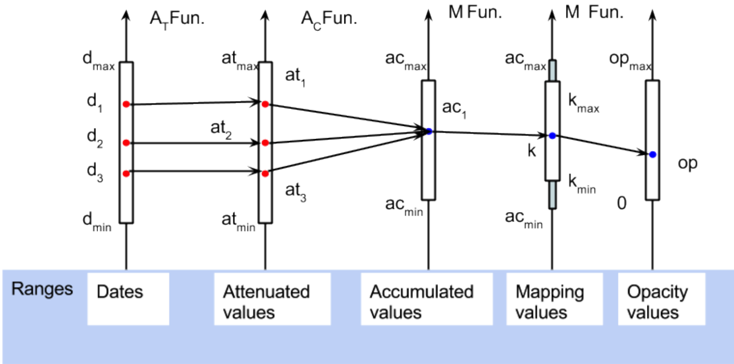


Figure 4.8: A diagram depicting the calculation of a pixel's opacity from the events that occurred at a pixel's location.

An opacity function is a tuple of one attenuation function, one accumulation function and one mapping function, that will give a certain analytical meaning to an AA-Map. The process of the opacity function can better be described by the diagram in figure 4.8, which depicts how the function behaves when evaluating a series of events of one single pixel. The function starts with a series of event instants, t_1, t_2, t_3 that are valued in the $[t_{min}, t_{max}]$ range, each of these values is attenuated using an attenuation function A_T to the range $[at_{min}, at_{max}]$. Afterwards, that series of attenuated values at_1, at_2, at_3 is accumulated to one accumulation value ac_1 using the accumulation function A_C. The accumulated value is in the range $[ac_{min}, ac_{max}]$ and is capped to the range $[k_{min}, k_{max}]$ as part of the mapping function M. Finally, as the second step of the mapping function the capped value k is mapped to the opacity range $[0, op_{max}]$ and assigned its opacity op.

The following sections describe the combinations of functions that were found to have some analytical meaning. The description of each function is done by describing which attenuation and accumulation functions are used, the mapping function is not specified because it is always considered the same function. Each function has a list of available parameters that will influence the opacity function, these parameters come from the selected attenuation and accumulation functions as well as the mapping function. The last component of the functions definition is a small description of how the selected functions determine the opacity function's analytical interest, and how the parameters affect a possible analysis. For each opacity function, a map is shown depicting the use of that opacity function to the fires dataset for the year 2012.

4.1.4.1 Counting Events

Attenuation Function - Constant.

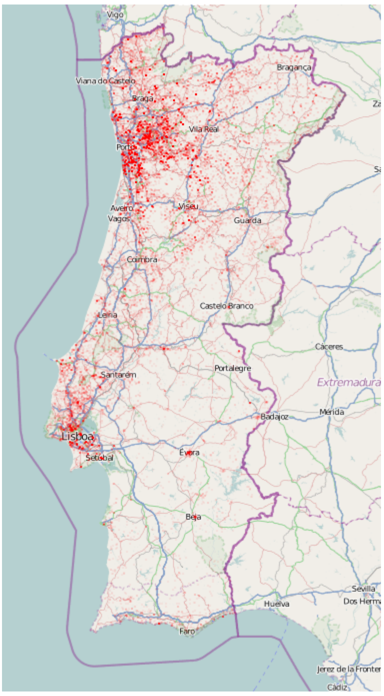


Figure 4.9: The effects of using the counting events opacity function on the fires dataset, for the year 2012. ($k_{min} = 0, k_{max=20}$)

Accumulation Function - Sum.

Available Parameters - From mapping function k_{min}, k_{max}.

Description - Considering that every event is attenuated as 1 and each pixel's series of attenuated events will be summed, each pixel's accumulated value will be the count of the events that occurred in that pixel. Depending on the k_{min} and k_{max} parameters, with this function, a pixel's opaqueness is directly related to the number of events, meaning that a more opaque pixel is equivalent to a pixel with more events.

Parameter effects on task

k_{max} - The k_{max} parameter determines the number of events that constitute an opaque pixel, so if set to 10, a pixel has to have 10 events to be opaque.

k_{min} - If greater than 0, this parameter determines the minimum number of events that need to occur in a pixel for it to be visible. If negative or 0, it sets a minimum opaqueness for points where the concentration of events is too low. The particular case of $k_{min} = k_{max}$ will show just the group of pixels where k_{min} (or k_{max}) events occurred. For instance, figure 4.9 shows concentrations of 20 or more fires for the year 2012 in the northern regions of Portugal and around Lisbon.

4.1.4.2 First Events

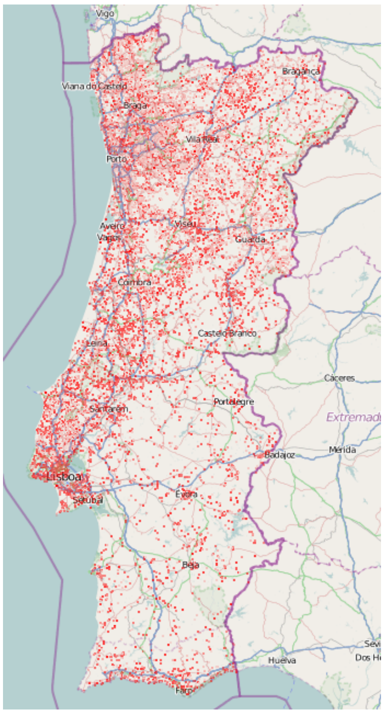


Figure 4.10: The effects of using the first events opacity function on the fires dataset, for the year 2012. (Linear,$at_{min} = 0$,$k_{min} = 0$,$k_{max} = 256$,$at_{max} = 256$)

Attenuation Function - Linear or Exponential.

Accumulation Function - Minimum.

Available Parameters - From mapping function k_{min}, k_{max}, from attenuation function at_{min}, at_{max}.

Description - For each pixel's series of events the one with the minimum attenuated values of the whole series will be the value considered for the mapping function. The linear attenuation assures that the oldest of the events in the series will get the lowest attenuation value, in other words the mapping function evaluates the first event that occurred in that series hence it evaluates the first event that occurred in each pixel. The mapping function will assign opacity values directly related to the age of the first event. With opaque pixels meaning the first event occurred recently, and almost transparent pixels meaning that the first event occurred a long time ago. Exponential may also be used, giving more relevance to younger events. The map in figure 4.10 shows a dark colored region in Lisbon, meaning it had its first fires in 2012 very late in the year.

Parameter effects on task

k_{max} - Decreasing this parameter will increase the range, starting from t_{max} that get the maximum opacity. In other words, a wider range of first events will have maximum opacity. Increasing the parameter past at_{max} will result in no events getting the maximum opacity.

k_{min} - Increasing this parameter causes pixels where the first event occurred near the beginning of the analyzed interval, or rather near t_{min}, to not show in the map. While, decreasing this parameter will make pixels where the first event occurred near t_{min} to become less transparent.

$[at_{min}, at_{max}]$ - The attenuation range, thats defined by these parameters, determines the granularity in opaqueness between old and young events. In other words, a wide attenuation range will allows for a more detailed choice using the k_{min} and k_{max} parameters then if the attenuation range was slimmer.

4.1.4.3 Last Events

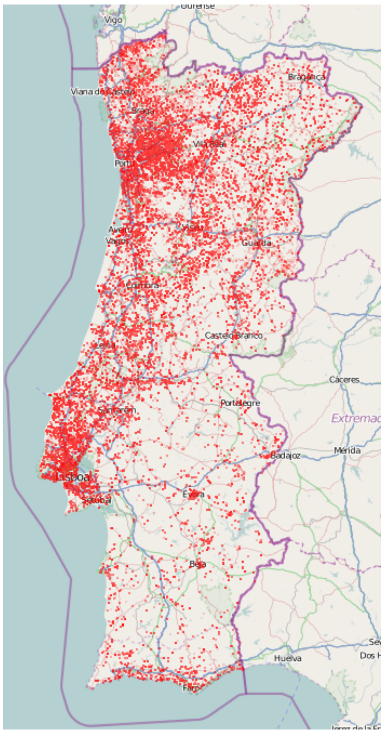


Figure 4.11: The effects of using the last events opacity function on the fires dataset, for the year 2012. (Linear, $at_{min} = 0$, $k_{min} = 0$, $k_{max} = 256$, $at_{max} = 256$)

Attenuation Function - Linear or Exponential.

Accumulation Function - Maximum.

Available Parameters - From mapping function k_{min}, k_{max}, from attenuation function at_{min}, at_{max}.

Description - For each pixel's series of events the one with the maximum attenuated values of the whole series will be the value considered for the mapping function. The linear attenuation assures that the youngest of the events in the series will get the highest attenuation value, in other words the mapping function evaluates the last event that occurred in that series hence it evaluates the last event that occurred in each pixel. The mapping function will assign opacity values directly related to the age of the last event. With opaque pixels meaning the last event occurred recently, and almost transparent pixels meaning that the last event occurred a long time ago. Exponential may also be used, giving more relevance to younger events. Figure 4.11 overall dark redness, shows that most of the last fires of 2012 happened at the end of that year.

