FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Spatiotemporal Phenomena Summarization through Static Visual Narratives

Daniel Filipe Santos Marques

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Alexandre Valle de Carvalho

Second Supervisor: Rui Rodrigues

July 20, 2020

Spatiotemporal Phenomena Summarization through Static Visual Narratives

Daniel Filipe Santos Marques

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. Augusto de Sousa

External Examiner: Prof. Paulo Dias

Supervisor: Prof. Alexandre Valle de Carvalho

July 20, 2020

Abstract

The process of automatically collecting spatiotemporal data has been made easier over the years, resulting in enormous amounts of data to be analysed. Data visualization aids the understanding of a phenomenon's evolution in space across a time period. Nonetheless, the most common visualization methods may lack efficiency or effectiveness in transmitting the story of a spatiotemporal phenomenon. Dynamic visualizations, such as animations, may not be time-efficient in scenarios that span for long periods as the time effort to consume the necessary visual content increases linearly with the data. Furthermore, dynamic visualizations are inherently prone to object changes being imperceptible in cases of small or fast-occurring transformations. On the other hand, static alternatives, such as map-based frameworks, result in a summary of the data, fixing the time effort and imperceptibility issues. However, current static methods are incapable of conveying dynamic phenomena effectively. Moreover, both static and dynamic visualization approaches currently lack the ability to filter the visual contents by what spatiotemporal events are of interest.

The current work proposes a novel approach to visually represent spatiotemporal phenomena based on automated generation of interactive storyboards that summarize the evolution of a spatiotemporal phenomenon through a set of frames that represent the most relevant change across all events of interest. Although a static approach, this storyboard can be dynamically modified through user interaction, enabling the viewer to filter through events of interest, adjust the level-of-detail, and navigate through the timeline. Towards corroborating the hypothesis that this approach would effectively and efficiently transmit the evolution of spatiotemporal phenomena, a visualization framework was conceptualized, identifying visual metaphors that map spatiotemporal transformations into visual content, defining the parameterization approaches for spatiotemporal features, and presenting a framework pipeline tailored to the envisioned solution. Afterwards, a functional prototype implementing the conceptual solution was developed, asserting the feasibility of the visualization framework and serving as a support tool for validation.

To assess the effectiveness and efficiency of the conceptual visualization framework, a questionnaire was produced and conducted on college students. The results concluded that static visual narratives are effective at transmitting spatiotemporal information with overall satisfactory precision, in spite of difficulties in the interpretation of the level-of-detail layout. The comparison between static and dynamic approaches towards spatiotemporal phenomena visualization concluded that static visual narratives are more efficiency in scenarios with potentially imperceptible transformations or for data that represents long periods of time. Nevertheless, users preferred the simplicity and intuitiveness of dynamic approaches for the remaining scenarios. Considering the assessment, the authors conclude that summarizing spatiotemporal phenomena through static visual narratives is a suitable alternative for understanding their evolution.

Keywords: spatiotemporal visualization, static visual narrative, spatiotemporal phenomena, events of interest

Resumo

O processo de reunir dados espaciotemporais tem sido facilitado ao longo dos anos, resultando em enormes quantidades de dados para análise. A visualização de dados auxilia a compreensão da evolução dos fenómenos no espaço ao longo de um período de tempo. No entanto, os métodos mais comuns de visualização podem ser ineficazes ou ineficientes na transmissão da história de fenómenos espaciotemporais. As visualizações dinâmicas, como animações, podem não ser eficientes em termos de tempo em cenários que se prolongam por longos períodos, visto que o tempo gasto para consumir o conteúdo visual aumenta linearmente com a quantidade de dados. Adicionalmente, visualizações dinâmicas são inerentemente propensas à impercetibilidade de pequenas ou rápidas transformações do objeto. Por outro lado, alternativas estáticas, como métodos baseados em mapas, resultam num resumo dos dados, resolvendo os problemas de tempo de consumo de conteúdo e impercetibilidade de alterações. No entanto, os métodos estáticos atuais são incapazes de transmitir eficazmente fenómenos dinâmicos. Em adição, ambas as abordagens não possuem, atualmente, ferramentas para filtrar o conteúdo visual por eventos que são de interesse.

O presente trabalho propõe uma nova abordagem para representar visualmente fenómenos espaciotemporais com base na geração automatizada de *storyboards* interativos que resumem a evolução de um fenómeno espaciotemporal através de um conjunto de vinhetas que representam a mudança mais relevante em relação aos eventos de interesse. Embora seja uma abordagem estática, o *storyboard* pode ser modificado dinamicamente através de interação do utilizador, permitindo a filtragem de informação por eventos de interesse, o ajuste do nível de detalhe, e a navegação pela *timeline*. Para corroborar a hipótese de que esta abordagem transmite eficaz e eficientemente a evolução de fenómenos espaciotemporais, uma estrutura de visualização foi conceptualizada, identificando metáforas visuais que mapeiam transformações espaciotemporais para conteúdo visual, definindo as abordagens de parametrização para transformações espaciotemporais e apresentando uma *pipeline* adaptada à solução proposta. Posteriormente, foi desenvolvido um protótipo funcional que implementa a solução conceptual, afirmando a viabilidade desta abordagem de visualização e servindo como ferramenta de suporte à validação.

De forma a avaliar a eficácia e eficiência da solução conceptual de visualização, um questionário foi realizado a estudantes universitários. Os resultados concluíram que as narrativas visuais estáticas são eficazes na transmissão de dados espaciotemporais com precisão satisfatória, apesar das dificuldades na interpretação dos vários níveis de detalhe. A comparação entre abordagens estáticas e dinâmicas para a visualização de fenómenos espaciotemporais concluiu que as narrativas visuais estáticas são mais eficientes em cenários com transformações potencialmente impercetíveis ou para dados que representam longos períodos de tempo. No entanto, os utilizadores preferiram a simplicidade de abordagens dinâmicas para os restantes cenários. Considerando a avaliação, concluiu-se que resumir fenómenos espaciotemporais através de narrativas visuais estáticas é uma alternativa adequada para a compreensão da sua evolução.

Palavras-chave: visualização espaciotemporal, narrativa visual estática, fenómenos espaciotemporais, eventos de interesse

Acknowledgements

I want to express my gratitude towards Professor Alexandre Carvalho and Professor Rui Rodrigues for providing me with constant guidance, learning opportunities, and for the knowledge supplied throughout this work. I am deeply indebted to all the professors that have crossed my life and allowed me to learn and pursue my dreams.