Parameter effects on task

k_{max} - Decreasing this parameter will increase the range, starting from t_{max} that get the maximum opacity. In other words, a wider range of last events will have maximum opacity. Increasing the parameter past at_{max} will result in no events getting the maximum opacity.

k_{min} - Increasing this parameter causes pixels where the last event occurred near the beginning of the analyzed interval, or rather near t_{min}, to not show in the map. While, decreasing this parameter will make pixels where the last event occurred near t_{min} to become less transparent.

[at_{min}, at_{max}] - The attenuation range, thats defined by these parameters, determines the granularity in opaqueness between old and young events. In other words, a wide attenuation range will allows for a more detailed choice using the k_{min} and k_{max} parameters then if the attenuation range was slimmer.

4.1.4.4 More Recent Events

Attenuation Function - Linear or Exponential.

Accumulation Function - Average.

Available Parameters - From mapping function k_{min}, k_{max}, from attenuation function at_{min}, at_{max}.

Description - Each pixel's opacity will be related to the average of attenuated values. Since the linear attenuation assigns high values to young events and low values to old events, so this opacity function gives relevance to series that have more young events then old events. In other words, pixels, where the average of its events is recent, are opaquer than pixels where the average of its events is older. Since the opaquest pixels have in average younger events than older events, these opaque pixels have more recent events than old events. The exponential attenuation may

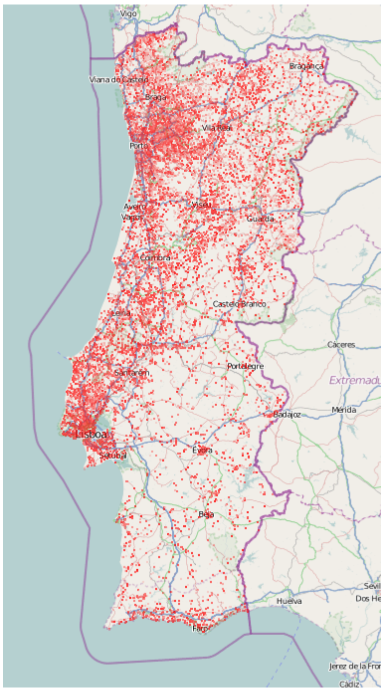


Figure 4.12: The effects of using the more recent events opacity function on the fires dataset, for the year 2012. (Linear, $at_{min} = 0$, $k_{min} = 0$, $k_{max} = 256$, $at_{max} = 256$)

also be used, giving less relevance to older events then if using the linear function. The city of Lisbon is shown in figure 4.12 as having fires more in the end of 2012 then in the beginning of that year, this was already confirmed by conjunction of figures 4.10 and 4.11, that showed the last and first fires of Lisbon occurring in the end of the year.

Parameter effects on task

k_{max} Decreasing this parameter bellow at_{max} defines a wider range of series of events that are considered to have more recent events than old.

k_{min} Increasing this parameter above at_{min} will increase the number series that are considered to not have enough .

[$at_{min}at_{max}$] The size of the attenuation range defines the granularity of the k parameters. For instance, if the attenuation range is widened and the k_{max} is decreased bellow at_{max} as described earlier, it would include less series of events than if the parameter was changed before the attenuation range was widened.

4.2 Overlapping Spatiotemporal Windows

Most techniques, like change maps, or chorems, almost ignore the temporal component's granularity, and are applied only to two different instants of time. This leads to a lot of data being presented to the user with almost no temporal relevance, since everything that occurs between the two instants is not shown by the technique. On the other hand, techniques like the space time cube or the growth ring maps show a lot of the temporal component's granularity, which makes them good at comparing two different regions at the same time interval.

The second technique proposed in this document was envisioned to remove one simple restriction that was found on all techniques, the ability to compare different spatiotemporal regions, meaning the ability to compare two different regions at two different time intervals. This objective was achieved with the overlapping spatiotemporal window concept. A OST-Window is a pair (geographical boundary, time interval) that when translated to a visual representation works like a window that is placed over the map, and allows the user to see a specific interval of time and space. The window defines a geographical boundary over the map, inside this boundary is shown all the spatiotemporal data that occurs in a certain time interval, as can be seen in figure 4.13, the window is placed over the states of Washington and Oregon, so it only shows data in that area, for the time interval between 2002 and 2010. For the purpose of this thesis the data inside the window is presented using the AA-Map concept presented in section 4.1, but other forms of data presentation may be used.

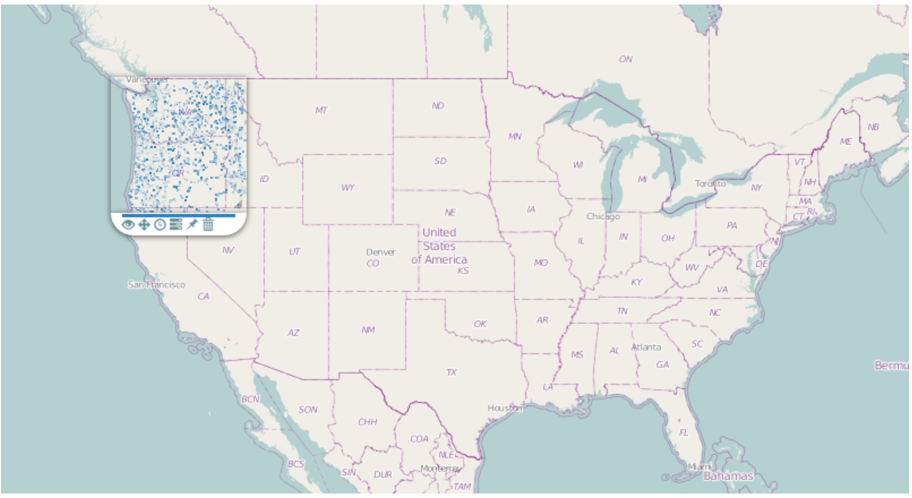


Figure 4.13: A OST-Window showing deadly accidents in the U.S. between 2002 and 2010.

For one OST-Window there is a set of available operations, the window can be moved around or resized changing its geographical boundaries, changing the focus of an analysis to different regions. With respect to the time component, a OST-Window shows what happens in a contiguous time interval, this time interval can increased, decreased or moved.

The regular operations of a map, zooming and panning, can affect the window's geographical position. There are two possible behaviors for a OST-Window when the surrounding map is zoomed or panned. In the first approach the window keeps its size and position considering its geographical position, so if a window is covering a one km square with a city at its middle, if the map is zoomed out, the window will become smaller on screen, but will still be centered on the city and covering only a one km square on the map. In the second approach the window keeps its size and position relative to the screen, for instance, if a window is occupying a 100 pixels square on screen, when the user zooms out the window will still be a 100 pixels square. When the window keeps its geographical position it is said to be in the pinned state, and when the window keeps its screen position it is said to be in the unpinned state. In the prototype that was implemented, and is presented in chapter 5, the pin and unpin state was implemented for zoom operation, while for the pan operations the windows are always unpinned.

The data presented inside each OST-Window is done using the AA-Map technique, so a base color has to be chosen for its visual representation, and it can be changed. When two overlapping spatiotemporal windows are used on the same map, the choice of each window's color is affected by the situation they're being used for. There are two different situations relevant for the use of two OST-Windows that affect the choice of color, which are:

- First, analyze two different regions over the same, or different time periods. In this situation the color used should be the same, to facilitate comparisons.

- Second, analyze the same region, or intersecting regions, over different time periods. In this situation, different colors should be used to distinguish different windows.

4.2.1 Overlay Operations

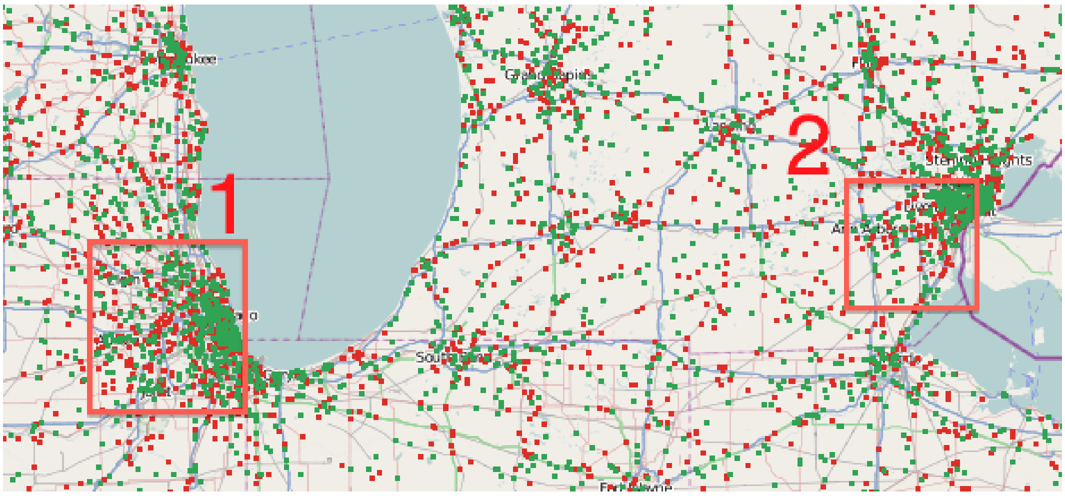
The analytical context of more than one OST-Window raises a new problem: what to do to a pixel when two different OST-Windows are going to paint that pixel? The resulting pixel will have an impact on the analysis that the map can provide, it can emphasize one window over the other, its differences, or similarities. There are several possible approaches, each with its own analytical value. In the following sections, three of this overlay operations are described, each with different analytical intent. There are more possible operations that may be relevant, and should be investigated, however the considered operations have given us a lot of possibilities.

4.2.1.1 Order decides

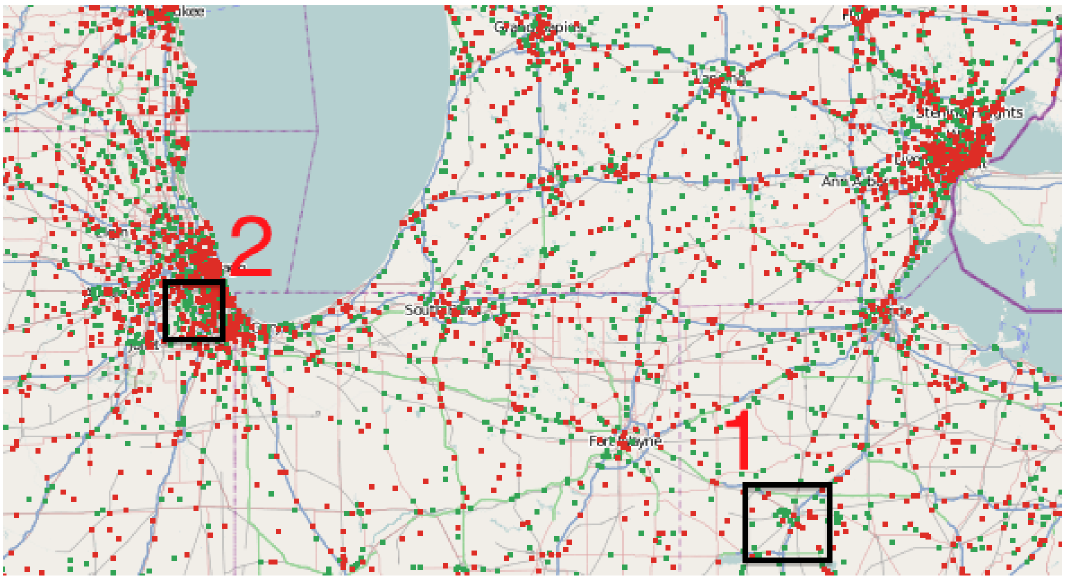
In this overlay operation, a pixel's visual properties (color and opacity) are decided by the order of the OST-Windows, when two windows are painting the same pixel the one on top decides how the pixel is painted.

For example, take two OST-Windows A and B that cover the same region and different distinct time periods, with A on top of B. With this operation, all of the events of window A will be shown on the map hiding any events of OST-Window B that occurred in the same place as A. However, the events that are shown in B will be events that occurred in places where no events during the time period A has occurred.

Figure 4.14 shows the effects the order has on the analyses. Two years of deadly accidents are depicted over the state of Maine, 2002 and 2003, painted green and red respectively. The year 2002 is placed over 2003 in 4.14(a) and some regions can be identified by having had a lot of accidents in the year 2003 but no accidents in the year 2002. This areas can be seen inside the areas marked with 1 and 2, and are characterized by the number red pixels. In figure 4.14(b), the year 2003 is placed on top showing other regions, marked with 1 and 2, where there were accidents in 2002 but not 2003.



(a) 2002 (green) over 2003 (red)



(b) 2003 (red) over 2002 (green)

Figure 4.14: Representation of the “order decides” overlay operation.

4.2.1.2 Paint the common pixels

The previous overlay operation focused on finding events that occurred in one time period but not another. This overlay operation, however, focuses on painting the common pixels in a third color derived from the colors of both windows. The method that was used sums the rgb value of both colors, so when using this operation it is better to use primary colors. The opacity levels are summed as well.

Considering, for instance, two OST-Windows A and B that cover the same region at different non-intercepting time periods. The A window is painted red(rgb=$(100\%, 0, 0)$) and the B window is painted yellow(rgb=$(0, 100\%, 0)$), this results in conflicting pixels being painted in yellow (rgb=$(100\%, 100\%, 0)$).

Those conditions were used for the map presented in figure 4.15, which displays, in yellow, the places where accidents have occurred in both 2002 and 2003. The same patterns from the "order decides" operation can be detected as well, notice the area marked with a 1, the cluster of red pixels that could be seen in 4.14(a) is still present. However, the cluster of yellow pixels in the marked region also denotes areas where a lot of accidents happened in both periods, showing that this operation gives more attention to the similarities rather than the differences between time periods.

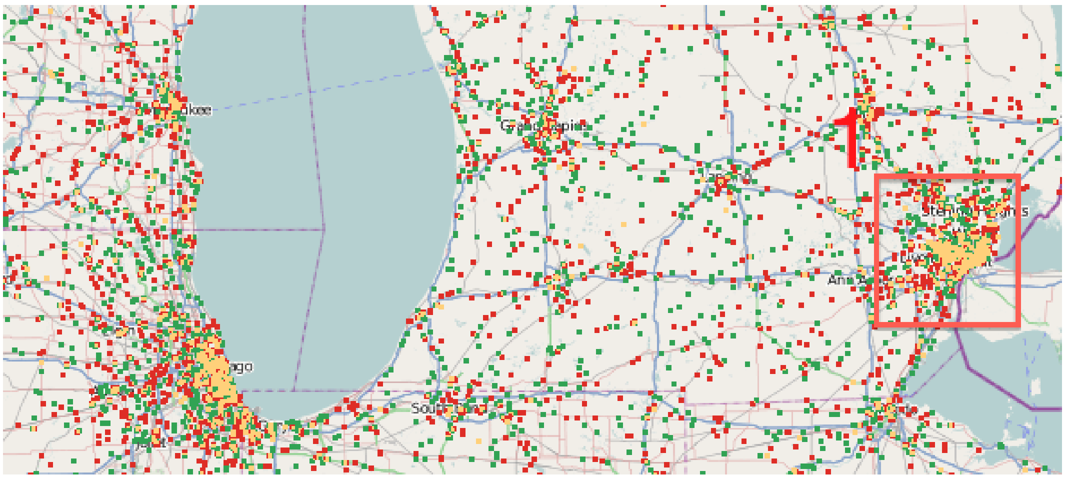


Figure 4.15: Representation of the "paint the common pixels" overlay operation.

4.2.1.3 Remove the common pixels

The last overlay operation has the opposite objective to the previous operation. When two OST-Windows paint the same pixel, the pixel is ignored and it is not painted at all. This way the similarities between both time periods are removed showing only the differences between them.

For instance, considering two OST-Windows A and B that cover the same region at different non-intercepting time periods. As before, windows A and B are painted red and green respectively. When both windows paint over the same pixel that pixel is cleared, this way any pixel that is shown in red represents a place where there are events that occurred in window A's time period, but there were no events occurring in window B's time period.

The map shown in figure 4.16 represents the same situation as in the previous sections, but applying the remove the common pixels overlay operation. The visualization may seem quite chaotic with accidents occurring in 2002 and 2003 all over the map. What is interesting its to still find the same clusters of accidents in 2002 and 2003 as before. On the one hand it is easier to see that some clusters of accidents that only occurred in 2003 are isolated from others like in square 2. On the other hand, clusters of accidents that occur in one year may be close to accidents that only occurred in the other like in squares 1 and 3.

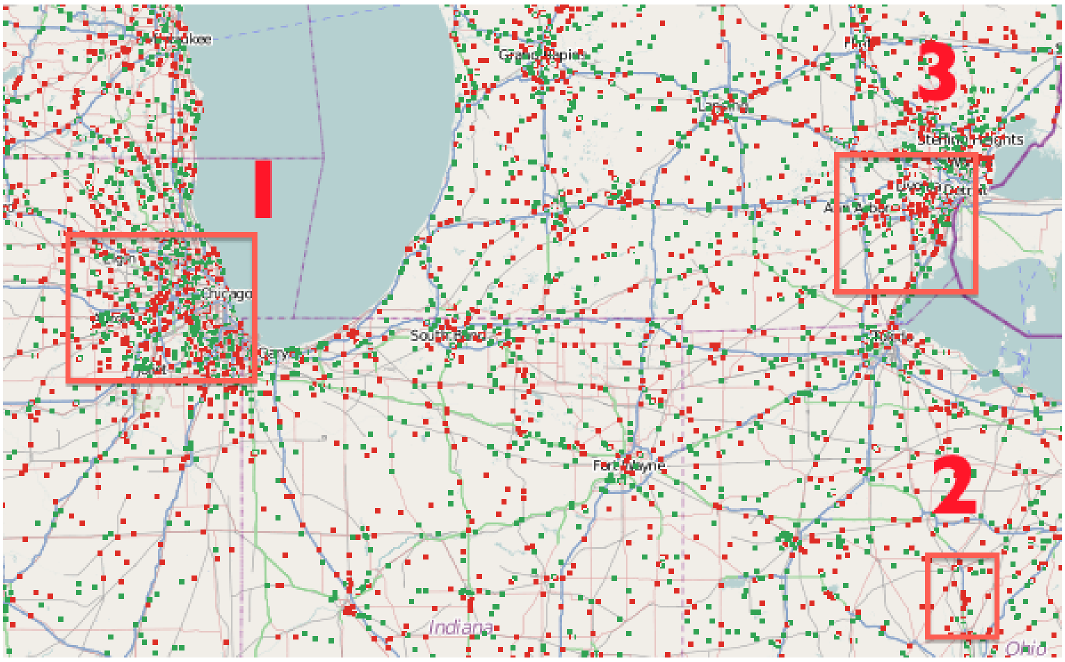


Figure 4.16: Representation of applying the "remove the common pixels" overlay operation.