I would like to thank my family, from the bottom of my heart, for their unconditional love, for educating me, and for their incentive and support towards my decisions. I also thank my friends for encouraging me throughout the hardest of times, for creating unforgettable memories, and for their constant selfless acts of friendship.

Finally, I thank everyone who has been a part of this work, from the students who answered the evaluation questionnaire to the kind people that have unknowingly contributed by answering my programming questions.

Daniel Marques

"Sic Parvis Magna"

Sir Francis Drake

Contents

List of Figures

List of Tables

Listings

Abbreviations

API	Application Programming Interface
CSS	Cascading Style Sheets
GIS	Geographic Information System
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
JS	JavaScript
LOD	Level of Detail
MVC	Model-View-Controller
SDK	Software Development Kit
UML	Unified Modeling Language

Chapter 1

Introduction

This chapter introduces the current work, starting by giving some context, in Section 1.1, regarding the spatiotemporal domain. Section 1.2 indicates the importance of this area and the reason to contribute to it. Afterwards, the problems with current spatiotemporal visualization frameworks are identified in Section 1.3. Subsequently, in Section 1.4, the objectives aimed to be accomplished with this work are listed. Finally, in Section 1.5, an outline of the document structure is provided, briefly explaining the contents of each chapter.

1.1 Context

Nowadays we have the technology to gather, store, retrieve and represent vast quantities of data. The spatiotemporal domain is not an exception to this progress. In fact, the more specific domain of Geographic Information Systems (GIS) is evidence of this as more satellites orbit the Earth and transmit numerous amounts of data back to be stored, analysed and visualized. With recent advances in the field of computer vision, it has become possible for this data to be in raster format, relying on automated tools to detect phenomena. These phenomena are of spatiotemporal nature, meaning they concern objects which exist in space, and their multidimensional attributes change over time [3].

Visualization tools are common approaches towards intuitively analysing data. In terms of visualization tendencies, web applications have been slowly substituting desktop applications over the years [48]. Contributing to this shift is the required installation and manual updates of desktop applications. Instead, web applications offer ubiquitous execution through web browsers, present in virtually any consumer operating system. Its wide availability through the Internet is also one of its biggest disadvantages, requiring internet connectivity which, at this time, is becoming less of a problem.

Regarding the enabling technologies for visualization, the computer graphics field has been adopting the previously mentioned shift towards web environments [4]. Over the last decade, as the OpenGL specification matured, other variants were created to cope with the technological changes. OpenGL ES was developed for embedded systems (e.g., smartphones) and WebGL based on OpenGL ES to fulfil the hardware-accelerated graphical needs of the web.

1.2 Motivation

With the evolution of science and technology, people have access to enormous amounts of data regarding spatiotemporal phenomena. However, having more data does not necessarily ease the perception, communication or analysis of a phenomenon's evolution. For centuries, people have resorted to visualizations as a means to understand the data in a more efficient manner.

The motivation for contributing to the area of spatiotemporal visualization lies in creating or improving visualization frameworks that help users understand relevant phenomena. These tools are responsible for aiding scientists to make discoveries, to find events, understand their causes and consequences, and discover patterns (e.g., an asteroid's path). Understanding certain past phenomena might also assist in predicting and preparing future situations (e.g., the advance of forest fires). Additionally, it is inherently a duty to inform the general public of relevant spatiotemporal occurrences that may impact the well-being of all, possibly to promote behaviour changes (e.g., the effects of pollution and climate change).

1.3 Problem definition

Information visualization is a field dedicated to easing the perception, communication, and analysis of information through visual content [10, 29]. In this field, it is common not to have a specific visualization method that *dominates* the others. Choosing a visualization framework over another is a process which depends on aspects such as the type of data, the expected tasks to perform, the data domain, and user preference. The specific case of spatiotemporal visualization is not an exception: there are various visualization frameworks available, but various aspects prevent them from being the irrefutable choice for every scenario.

The problem regards how to visualize spatiotemporal phenomena is an effective and efficient manner, considering the limitations of the current visualization methods. In the case of dynamic visualizations (e.g., animations or videos), the core issue is related to efficiency and accuracy. In terms of efficiency, in most cases, the user would have to consume the complete visual content to understand the story of the phenomenon. This logic can be worsened in scenarios where an object remains immutable for long periods of time, assuming the interest is detecting object change. Additionally, in regards to accuracy, users might be tempted to skip or adjust the play speed of the visualization with no prior knowledge of the data, possibly resulting in loss of information given that changes to an object may occur very fast and very limited in time.

In regards to static visualizations, the problem derives from the inherited difficulty of depicting the dynamic aspect of spatiotemporal phenomena in static representations. The effectiveness of this type of visualizations is therefore strongly dependent on the visual metaphors used to convey the different types of transformations, which are often not formally documented as such.

Common to both approaches is the lack of filtering through events of interest. In other words, the ability to interactively choose what type of behaviours are relevant for the given domain or task and the minimum amount in these behaviour values that should be considered a relevant event. The inexistence of such a feature hinders usability and efficiency since, as stated, users must consume visual content that might not be relevant.

1.4 Objectives

Visually representing the spatiotemporal phenomena through *Static Visual Narratives* can be a more efficient alternative compared to the previously mentioned approaches. Therefore, the objective of this work is to assess this statement through studying, modelling, prototyping and testing a visualization framework that generates static visual narratives. This novel approach consists of an automated and interactive storyboard that visually summarizes the story of a spatiotemporal phenomenon through a set of static frames that represent the most relevant changes that occurred to the object, according to the user's interests. The ability to interact lets the user tweak the visual summary's level-of-detail, select which events are of interest, and progress through the timeline. User testing is also crucial to evaluate the effectiveness and efficiency of the visualization.

A framework for extracting and visualizing spatiotemporal information can be decomposed in the following stages:

1. Study the representation of the information in data structures suited for spatiotemporal data and quantify the changes that occur to the phenomenon;

2. Study, define and develop mappings that convert data into meaningful static visual narratives, visually summarizing the evolution of a spatiotemporal phenomenon, perform tests and evaluate the results.

This work focuses on the second stage, intending to contribute with a functional and tested prototype. The definition of this framework uses concepts from visual storytelling and information visualization which are reviewed from the literature as these fields' purpose is to improve the understanding of the presented information. Its implementation requires technical knowledge from computer graphics, user interfaces and software engineering in order for the prototype to be functional. As with any visualization module of a software system, it is dependent on having data. The expected inputs, resulting from the previously presented first stage, can be mocked based on real data in order to simulate the stage and an input format is proposed for future integration.

1.5 Document structure

This document is structured into seven chapters. After this first introductory chapter, Chapter 2 summarizes the state of the art in the visualization of spatiotemporal phenomena through a systematic literature review. Chapter 3 defines the problem in regards to the current approaches of visualizing spatiotemporal phenomena. Chapter 4 describes the rationale of the proposed solution to the aforementioned problem, enlightened by the knowledge gathered from the literature review. Chapter 5 details the development of a functional prototype that implements the conceptual solution presented in the previous chapter. Chapter 6 describes the evaluation process adopted to test the effectiveness and efficiency of the developed prototype and discusses its results. Finally, Chapter 7 presents the conclusions from this work and proposes future work on this topic.

Chapter 2

State of the Art

This chapter reviews the related work of each of the different areas that the current work regards. Section 2.1 presents key concepts of the spatiotemporal domain, possible transformations that may occur to a phenomenon or a set of phenomena, and the representation of spatiotemporal changes. Section 2.2 defines, presents and classifies visual narratives according to accepted taxonomies and lists the most important elements in these narratives. Section 2.3 describes the taxonomies of visualization converging towards the spatiotemporal domain, identifies visual variables used in creating visual metaphors, outlines methods for visualizations, and reviews the process of evaluating the effectiveness of a visualization. Lastly, Section 2.4 identifies the available enabling technologies for rendering the visual elements introduced in Section 2.3.

2.1 Spatiotemporal phenomena

Spatiotemporal phenomena is a term that describes objects that exist or occur in space and the transformations that occur to their multidimensional attributes over time [3]. An example of a spatiotemporal phenomenon occurs when an iceberg (the object), which has a position in space (an attribute) that changes over time and, for instance, due to warmer temperatures, it may melt, changing its shape, or break resulting in two or more new objects.

2.1.1 Phenomena transformations

Spatiotemporal phenomena, by definition, encompass the changes undergone by their objects over time. Therefore, it is important to define their properties and possible transformations. Blok [6] defines a taxonomy for spatiotemporal transformations which divides a phenomenon into two domains: spatial and temporal.

For the spatial domain, Blok identifies the concepts of *existential change* (appearance or disappearance of an object), *mutation* (transformations of the thematic attributes of a phenomenon), and *movement* (change in the phenomenon's position in space or its geometry). Mutations can be subdivided based on their scale of measurement. Stevens [47] introduced four scales of measurement:

- Nominal — used for labelling variables into categories, without quantitative values (e.g., classifying a phenomenon as related to an iceberg or to a human cell);
- Ordinal — the variables are ordered in a sequence and its values can be non-numeric (e.g., a sequence describing the size of an object between 'small', 'normal' and 'big');
- Interval — a numerical scale where the difference between values is known (e.g., representation of dates);
- Ratio — a scale to provide meaningful ratios (i.e., the variables can be meaningfully multiplied and divided) and a true zero point (e.g., there can be an absence of distance by using the value of zero).

The classification of mutations according to the scales of measurement result in:

- Mutations at the nominal scale of measurement (a transformation of its nature, e.g., rain that turns into snow);
- Mutations at the ordinal, interval or ratio scale of measurement (a quantifiable change on its thematic attributes, e.g, the thickness of a cloud).

Movements are also subdivided into two types, resulting in movement across a trajectory (where the entire object changes position in space, forming a path) or boundary shifts (where only a fraction of the phenomenon moves, while the rest remains stationary).

For the spatial domain, Worboys [57] introduces more concepts related to the existential changes mentioned previously: *change primitives*, which include *fission* and *fusion* as phenomena can physically split or merge.

For the temporal domain, Blok proposes the concepts of *moment in time* (the instant of time that marks the beginning of a change of a phenomenon), *pace* (the rate of change over time), *duration* (the length in time between the beginning of a transformation and its end), *sequence* (the order of a set of changes), and *frequency* (the number of occurrences of a change).

Shekhar et al. [46] define a similar classification, identifying three distinct data attributes: non-spatiotemporal attributes, spatial attributes, and temporal attributes. The major difference is the addition of non-spatiotemporal attributes which can be, for example, the name of a city.

Dodge et al. [17] define another conceptual framework for movement, shown in Table 2.1, which decomposes movement into parameters and dimensions. The authors consider primitive parameters and derivative parameters (which are derived from the primitives and can be primary or secondary depending on the derivative level). These parameters are organized into the spatial, temporal and spatiotemporal dimensions. The latter constitutes a major addition to the other classification methods, as the spatial and temporal dimensions are joined, allowing for features whose existence depends on both dimensions, like velocity and acceleration. The authors also document the concept of *relative movement*, as opposed to absolute, as a means to compare multiple objects.

Parameters / Dimension	Primitive	Primary derivatives	Secondary derivatives
Spatial	Position (x,y)	Distance $f(posn)$ Direction $f(posn)$ Spatial extent $f(posn)$	Spatial distribution $f(distance)$ Change of direction $f(direction)$ Sinuosity $f(distance)$
Temporal	Instance (t) Interval (t)	Duration $f(t)$ Travel time $f(t)$	Temporal distribution Change of duration $f(duration)$
Spatiotemporal	—	Speed $f(x,y,t)$ Velocity $f(x,y,t)$	Acceleration $f(speed)$ Approaching rate

Table 2.1: Movement parameters [17]

2.1.1.1 Planar transformations

The majority of phenomena properties presented in the previous section are agnostic to the number of dimensions used for space. Hence, they can be instantiated onto 2D space, matching the context of the current work. As a result, the changes that can be perceived are reduced (for example, in a top-down visualization a change in altitude of an object can be difficult to understand). On the other hand, being in 2D enables thinking of the changes in terms of planar transformations, which matches the scope of this work. Therefore, identifying these planar transformations is essential for the definition of visual metaphors for each transformation that occurs to the phenomenon and for its internal data representation.

Wilkinson [56] defines a hierarchy of planar transformations through a set of classes where each has an invariance (a feature of the object that remains unchanged after the transformation) and a set of possible transformation methods.

1. *Isometric* transformations are characterized by preserving the distance between points, thus earning the name of "rigid transformations". These include translation, rotation, and reflection.

2. *Similarity* transformations are characterized by changing the size of the object while maintaining its original shape. The transformation identified is called dilation which includes both forms of object resizing (shrinking and enlargement).

3. *Affine* transformations cause one dimension to stretch or shear independently of the other. However, affine transformations are still able to maintain the parallelism of lines.