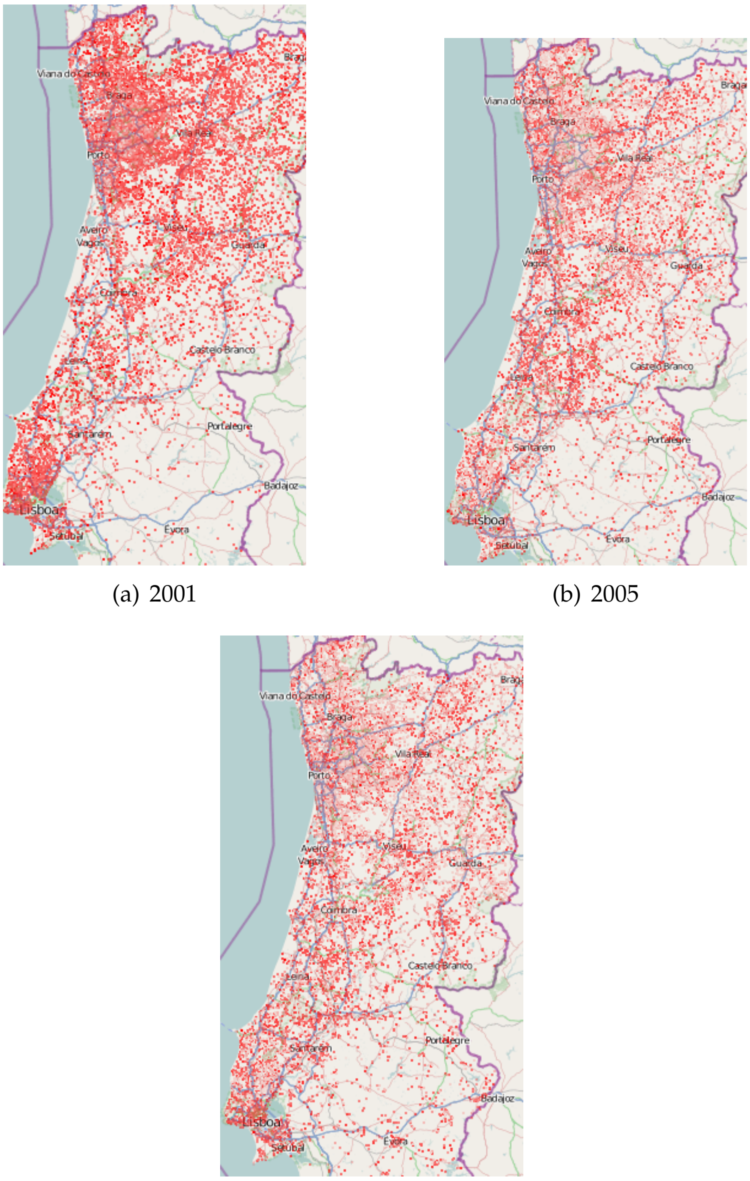
4.3 Conclusion

The techniques that are proposed in this chapter are part of the same context as the other techniques that were presented in earlier chapters. Both focus on event scenarios, mainly the (P, t) scenario, but may be extended to other scenarios. According to the Catalog, as seen in figure 3.12, for the (P, t) scenario only the growth ring map, and the space time cube are applicable techniques. However, the possible analytical tasks that can be done on that scenario with the techniques proposed in this chapter are completely different from the ones that can be done with the space time cube and the growth ring map.

For instance the AA-Map technique, as seen in figures 4.17, shows the effects of applying the technique with the *First Events* opacity function to the fires dataset, to the years 2001, 2005 and 2012. It shows some relevant information, mainly in the year 2001, when fires started to appear later than in 2005 and 2012. This figure shows an interesting occurrence, that would be rather hard to collect using the space time cube or the growth ring maps technique. The first events in the space time cube, would be the dots closer to the map plane, and in the growth ring maps, the information is too aggregated to retain this kind of information.

Another example, recurring to the OST-Window technique, as used in figure 4.18, to analyze three different time periods at the same time. The similarities and oddities are distinguished between the three periods due to the "paint the common pixels operation". In the referenced figure, the year 2001 is painted blue, the year 2005 is painted green and the year 2012 is painted red. Masses of white pixels mark points where fires occurred in all three years, while masses of complementary colors mark points where fires occurred in two years. Orange pixels are for 2005 and 2012, pink pixels are for 2012 and 2001, while cyan pixels are for 2005 and 2001. The square number 1 identifies an area with a lot of fires in three years with white pixels. The square number 2 identifies an area with a lot of fires in 2005 and 2012 with orange pixels. Lastly the square number 3 identifies with cyan pixels a lot of fires in 2005 and 2001.

The provided analyses in the previous example could be done with other techniques but at some cost. With the space time cube it is not that easy to compare three different time intervals, while the growth ring map does not show the spatial distribution of the events, which is required for this kind of analyses.



(a) 2001

(b) 2005

(c) 2012

Figure 4.17: The first events opacity function applied to the fires dataset, using the linear attenuation function. K_{min} = at_{min} = 0, K_{max} = at_{max} =256

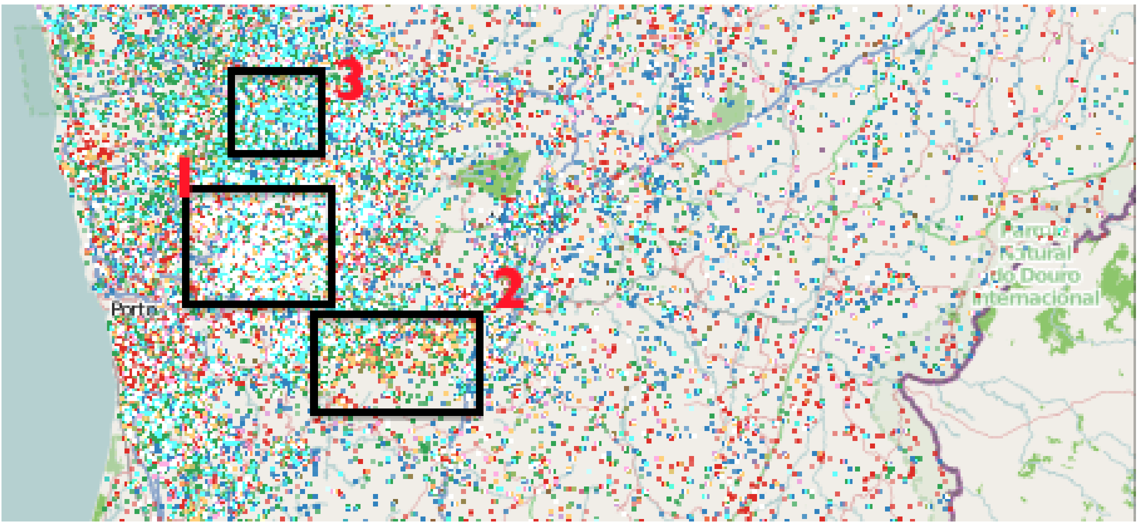


Figure 4.18: The “paint the common pixels” overlay operation applied to the fires dataset with three different OST-windows.

5

Prototype

To test and illustrate both the attenuation & accumulation map technique explained in section 4.1, and the overlapping spatiotemporal window technique explained in section 4.2, an HTML5 web application prototype was developed. The main objective of the prototype is to show and research the analytical and capabilities of both techniques. This imposed a few restrictions on the modeling of the prototype, mainly the prototype needed:

- to handle considerable big datasets, with both of the datasets that were used containing more than 300 thousand entries.
- to be responsive for the main analytical operations of the techniques, meaning that, when a user performs any of these operations the results should be displayed within an acceptable time frame. Those operations, as explained in chapter 4, are: For the AA-Map map:
  - changing the opacity function and its parameters.
  - changing the time interval.

  For the OST-Window technique are:
  - changing the window's time interval.
  - changing the window's geographical boundaries by resizing the window or moving it.
  - changing the overlay operation.

This chapter is divided into three sections. Section 5.1 explains the different technologies and libraries that were used for the development of the prototype. The following section,

section 5.2, explains the implementation of the prototype, presenting the design decisions and the systems architecture. Finally, in section 5.3, the interface of the prototype is presented and explained.

5.1 Technologies & Libraries

For the implementation of the prototype, the more important technologies and libraries used are explained in the following sections.

5.1.1 *LeafletJS*

LeafletJS[lea] is an *OpenSource Javascript* library for the development of HTML5 map applications. It is small in size, with around 30KB, and uses *OpenStreet Maps*[ope] as the map provider, though it allows the developer to change to other map providers.

The library offers a simple Application Programming Interface (API) that covers all the operations needed for developing map applications. It has a layer based paradigm meaning everything that is shown to the user, map, zoom controls, belongs in a layer. A layer can either be attached to the map coordinate system or attached to the screen coordinates. For instance the AA-Map map will be visually represented as layer that is attached to the map coordinate system, while zoom or pan controls are represented as layers that are attached to the screen coordinate system.