4. *Projective* transformations indicate the projection of the object onto another plane with the invariance of maintaining the straightness of the lines.

5. *Conformal* transformations are operations which preserve local angles while possibly distorting the global shape of the object.

In terms of 2D geometrical transformations, this work focuses on the first three types: isometric, similarity and affine. Towards building a conceptual model of the internal representation of phenomena transformations, it is necessary to dive into each considered transformation and its mathematical representation.

Translation is an *isometric* transformation which moves an object along a straight-line path. To translate the two-dimensional point $p = (x, y)$, a translation vector $t = (t_x, t_y)$ — a vector which contains the distances to move the object in each dimension — is added to it, moving the point p to a new position p' [25].

$$p' = (x', y') = (x + t_x,\ y + t_y)$$

Rotation is an *isometric* transformation which moves an object along a circular path. To rotate the two-dimensional point $p = (x, y)$, the rotation angle θ and the pivot point $p_r = (x_r, y_r)$ (the centre of the circular path) needs to be specified [25].

$$p' = (x', y') = (x_r + (x - x_r)cos\ \theta - (y - y_r)sin\ \theta,\ y_r + (x - x_r)sin\theta + (y - y_r)cos\ \theta)$$

This equation can be simplified if the centre of rotation is the origin of the coordinate system:

$$p' = (x', y') = (x\ cos\ \theta - y\ sin\ \theta,\ x\ sin\ \theta + y\ cos\ \theta)$$

Scale is a transformation which can group both *dilation* and *stretch*, meaning it can either scale the object uniformly (*isotropic scaling* — *similarity* transformation) or independently between dimensions (*anisotropic scaling* — *affine* transformation). The reasoning behind the grouping of these transformations is due to them having similar mathematical representations. To scale the two-dimensional point $p = (x, y)$, it is multiplied with a scale vector $s = (s_x, s_y)$ — a vector which contains the scale factor for each dimension —, resulting in a new point p' [25]. Regarding the classification, if s_x equals to s_y then the transformation is isotropic, and anisotropic otherwise.

$$p' = (x', y') = (s_x * x,\ s_y * y)$$

2.1.2 Events of interest

Worboys presents different ways of discretely representing spatiotemporal phenomena with a set of progressively improving stages [57]. In *Stage Zero* (static) the system represents a single state of the domain object. It may allow the update of the state, but, at a given moment in time, no more than one state is visible, thus making comparison between states impossible.

In *Stage One* (temporal snapshots) the domain is viewed as a sequence of temporal snapshots. Each snapshot represents a state of domain objects in a moment in time. A collection of temporal snapshots can be viewed at the same time, making comparisons possible. This approach is directly linked to the concept of temporal granularity which determines the temporal (timestamp) difference between two subsequent snapshots. This definition raises the question of "What is an appropriate temporal granularity?". The answer depends on the task: if it is based on the change in object states or purely on the temporal aspect.

If the former is considered, a "perfect" temporal granularity may not exist as changes in the domain may take either very short or long intervals of time to occur. In *Stage Two* (object change), this problem is addressed. In this stage, the focus is on representing the changes that happen to objects, attributes, and relationships. The primitive shifts from a state, in previous stages, to a change (e.g., the disappearance of an object in a given time interval). In *Stage Three* (events and actions), the authors take the object changes a step further by taking into account relationships between events (e.g., overlapping, cause and effect, etc).

Therefore, transformations to spatiotemporal phenomena can be defined as a set of events, representing object change. An important concept for this work is the *events of interest*. Introduced by Tominski et al. [50], events of interest emerge as a consequence of large datasets with numerous types of phenomena transformations. It is presented as a concept that allows the user to choose, depending on the task, which events are of interest. An event is marked when:

1. an attribute value exceeds a given threshold (e.g., the temperature of a computer component surpassing its working limit);
2. an increase in an attribute value from one instance in time to another exceeding a given threshold (e.g., a human cell increasing its size far more than expected).

The authors also explore the dichotomy between locally and globally detected events. In regards to the temporal aspect, the difference would be detecting events for a single time step or in the entirety of the considered time domain. Similarly, regarding the spatial aspect, an event can be detected by considering a single area in space or the complete considered space domain. It is also pointed out that considering the totality of a certain domain, either spatial or temporal, enables the identification of more complex patterns at the cost of requiring more effort from the user.

2.2 Visual narratives

Visual Narratives are visuals that narrate a story. The term was only recently defined in the literature, by Pimenta and Poovaiah [40]. The authors establish three types of visual narratives:

- *Static Visual Narrative*: A set of visuals that remain fixed and unchanging on the surface of the medium. It fully engages the viewer as it heavily relies on each individual to perceive the dynamic aspects of the story, giving the user the freedom to decide the speed to consume the visual content. With the images frozen in time, the perception of movement results from the active participation of the spectator, whose mind can infer what is missing and complete the actions. One of the most recognized examples of static visual narratives is comic art.

- *Dynamic Visual Narrative*: The visuals of the narrative constantly change over time. The story is constructed before it is presented to the spectator. This effectively limits the viewer's control over the narrative. The predetermined nature of dynamic visual narratives means the viewer cannot control the pace of the narrative or which events to see: one would have to wait for the story to progress. Some examples of dynamic visual narratives include animations and movies.

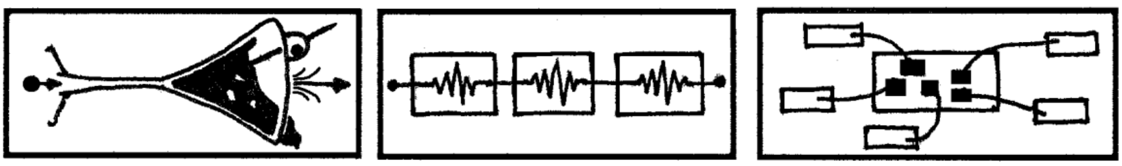
- *Interactive Visual Narrative*: The visuals of a narrative can be impacted by the viewer's actions. This means that an interactive visual narrative incorporates the interaction from the viewer towards unfolding the story. These types of visual narratives can be, as a base, either static (e.g., interactive storybooks) or dynamic (e.g., video games).

Another classification system can be connected to the previously described. Visual narratives are designed to balance the narrative intended by the author(s) with story discovery by the reader [45]. Consequently, the authors distinguish between author-driven and reader-driven approaches:

- *Author-driven*: Characterised by their strict linear narrative path through the visualization, author-driven approaches tell the story the author wants to be told, heavily relying on messaging and disregarding viewer interactivity. An example of author-driven stories is a movie, which corresponds to dynamic visual narratives from the classification system above. However, there are cases of author-driven approaches in static visual narratives as well.

- *Reader-driven*: Characterised by not enforcing the order of the visual contents shown to the reader which, by means of interactivity, can discover various stories. When adapting to data story-telling, the viewer is free to analyse data, discover patterns, find events, and suggest hypotheses.

Despite these categories, the visualizations should be considered in a spectrum along author and reader-driven approaches, allowing for hybrid models to emerge. The hybrid models identified by Segel and Heer [45] — martini glass structure, interactive slideshow, and drill-down story — shown in Figure 2.1, move along the spectrum from prioritizing author-driven approaches to reader-driven approaches. The interactive slideshow enforces event order (since the events do have an occurring order) but allows the user to freely explore through the stages of the story through interaction (instead of a predetermined constant display of images). The others, however different, can contribute to some lacking aspects of the interactive slideshow. The martini glass structure imposes an initial author-constructed story which introduces the visualization, unlocking the reader to explore the data when finished. The drill-down story allows the reader to choose which part of the story to consume, allowing for more detailed views to appear through interaction with the stories of interest.



(a) Martini glass structure (b) Interactive slideshow (c) Drill-down story

Figure 2.1: Hybrid models of the author/reader-driven spectrum [45]

2.2.1 Visual narrative elements

Visualizations can contribute towards making data-driven stories more engaging to the reader. More recently, McKenna et al. [36] have corroborated this hypothesis with a survey where readers ranked the engagement of stories with visualizations higher than the text-only alternative. McKenna et al. [36] also identify seven impactful characteristics of visual narratives which the authors named *visual narrative flow factors*:

- *Navigation input* is a technique that allows the user to interact with the narrative in order to progress through it. The most common input types are scrolling (to move the document), button (click, tap or keyboard press) and, although rarer, a slider (drag or select where to be in the story). These input types can occasionally be combined into hybrid navigation systems.

- *Level of control* describes how much control a user has over the motion or animated transitions of story components. The level of control can be categorized as either being discrete or continuous. A reader has discrete control if the motion is triggered by a user action. On the other hand, continuous control means the user has control over each individual keyframe of the animation which is played as the user progresses throughout the story.

- *Navigation progress* keeps the readers updated of their place within the entire story. Usually, the progress is tightly coupled with the navigation input. Stories which use scrolling inputs generally don't explicitly add more progress feedback, relying on the position of the scrollbar. Others demonstrate the progress through dots or numbers. Exceptionally, stories may showcase domain-specific visualization elements conveying navigation progress such as a map with points depicting a journey and its stops.

- *Story layout* describes the type of layout model used in the story. A story's layout model can either be a document or a slideshow. The former seen mostly in scrollers, while the latter in steppers. A hybrid approach is also occasionally used.

- *Role of visualization* defines the relative importance of the visualizations in the entire story. This flow factor is determined by how visualizations and text interact with each other. They can (1) both play an equal part; (2) visualizations might serve only as a figure to complete part of the text story; (3) visualizations can lead the story as the text components are primarily annotating the visual narrative.

- *Story progression* is the flow factor that explains the possible story paths in a narrative. They can be divided into *linear* and *non-linear* paths. Linear, the most common, is observed when a reader is guided linearly through each point of the story. A variation of this progression is the linear skip, where users can jump backwards or forwards. Non-linear approaches are most used to include loops or deviations to the story and are usually implemented as trees or graphs.

- *Navigation feedback* combines animations, transitions or state changes on visualizations, text or widgets to denote the effects of user input.

2.3 Visualization

Card et al. define *Information Visualization* as the representation of abstract data displayed to a viewer in an interactive and visual way using computer graphics techniques as enabling technologies for the ultimate goal of improving the user's perception of the data [10]. Kirk reinforces the aspect of cognition amplification by defining information visualization as "the representation and presentation of data to facilitate understanding" [29].

Kirk also differentiates between *representation* and *presentation*. The former concerns the form in which the data is visually portrayed (e.g., using a bar chart). The latter concerns the design decisions like interactivity, annotations, colour and composition.

2.3.1 Visualization taxonomies

Some authors identify more specific areas inside of information visualization. Card et al. [10] identifies *scientific visualization* as a subcategory of visualization that applies the concepts of information visualization to scientific data, which is typically physically-based. A more specific case of scientific visualizations is *geovisualizations* [9], often found in *Geographic Information Systems* (GIS). These are inherently reserved for geographic data which contains both spatial and temporal variables, commonly a timestamp (or date and time) and location (defined in terms of longitude, latitude, and elevation). The core problem of this work can be considered to lie in a layer more specific than scientific visualization, but more abstract than geovisualization: spatiotemporal visualization.

Lengler and Eppler [31] identify other categories:

- Data Visualization — visual representations of quantitative data in schematic form (e.g., table and bar chart);
- Concept Visualization —- represents qualitative concepts, ideas or plans (e.g., Gantt chart);
- Metaphor Visualization — simple templates that represent the structure and organization of the information to convey complex insights (e.g., metro map);
- Strategy Visualization — complementary visualizations for the analysis, development, formulation, communication and implementation of strategies (e.g., value chain).

The authors also identify the possibility to combine the previously mentioned formats into a *Compound Visualization*.

2.3.2 Visualization pipeline

In 1990, Haber and McNabb conceptualized a visualization pipeline, named *visualization process* [23]. The proposed pipeline, shown in Figure 2.2, conceptualizes visualization as a series of transformations that convert raw data into a displayable image. According to the authors, the transformations (shown in red) progressively convert the data (shown in blue) into formats perceptible by humans while maintaining the integrity of the information.