5.1.2 HTML Canvas

Initially the HTML standard did not allow the use of drawing primitives, so, developers had to use third party plugins, like flash, to create web applications that required some visual representation. The *Canvas 2D Context* specification [htm] was proposed to dismiss the need for plugins, and provide the HTML standard with a new HTML element that developers can use for drawing operations.

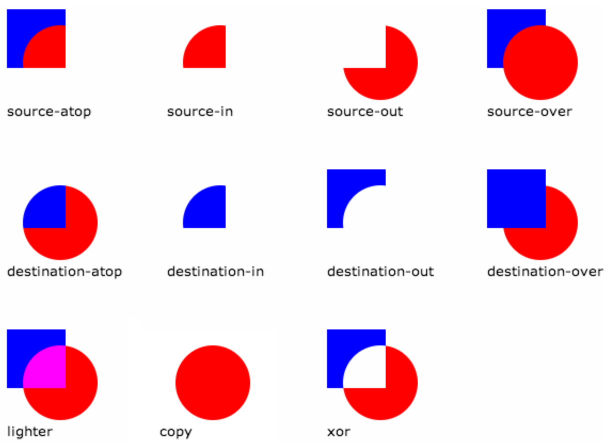


Figure 5.1: The various composite operations of the *Canvas* specification.

The new element, appropriately named the Canvas element, has an associated 2D context which provides an API of drawing primitives. This API can be used to draw simple 2D geometric shapes and images. The *LeafletJS* library presented in section 5.1.1 uses the canvas element for canvas map overlays, a special kind of overlay that is entirely drawn on the client side of the web application. Furthermore, a canvas allows compositing an image from two or more different images, this image composition is based on the choice from 11 available composite operations, depicted in figure 5.1. The more relevant operations for the prototype are the *XOR* operation and the *lighter* operation. The *XOR* operation removes pixels from the composited image that both images have in common, this operation can be used to achieve the *Remove common pixels* overlay operation from section 4.2.1.3. The *lighter* composite operation paints common pixels with by adding the color information of both pixels, this operation was useful to achieve the *Paint the common pixels* overlay operation from section 4.2.1.2.

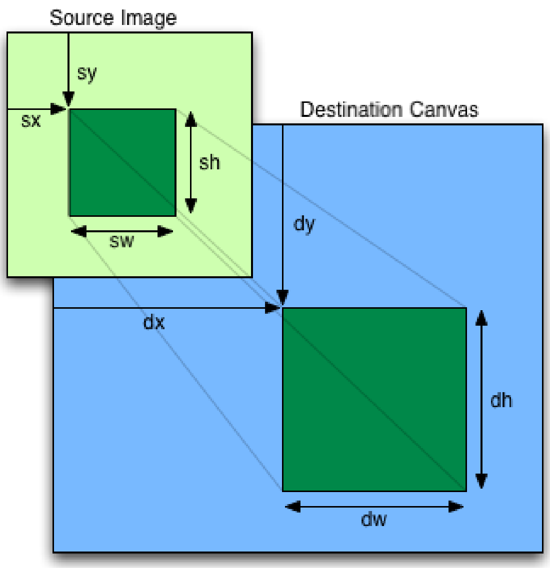


Figure 5.2: A visual representation of the HTML *Canvas* drawImage method, which allows for the cropping of images before drawing them

The specification allows also for the creation of images by supplying pixel information. The pixel information is stored in a *Javascript* object called an *ImageData* object. This object contains the width and height of the image and an array with the pixel information of the image. The pixel array has $4 * width * height$ as its size and stores the information of red, green, blue and alpha for each pixel in groups of 4.

To accomplish the OST-Window drawing, the extended version of the *drawImage* was used. This version with the following signature *drawImage(image, sx, sy, sw, sh, dx, dy, dw, dh)*, allows the cropping of an image at its source and repositioning it at the destination canvas, as can be seen in figure 5.2.

5.1.3 HTML Web Workers

The development of client side web applications has always been limited to the creation of single threaded applications. This causes the use of the same thread to draw the web page and run any logic the application might need, meaning that, any heavy computation will make the UI freeze and become non-responsive. The idea of the web workers specification [web] is to remove this limitation and provide a simple way to bring multi threaded applications to the development of client side applications.

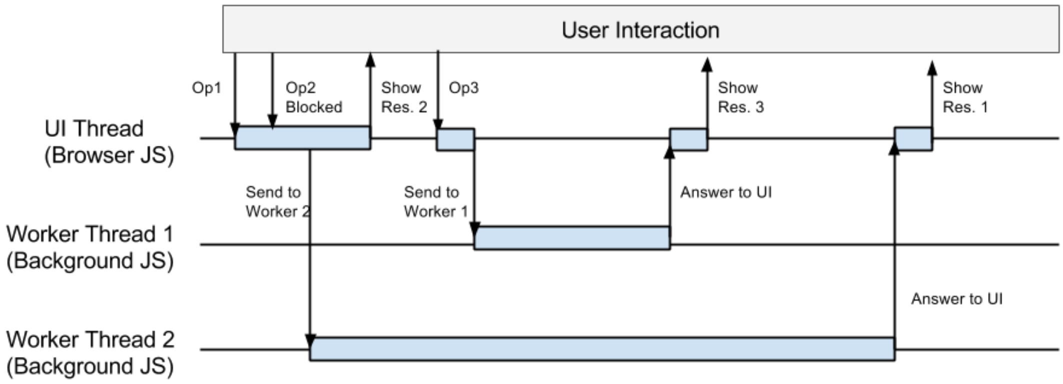


Figure 5.3: A diagram depicting the communication between a UI thread and two worker threads.

A web worker is a script that is run on a thread independent of the main thread also known as the UI thread. There can be more than one worker running in a web application and none of them share any memory between themselves or the main thread, not even the *DOM*. A worker script is supposed to be used for heavy computations and not any kind of DOM manipulation. The communication between the UI thread and the worker thread is done through message passing. The diagram in figure 5.3 depicts the use of two worker threads, how the communication is made, and how it affects the user interaction.

A web worker is spawned in the main thread through the creation of a *javascript* web worker object, by specifying a *URL* to the web worker script. The web worker specification [web] defines one method to pass messages between the UI thread and a worker thread.

The *postMessage* method allows one thread to pass a message to the other thread, this message can be any *javascript* object as long as it does not have any circular references or references to the *DOM* object. On the UI thread the *postMessage* is called on the worker object, and on the worker thread the method is called on the *self* object.

To receive a message, a thread can listen to the *onmessage* event. This event is fired on the worker object in the UI thread when the worker sends a message. In a worker thread the *onmessage* event is fired on the *self* object when the UI thread sends a message to the worker. The message passing between threads is done by cloning the object, so passing big objects may take some time.

5.2 Architecture

This section provides an overall explanation of the prototype's design, architecture and implementation, first, an explanation for the most relevant design decisions in section 5.2.1. Second, a general overview of the system is provided in section 5.2.2. Finally, section 5.2.3 explains the most relevant aspects of the prototype's implementation.

5.2.1 Design Decisions

The prototype was implemented as a *HTML5 / Javascript* application, meaning that it is a web application with a thick client.

The environment web application development is filled with *opensource* libraries and frameworks with a vast community support. Aided by the fast build process of web applications, this environment is ideal for the fast prototyping of ideas. Initially the concepts of both techniques were not fully established, and required a lot of experimenting, so this development environment seemed ideal for the prototype.

The choice between thin and thick client refers to the computational load of the web application, if most of the computations are made in the server side of the application then it is a thin client, if most of the computations are made on the client side then the application is said to be a thick client application. As stated in the introduction of this chapter, the main analytical operations of the techniques need to be responsive. Some of these tasks have a heavy computational load taking a significant amount of time to be performed. Considering the interaction with the *Canvas* element, and the necessary heavy computation with a quick response by the UI element, a thin client model would not be possible. Some of the HTMLspecifications used, like the workers specification, are still candidate recommendations. This means that those specifications are not implemented by all the browsers and the ones that implement them may have different APIs, these APIs can have different behaviors and some even have different names for the same functions. Since the need of having the prototype work in several browsers was not a main priority, the *Chrome* browser was chosen, because it implemented all of the specifications required.

5.2.2 Overview

As mentioned in the previous section, the system is a web application and because of that it is divided in two main components, client and server. These two components communicate via a *HTTP* requests.

Since it is a thick client application, the role of the server in this application is limited to providing some information and and entry point for the client side application. With this in mind the server responds only to three requests:

dataset/:id/meta serves a JavaScript Object Notation (JSON) document with the metainfo of the dataset specified by the :id parameter.

dataset/:id/view the entry point of the client side application. It serves the *javascript* and other assets required by the client application. The dataset viewed in the client side is the one specified by the :id.

dataset/index a list of links to view the available datasets.

The meta information for each dataset is stored directly in the server's *SQL* database and consists of: dataset name, geographical bounds, time interval and URL. The geographical bounds of the dataset are the minimum sized bounding box that includes all the events of the dataset. This bounding box is stored as the latitude and longitude coordinates of northwestern and southeastern points of the box. The time interval is comprised of the timestamps of the youngest and oldest events of the dataset. The URL of the dataset is the location in the web that the dataset is stored in. The dataset is stored as a JSON document structured in a simple array of tuples (latitude, longitude, timestamp).

The HTML/*Javascript* application on the client side is separated in two major components: the User Interface(UI) and the workers. The UI displays the map and controls to the user by using a combination of HTML, *CSS* and *Javascript*. Any operation that takes too long may block the UI, so it is done on a worker thread instead. The next section explains the whole client application with more detail.

5.2.3 Client Application

This section describes the main aspects behind the design of the client application; an overall explanation of the application is made in the following paragraphs, followed by a detailed explanation of each class used in the application with the aid of a class diagram shown in figure 5.5.

Initially, the application loads the metadata of the dataset that was selected by requesting a JSON file via an *AJAX* request. After the metadata is loaded, the client application makes another *AJAX* request for a JSON file containing the entire row set of the dataset. The row set is loaded into a Quadtree and the map is initialized using the metadata from the first request, setting the bounds and zoom level of the map so it can show all the dataset's points.

The map representation is composed of three layers: the geographic layer, the information layer and the control layer; these layers are placed over the map in the order they were listed, with the geographic layer on bottom and the control layer on top. The geographic layer shows the geographic information, like places and rivers, while the information layer is composed of several sub-layers that display the information of the dataset. There is an information sub-layer for displaying the AA-Map map and also an information sub-layer for each OST-Window that is on the map. The control layer shows the controls for the map, the controls for attenuation and accumulation technique and the

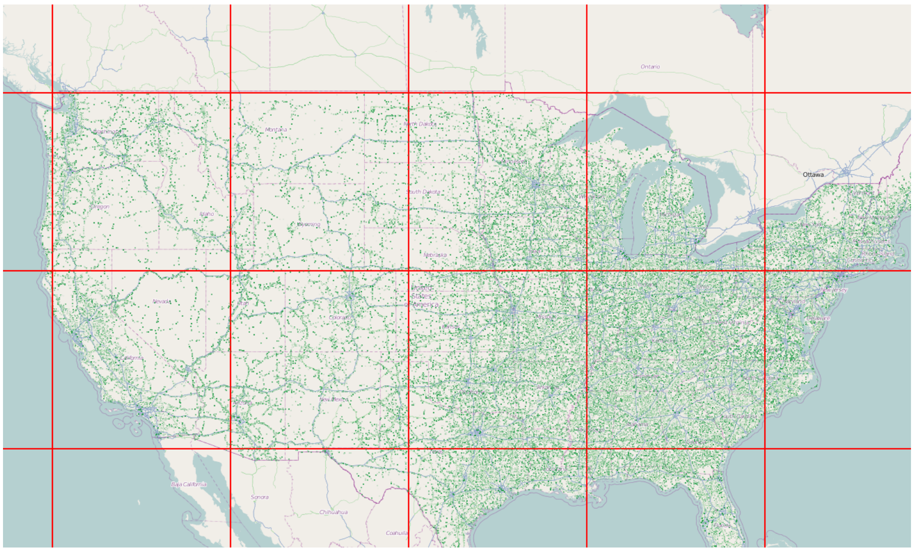


Figure 5.4: A depiction of the tile concept in a map by marking each tile with a red border.

UIs for each OST-Window. This composition results in the interface present in figure 5.7, which will be discussed in section 5.3.

Both the geographic and the information layers are divided in tiles as seen in figure 5.4, where each tile is responsible for showing the appropriate information for that part of the map. It is the *LeafletJS* library that creates the tiles, positions them in the right place and handles the placement of new tiles when the user zooms or pans the map. For the the geographic layer, each tile is an image that is loaded from a map provider, in this case the map provider is *OpenStreet Maps*[ope]. For the information layer, each tile is a canvas element where the information sub-layers are drawn according to the overlay operation of the OST-Windows, as defined in section 4.2.1. Each of these tiles has an image for each information sub-layer that is currently being viewed; it is these images that are drawn on that tile's canvas.

It is the tile division of the information layer that allows the use of web workers to create the visualization concurrently. Every time a tile is added to the map, via zooming or panning, the data that is inside the geographical bounds of that tile is sent to a worker. This data is quickly retrieved from the Quadtree but sending the data to the web worker is slower, making operations that add a lot of tiles to the map, like the initial setup or zooming, to take some time. When one of the information sub-layers (AA-Map or OST-Window) changes state, for each of the tiles that are being shown the state is sent to the appropriate worker. The worker then uses the state to create an image accordingly, using the algorithms explained in chapter 4. A sub-layer state includes: the pixel's color, the opacity function that is being used along with its parameters values and the time interval that is being analyzed.

A OST-Window sub-layer actually covers the whole map, with every tile containing an image describing the state of the window in that tile. For each tile a OST-Window image is drawn according to the tile's position relative to the geographic bounds of the OST-Window of the image. If a tile's boundaries are outside the OST-Window's boundaries then that window's image is not drawn. If the tile's boundaries intersects the OST-Window's boundaries, the intersection is calculated which results in a rectangle. The image is drawn on the canvas limiting it to the intersection.

The class diagram that is displayed in figure 5.5, shows the main classes of the client application, the *Map*, *TileLayer* and *ILayer* classes are from the *LeafletJS* library. The *TileLayer* class handles the geographic layer of the map.

The first class to be used in the application is the *DataSource* class; it loads the initial metadata, as well as the dataset which is stored in a Quadtree. It is the *DataSource*'s *getData* method that is used to retrieve the data for each of the tiles.

The *Workers* class handles the messages between the UI thread and the workers thread. The *paintImage* is used to paint an image of the specified tile using the given state. The image is painted using an *ImageData* object, and specifying the information for each pixel using the algorithms established for the AA-Map technique. This class can assign a tile to a worker, via the *assignWorker* method, by sending all the data inside that tile to a specific worker. This means that, for any method call that involves a tile and sending a message to a worker, the message will be sent to the worker assigned by the *assignWorker* method.

The *State* class stores all the information of the analytical context that is needed to paint a tile. This means the time interval of the analysis, the pixels color, the opacity function and its parameter values. The *OSTWindow* and *AttAccMap* classes handle the controls to the OST-Windows and the AA-Map map, and fire events according to the state change.

The *CanvasOverlay* class handles the information layer of the map. Each sub-layer and each tile is assigned an identifier, the tile identifiers are part of the *LeafletJS ILayer* and are the same used for the *GeoLayer* tiles. The *subLayerKeys* array has all the identifiers for all the sub-layer that are currently on the map. The tiles dictionary matches each tile to the corresponding *Canvas* element. The *SubLayerObject* stores for each tile the image corresponding to each sub-layer, this is done by matching each sub-layer identifier to a map that matches each tile identifier to the right image. Sub-layers are added and removed using the *addSubLayer* and *removeSubLayer* respectively. Every time a sub-layer is added, the *CanvasOverlay* registers handlers for each of the state changed events of the *SubLayer* class. There are events that affect just one *SubLayer* which are: the color changed event, time interval changed event. These events are handled by calling the *redrawSubLayer* method, which uses the *Workers* class to redraw the tiles for that sub-layer. Every time a worker returns an image the *repaintTile* method is called, which repaints the tile where the worker drew a new image for the sub-layer.

Events that involve changing the opacity function and its parameters need to redraw all the sub-layers, so the *redrawAll* method is called. The change of overlay operation simply causes the change of the *Canvas* overlay operation, so there is no need to use the workers; in this case the *repaintAll* method is called, which repaints all tiles using the