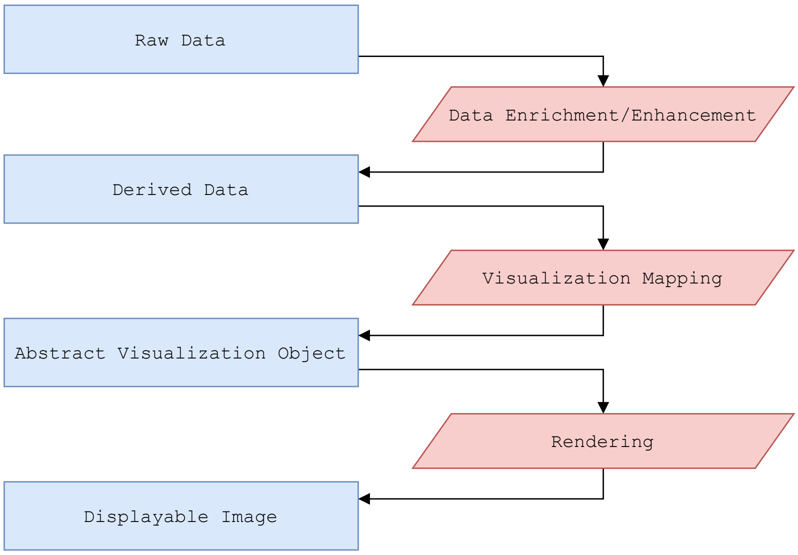


Figure 2.2: Visualization pipeline, adapted from Haber and McNabb [23]

The first transformation, data enrichment and enhancement, receives the raw data as input and outputs derived data. This transformation includes representational changes (e.g., from continuous to discrete representations or vice versa), filtering and smoothing (e.g., removing noise), and calculation of additional data (e.g., velocity, given distance and time). The second transformation, visual mapping, constructs an imaginary object called "abstract visualization object" (AVO) which conceptually defines how the data should be visualized in terms of attribute fields (e.g., colour, transparency, geometry, etc). The authors identify linear mappings which preserve quantitative information (e.g., density being linearly mapped into a grey-scale gradient) and non-linear mapping which allows the revelation of subtleties of structure (e.g., the non-linearity of depth buffers, favouring the detail in depth of closer objects). The last transformation — rendering — produces a displayable image based on the abstract visualization object. The rendering operation involves view transformations (e.g., rotation, translation, scaling, perspective division, and clipping) and optical models (e.g., culling, shading, and anti-aliasing).

Santos and Brodlie have extended this model to accommodate multivariate multidimensional data in a visualization [18]. Their model, shown in Figure 2.3, replaces the first stage of *data enrichment and enhancement* with *data analysis* and *filtering*. Data analysis is, according to the authors, a computer-centred process which defines an interpolation function to be applied to the raw data with the goal of estimating the underlying entity being visualized. Filtering is a human-centred process which allows the user to interactively apply filters in order to select the part of the data that should be visualized. Therefore, this model decouples the data resulting from data analysis from the data selected by filtering which allows for the interactivity to be independent of the pre-computations.

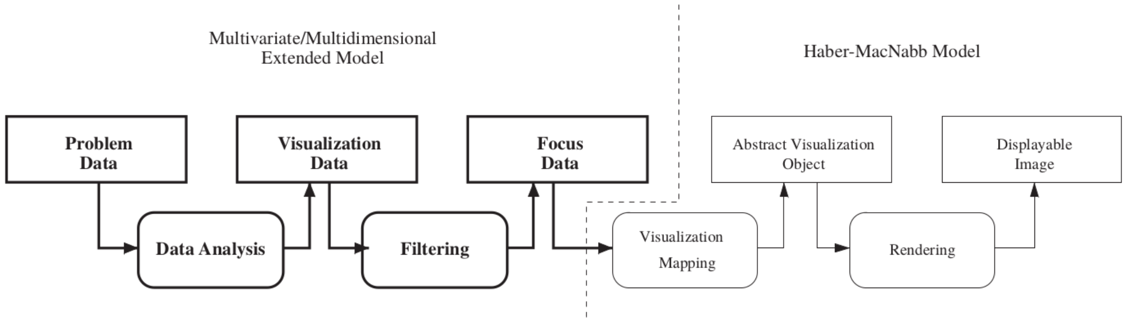


Figure 2.3: Visualization pipeline by Santos and Brodlie [18]

2.3.3 Visual variables and visual metaphors

The pipeline stage of visualization mapping, presented in the previous section, takes the processed data and creates abstract visualization objects. This mapping process from abstract and non-visual properties to visual properties encompasses the definition of *visual metaphors* [21]. Formally, visual metaphors are visual representations of metaphorical concepts [20]. The components of a visual metaphor are referred to as *visual variables* [5]. The author behind the definition of visual variables, Bertin, identifies seven of them: planar dimensions (position), size, value (brightness of the colour), texture, colour (hue), orientation and shape. These are shown in Figure 2.4 along with their *levels of organization*:

1. Associative perception — the assessment of equality of a variation or category classification (i.e., the perception of two instances of the visual variable representing the same category or value);
2. Selective perception — the ability to instantly isolate all the elements of a category (i.e., being able to answer the question "Where is a given category?");
3. Ordered perception — the instantaneous estimation of relative order between the elements (i.e., being able to order the categories without consulting the legend);
4. Quantitative perception — the immediate perception of the numerical ratio between two elements (e.g., understanding when a sign represents the double of another).

Later, MacEachren [32] documented three more variables: crispness (scale of detail), resolution (quality and blurriness) and transparency.

The domain nature of this work, spatiotemporal phenomena, dictates its planar dimensions (location), size and shape (its geometry) and orientation, limiting, from a design perspective, the variations of these variables to represent information. Therefore, Subsection 2.3.3.1 explores the colour variable, including its value, hue, and transparency. Afterwards, Subsection 2.3.3.2 analyses visual metaphors that convey motion, especially in static mediums which is the focus of this work. Finally, Subsection 2.3.3.3 explores the visual metaphors which allow different levels-of-detail in visualizations.

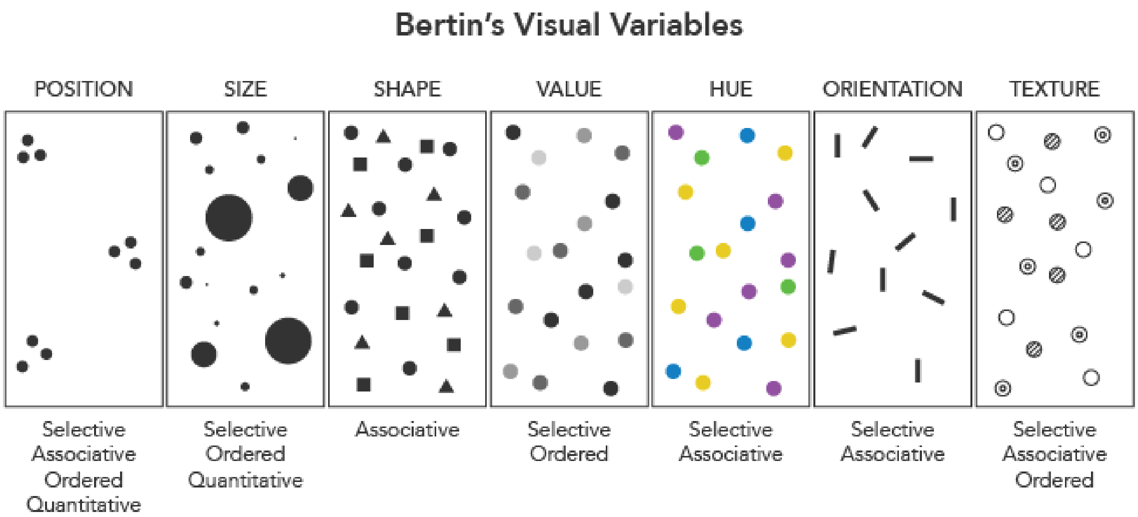


Figure 2.4: Bertin visual variables, adapted from Axis Maps [33]