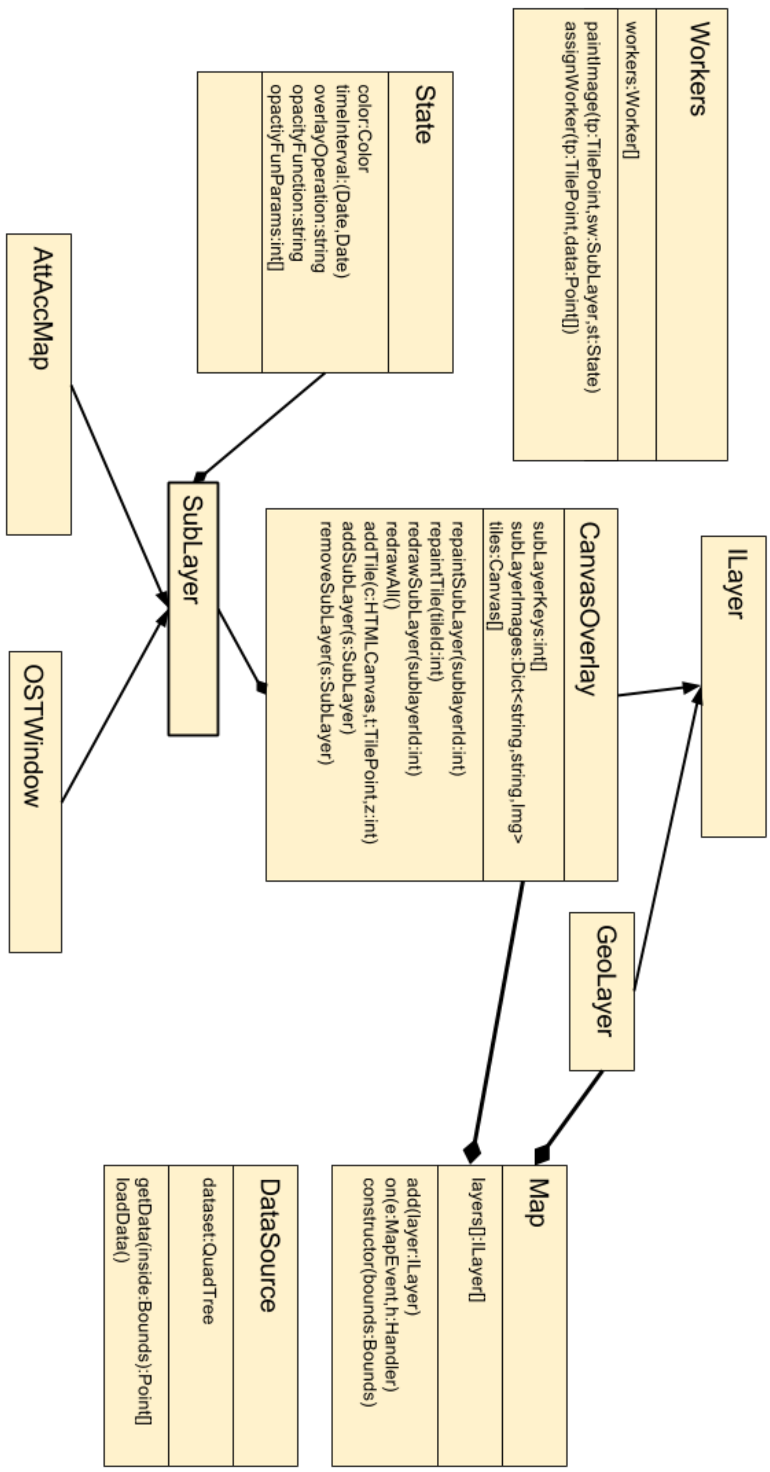


Figure 5.5: The class diagram of the client application.

new operation. The table in figure 5.6 condenses all the most relevant operations, and involved methods and identifies the most costly operations of the prototype.

Event Triggers	Called Methods	Description
Zooming in or out.	*AssignWorker*, *RedrawAll*	For each tile the new information is sent to a worker and then each information layer is drawn and each tile is painted.
Changing opacity function and any of its parameters.	*repaintAll*,*redrawAll*	It is called per information layer on the map, causes a redrawing of all information layers and a repainting of all the tiles.
Changing an information layer's time frame, or color. Adding a OST-Window. Changing the Overlay operation	*redraw*, *repaintAll*	The affected information layer is repainted, and then all information layers are redrawn.
Moving a OST-Window.	*redrawAll*	The OST-Window's boundaries are updated and all the tiles are redrawn.

Figure 5.6: A summarization of the various operations that involve web workers in the prototype, ordered in descending order by computation cost.

5.3 Interface

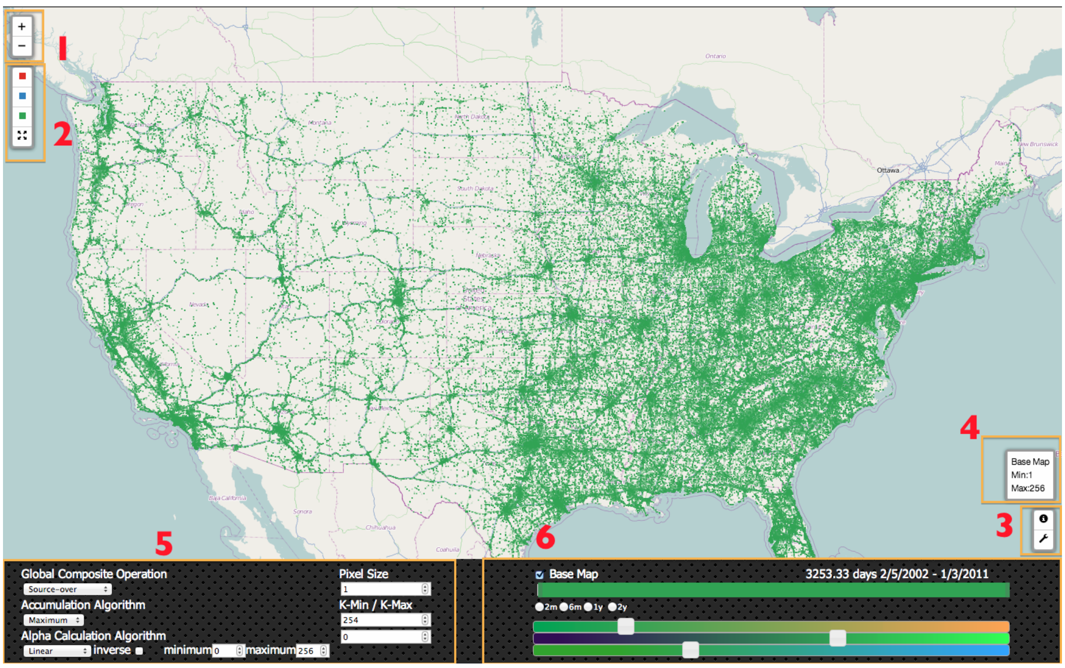


Figure 5.7: An overview of the prototype's interface

The interface designed provides all the operations described in chapter 4. The main focus of the interface is to experiment with the various technical aspects of the technique, and is by no means an interface designed for an end user. For this reason, most UI elements are used to control the algorithms that were described before like, the attenuation, accumulation and overlay functions. In the end, if the user is not familiarized with this concepts he will not be able to: (1) understand the analyses, because he does not know the algorithms (2) produce significant maps, because he does not know which parameters to use.

The user interface is depicted in figure 5.7, where each numbered rectangle identifies a set of different controls. The controls marked 4,5 and 6 are initially hidden so they do not clutter the analyses that is provided by the map. The various set of controls are described in the following list:

1. The typical zooming controls of any map application.
2. These controls allow two different operations. The first three icons, the red, blue and green squares, are used to add OST-Windows to the map, a square's color determines the base color of the pixels for the added OST-Window. The second operation, done by the last icon of the set, is to show the map in full screen mode.
3. The first icon, the *i* icon, is used to show and hide the 4th set of controls. The second icon, the wrench icon, is used to display or hide, both the 5th and 6th set of controls.
4. Shows the maximum and minimum attenuation that was attributed to the base map and each OST-Window.
5. Controls the operations that affect every sub-layer. It is used to change the overlay operation, the attenuation, accumulation functions. It is also used to change the parameters of the attenuation and mapping functions, as well as the pixel size.
6. These set of controls are for the AA-Map map, called the base map in the prototype, they control the time interval that is being shown and the pixels' color.

5.3.1 Overlapping Spatiotemporal Window Interface

In figure 5.8(a) is a zoomified version of a OST-Windows interface with each icon identified by a number, each icon as the following meaning:

1. The eye icon is used to show and hide the rest of the options, the idea is to not clutter intersecting OST-Windows with the other controls.
2. The move icon is used to drag the OST-Window over the map, changing the windows geographical bounds.
3. The clock icon is used to display the temporal controls as seen in figure 5.8(d), this controls allow for the manipulation of the time interval that is being analyzed. The time interval can be moved and resized, the radio buttons can be used to set a specific time interval of: 6 months, 1 year and 2 years.

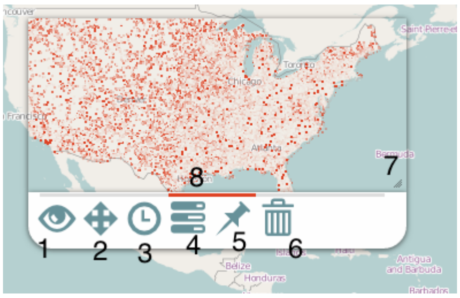
4. The bars icon is used to show the color picker controls that allow the user to change the base color of the window's pixels. These controls are displayed on the window's toolbar resulting in something similar to figure 5.8(c).

5. The pin icon is used to pin the OST-Window its geographical boundaries. When clicked feedback is given by painting the icon red as seen in figure 5.8(b).

6. The trash can icon when clicked removes the OST-Window from the map.

7. The usual resize icon, when dragged it resizes the OST-Windows geographical boundaries.

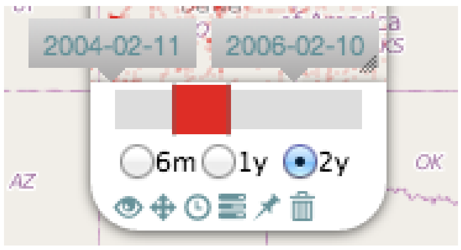
8. A visual representation of the time interval that is being analyzed in relation to the dataset's time interval. The whole time interval of the dataset is represented by a gray bar, while the current interval is represented by a bar painted with the OST-Window's pixel color, in this case the pixels of the window are red so the time bar is red.



(a) A scaled version of a OST-Windows toolbar

(b) A pinned OST-Window

(c) The color picker



(d) The time window selection

Figure 5.8: The UI of one OST-Window focusing on the window's toolbar, and the various effects the operations have on the toolbar's state.

5.4 Conclusion

The main purpose of the prototype was to create a proof of concept of both techniques, so the focus during development was to make the core operations of the technique fast and smooth. Moving and resizing OST-Windows, was crucial for the concept to work, and can be done with the prototype instantly, no matter the size of the data. Other important operations like changing the opacity function, the overlay functions and changing time intervals, take around one hundred milliseconds to be performed, all this considering the test datasets of over three hundred thousand entries.