2.3.3.1 Colour

The value that colours bring to a visualization is limited but powerful. In 2D space, it may not help in understanding an object's shape, its layout in an environment, or stereoscopic depth. Instead, colours are perceived as an attribute of an object, which suggests an important role in visualization: labelling and categorization [54].

Labelling, or nominal information coding, is a common requisite in visualizations for which colour is used. As stated, colour is very effective in classifying objects into categories (Figure 2.5). Ware defines a set of *perceptual factors* to be considered when choosing colours for labelling [54]:

1. Distinctiveness — the degree of perceived difference between colours placed close together;

2. Unique hues — suggests that for each category there should be one colour, sufficiently separated from the others in the colour space;

3. Contrast with background — the background colour can drastically alter the perception of a colour. One could either place a black border around the object or ensure a significant difference in luminance between both colours;

4. Colour blindness — take in consideration that there is a portion of the viewers which cannot distinguish between some colours;

5. Number — limit the number of colours to a range of five or ten;

6. Field size — the realization that very small colour-coded objects are difficult to distinguish. Smaller objects should have highly saturated colours to increase discrimination, while larger objects the colours should have lower saturation, differing only slightly between each other;

7. Conventions — consider common colour conventions, like using red for high temperatures or for dangerous objects, without disregarding cultural differences (death can be depicted using black in most European countries but in China the same phenomenon is symbolized by the green colour).

With these aspects in mind, Ware recommends a set of 12 colours for labelling: red, green, yellow, blue, black, white, pink, cyan, grey, orange, brown and purple [54].

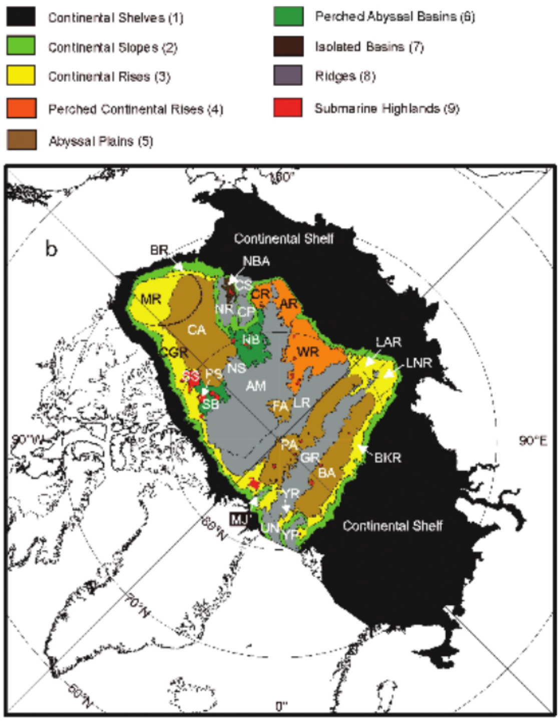


Figure 2.5: Arctic seafloor features as an example of categorization with colours [54]

A different use of colour is *pseudocolouring* — "the technique of representing continuously varying map values using a sequence of colours" [54] — and it is commonly seen in maps. Many coding schemes exist for pseudocolouring. The most common in scientific visualizations is the physical spectrum (i.e., from red to violet, shown in Figure 2.6). However, this is not a perceptual sequence without a legend. Ware exemplifies this by realizing that, if given a set of colours from the spectrum to people to place in order, the results would vary between people. In contrast, sections from the spectrum can be perceptually ordered (e.g., yellow to red) but not the complete spectrum.

Colour is thus a common feature of visualizations, but its usage often disregards human perception. The perception of colour is always limited by the *gamut* — the set of all colours that can

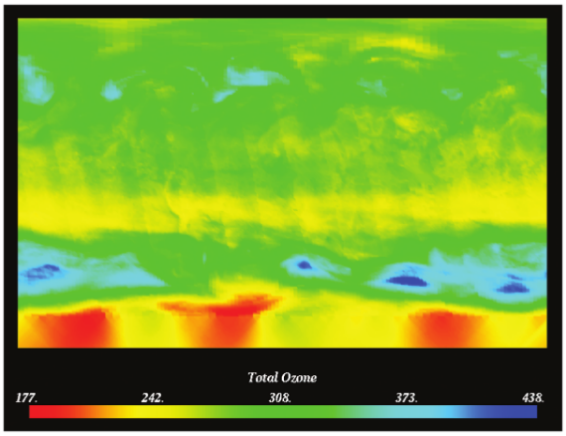


Figure 2.6: Ozone concentration in the atmosphere as an example of pseudocolouring with a physically-based colour spectrum [54]

be produced by a device or sensed by a receptor system [54]. For instance, when using colour scales that map to data values (e.g., blue to red to depict temperature) there are two main approaches [24]: (1) linear scaling — the differences in the data linearly correspond to differences in the colour scale; (2) non-linear scaling — the differences in the data correspond to differences in the colour scale in a non-linear way. The second approach allows visualization designers to take human perception into account. A possible use case for this technique would be to emphasize differences in certain parts of an attribute's domain (e.g., using logarithmic mapping for the lower values and exponential mapping for the higher ones).

The need to display data in layers is common in visualizations, especially with spatiotemporal data. A technique used for presenting multiple layers of data is treating each as a semi-transparent layer [54]. Although useful, this might lead to visual interference between layers which can be minimized by separating them through visual channels, like colour.