The data loading phases of the prototype takes a considerable amount of time, and is one of the weakest points of the prototype. The initial load of data takes around fifteen seconds, while loads of data between zooms and pans take between two to fifteen seconds, depending on the amount of data. These are operations that can be improved in the prototype but are acceptable values. A quick improvement for the zoom and pan operations could be to pre-load the data for each tile, this would increase the initial load of the prototype but significantly decrease the time it takes to zoom in and zoom out in the application. Another improvement would be to have the tile logic implemented at server side, meaning the server would provide the data already split by tiles.

In terms of interface, the one presented is targeted at developers to find the better combinations of parameters, and functions. A UI for analysts should focus more on the analytical tasks that a technique provides and less on the technique itself. For instance, with the "Counting Events" opacity function from section 4.1.4.1, the UI could be reduced to just a time interval and 1 parameter that defines the number of events to count. These controls would shown to the user after he chose to count events.

Video showing some use cases of the prototype can be found at: `http://centria.di.fct.unl.pt/star/resources/files/AAMap.mp4` and `http://centria.di.fct.unl.pt/star/resources/files/OSTWindow.mp4`.

6

Conclusion and Further Work

In this final chapter an overall analyses is provided for the work that was done. The main conclusions are presented, followed by a presentation of further work that may be done over the various contributions of this document.

6.1 Conclusion

In this dissertation, a theoretical framework was designed that conceptualizes the problem of analyzing spatiotemporal data, which resulted in the creation of the catalog for spatiotemporal patterns. Beginning with a research in analytical techniques for spatial data and temporal data, the objective was to identify what are the basic concepts and principles in spatial and temporal data that can be used to facilitate the detection of spatial and temporal patterns. The different perceptions of time and the various principles that are the basis of thematic maps among other concepts, were all taken into consideration when designing the catalog.

Following the research of those techniques was a survey of spatiotemporal techniques. The techniques analyzed were chosen for the different concepts that each technique used to achieve its goals, and for providing distinct analytical possibilities. This research was presented with an explanation of the spatiotemporal techniques focusing on each technique's design concepts and general ideas. Furthermore, for every technique, a detailed evaluation of its analytical capabilities was made, revolving on the situations they were used for in the literature.

The main contribution of this thesis, the catalog, is based on two frameworks that were created to better describe the spatiotemporal problem. The first was the data scenarios framework, a theoretical model that can be used to describe spatiotemporal datasets.

With this framework 23 different scenarios were identified and characterized, all with a real world example that validates that scenario. The model in itself can be used to describe spatial datasets as well as temporal datasets. The second framework describes a model to characterize analytical tasks for spatiotemporal data, it was used to describe 9 different analytical tasks found in the literature. This model can also be used to describe tasks for spatial data, and temporal data.

The categorization of the surveyed techniques completes the catalog and validates the suggested frameworks. In the end, a document is presented where, if proposed a spatiotemporal problem by conforming its dataset to the corresponding data scenario and choosing an appropriate analytical tasks, a technique can be identified for that problem.

However, the catalog does not provide a definitive answer to all analytical spatiotemporal problems. This is in part because of the low number of analytical tasks characterized in the catalog but also because of the lack of techniques that exist for these problems.

To answer the need for more spatiotemporal techniques, and as a result of the research on the spatiotemporal problem, two techniques for analyzing spatiotemporal data were idealized: the attenuation & accumulation map (AA-Map) and the overlapping spatiotemporal window (OST-Window) technique. A formal definition of the concepts and ideas of both techniques was presented, delineating each technique's uses and limitations.

A prototype was developed as a proof of concept for both techniques. The developed system was for an academic environment, with a very technical interface that allowed to investigate the various possibilities for both techniques.

6.2 Further Work

Upon some contributions of this document: the OST-Window technique and the AA-Map technique, there is further work that can be done. The two spatiotemporal techniques proposed in this document were proof of concepts, idealization of a new way of analyzing a (P, t) scenario. This resulted in a prototype that was suited to prove the techniques importance, but is unsuited to prove the techniques usability. As such, an interface should be designed with an end user in mind that is not an academic, familiarized with the techniques algorithms and functions, but an analyst that wants to extract information from data without caring how it is done.

Furthermore, said interface will have to be validated recurring to usability tests. This tests will eventually prove, or not, the usability of that interface and to a greater extent validate the importance of both techniques.

Bibliography

[AAB+11] G. Andrienko, N. Andrienko, P. Bak, D. Keim, S. Kisilevich, and S. Wrobel. A conceptual framework and taxonomy of techniques for analyzing movement. *Journal of Visual Languages & Computing*, 22(3):213–232, 2011.

[AAG03] Natalia V. Andrienko, Gennady L. Andrienko, and Peter Gatalsky. Exploratory spatio-temporal visualization: an analytical review. *J. Vis. Lang. Comput*, 14(6):503–541, 2003.

[AAJ+06] Gennady Andrienko, Natalia Andrienko, Piotr Jankowski, Daniel Keim, Menno-jan Kraak, Alan Maceachren, and Stefan Wrobel. Geovisual Analytics for Spatial Decision Support : Setting the Research Agenda. pages 1–19, 2006.

[AAM+10] Gennady L. Andrienko, Natalia V. Andrienko, Martin Mladenov, Michael Mock, and Christian Pölitz. Discovering bits of place histories from people's activity traces. In *IEEE VAST*, pages 59–66. IEEE, 2010.

[AMM+08] Wolfgang Aigner, Silvia Miksch, Wolfgang Müller, Heidrun Schumann, and Christian Tominski. Visual methods for analyzing time-oriented data. *IEEE transactions on visualization and computer graphics*, 14(1):47–60, 2008.

[Ber83] Jacques Bertin. *Semiology of graphics*. University of Wisconsin Press, 1983.

[BMJK09] Peter Bak, Florian Mansmann, Halldor Janetzko, and Daniel a Keim. Spatiotemporal analysis of sensor logs using growth ring maps. *IEEE transactions on visualization and computer graphics*, 15(6):913–20, 2009.

[Car03] M. S. T. Carpendale. Considering Visual Variables as a Basis for Information Visualisation. Technical report, University of Calgary, Calgary, AB, 2003.

[CFL+11] Davide De Chiara, Vincenzo Del Fatto, Robert Laurini, Monica Sebillo, and Giuliana Vitiello. A chorem-based approach for visually analyzing spatial data. *J. Vis. Lang. Comput*, 22(3):173–193, 2011.

[Few08] Stephen Few. Time on the Horizon. pages 1–7, 2008.

[GAA04] Peter Gatalsky, Natalia V. Andrienko, and Gennady L. Andrienko. Interactive analysis of event data using space-time cube. pages 145–152, 2004.

[Gov] United States Government. Data.gov. http://www.data.gov.

[htm] Canvas 2d context w3 specification. http://www.w3.org/TR/2dcontext2/.

[Hä70] Torsten Hägerstrand. What about people in regional science? *Papers of the Regional Science Association*, 24:6–21, 1970.

[ICN] ICNF. Dados sobre incêndios florestais. http://www.icnf.pt/portal/florestas/dfci/inc/estatisticas.

[JME10] Waqas Javed, Bryan McDonnel, and Niklas Elmqvist. Graphical perception of multiple time series. *IEEE transactions on visualization and computer graphics*, 16(6):927–34, 2010.

[KO03] M.J. Kraak and F. Ormeling. *Cartography: Visualization of Geospatial Data*. Prentice Hall, 2003.

[KPS05] Daniel A. Keim, Christian Panse, and Mike Sips. Information Visualization: Scope, Techniques and Opportunities for Geovisualization. In Jason Dykes, Alan M. Maceachren, and Menno J. Kraak, editors, *Exploring Geovisualization*, pages 23–52. Elsevier, Amsterdam, 2005.

[Kra] MJ Kraak. What has ITC done with Minard's map? http://www.itc.nl/personal/kraak/1812/minard-itc.htm. [Online; accessed 27-January-2012].

[Kra03] MJ Kraak. The space-time cube revisited from a geovisualization perspective. *Proc. 21st International Cartographic Conference*, (August):10–16, 2003.

[lea] Leafletjs opensource javascript map library. http://leafletjs.com/.

[ope] Open street maps "the free wiki world map". http://www.openstreetmap.org/.

[RFF+08] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *IEEE transactions on visualization and computer graphics*, 14(6):1325–32, 2008.

[Ros] Hans Rosling. Gapminder. http://www.gapminder.org.

[TH10] Sidharth Thakur and Andrew J. Hanson. A 3D Visualization of Multiple Time Series on Maps. *2010 14th International Conference Information Visualisation*, pages 336–343, July 2010.

[TSAA12] C. Tominski, H. Schumann, G. Andrienko, and N. Andrienko. Stacking-Based Visualization of Trajectory Attribute Data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2565–2574, 2012.

[TSWS05] C. Tominski, P. Schulze-Wollgast, and H. Schumann. 3D Information Visualization for Time Dependent Data on Maps. *Ninth International Conference on Information Visualisation (IV'05)*, pages 175–181, 2005.

[Tuf06] Edward R. Tufte. *Beautiful Evidence*. Graphis Pr, 2006.

[WAM01] M. Weber, M. Alexa, and W. Müller. Visualizing time-series on spirals. In *2001 IEEE Symposium on Information Visualization (INFOVIS '01)*, pages 7–14, Washington - Brussels - Tokyo, October 2001. IEEE.

[web] Web workers specification. `http://www.w3.org/TR/workers/`.