2.3.3.2 Motion

Displaying motion in dynamic visualizations often implies adding animation [24]. For instance, to represent the movement of an object, one could define the animation path as a mathematical function of time and for each frame displayed, the object's position would be updated by sampling the mathematical function with the current timestamp. This method is characterized by one big disadvantage: the time to "consume" the information regarding the movement is equal to the time it takes for the object to complete its movement. As a possible solution, one could envision the display of changes through discrete steps which can immediately tell the story of a phenomenon. However, representing such dynamic features in static mediums is difficult, but it has been accomplished in the past. For the remainder of this section, various visual metaphors for representing motion in static visualizations are presented.

Onion Skinning *Onion Skinning*, or *ghosting of multiple images* as introduced by Tufte [52], is a visual metaphor used to depict motion in static visualizations. With this technique, multiple states of an object are simultaneously visible. These additional states are commonly drawn with diminishing visibility in relation to the initial or final state [7]. This technique is commonly found in animation, either on paper as demonstrated by Disney [49] or in authoring tools like Autodesk 3ds Max [38]. McCloud demonstrates this technique, introduced as "multiple images of the subject", to depict motion within comic book panels [35], a form of static visual narratives. The author also identifies other techniques for the same purpose, as seen in Figure 2.7, based on motion lines either from the subject or the background. Onion skinning is also capable of representing velocity and acceleration. This is possible because it displays a position across time and therefore, if all the intermediate states temporally differ in equal amounts between each other, it is possible to visually estimate on velocity and acceleration. For example, if all the intermediate positions are spatially separated by similar distances, one can conclude the object is moving at a constant speed. Onion skinning can also be used to depict morphological changes — transformations to the geometry of an object — as seen in the work of Vronay et al. [53], where the authors develop a system that performs shape morphing. In it, artists can draw two shapes and the system generates intermediate shapes that smoothly interpolate between them. Its visualization, shown in Figure 2.8 uses *Smart Onion Skinning*, with the "smart" concept being the previously mentioned contrast, through transparency, between key-frames and the frames in-between.

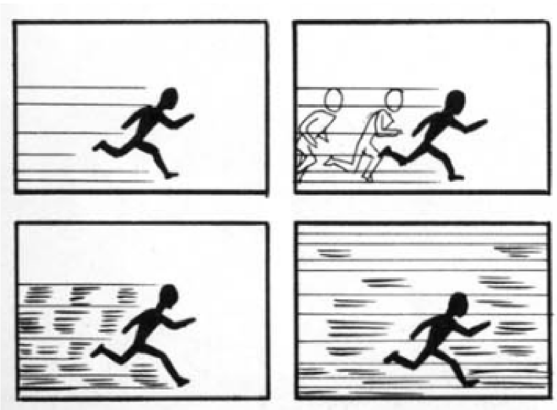


Figure 2.7: Motion representation techniques identified by McCloud [35]

Figure 2.8: Smart Onion Skinning applied to shape morphing [53]

Movement in cartography In 1967, Bertin provided a new perspective to the visualization field with his semiological approach to the visual arts. The concept introduced, the *semiology* of graphics, defines the principles and characteristics of signs and symbols that rule the arts and suggests objective criteria for art criticism [5]. Bertin acknowledges that representing movement in cartography, a static visualization medium, is a problem and suggests three solutions:

1. Drawing a series of contiguous images showing the various stages of the object. However, the author recognizes its limitations as a long series might hinder readability.

2. Drawing the path of the moving object (shown in Figure 2.9). Considered by Bertin to be the best approach to represent continuous movement, this technique traces the path and its semiotic depends on the object (a point traces a line, and a line or an area traces an area). However, as movements become complex, the area traced will become cluttered, failing to indicate direction.

3. Utilizing a retinal variable (e.g., size, colour hue, colour saturation, etc). The representation of movement would be accomplished by corresponding the steps of the object with the ordered retinal variable. If colour transparency is used as a retinal variable, the result would be similar to using onion skins.

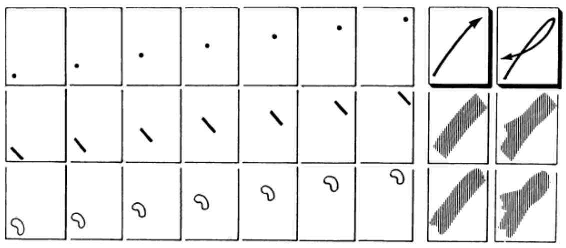
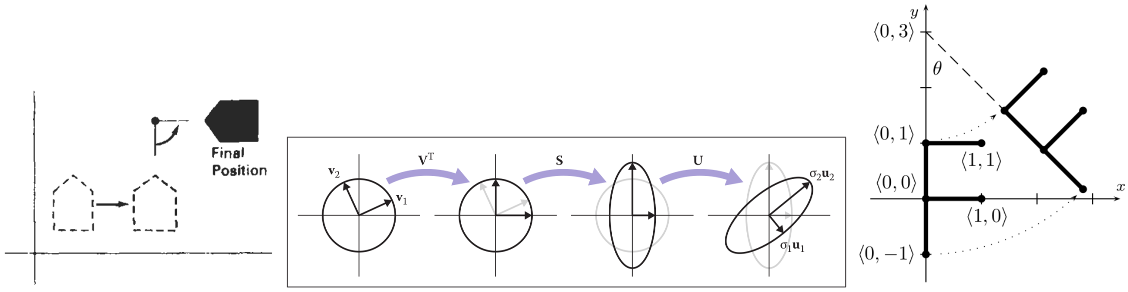


Figure 2.9: Path tracing of a point, line and area as a visual metaphor for movement [5]

Geometric transformations *Semiology*, the study of signs and symbols and their representation towards conveying meaning, was first applied to graphical visualizations by Jacques Bertin [5]. This section takes a semiotic approach in the representation of spatiotemporal phenomena changes by analysing common static visualizations and their visual metaphors — visual representations of metaphorical concepts [20].

As shown in Section 2.1.1, most spatiotemporal phenomena changes correspond to geometric transformations. These are often represented as static visuals in Computer Graphics books. Figure 2.10a shows the translation of a pentagon which is then rotated in relation to a pivot. In terms of visual metaphors, the image contains plenty. First, the usage of dashed lines in drawing the polygon to represent states that are not final (both initial and intermediate). This visual cue was also identified by Tufte [52]. Second, the movement associated with the translation is represented by an arrow from the previous state to the next. Third, the rotation around a pivot is depicted by two symbols — the dot, for the pivot, and the rounded arrow for the angle. Fourth, the polygon in the final position is rendered differently (filled with black and with solid lines) to indicate higher importance. Figure 2.10b, on the other hand, deals with previous states of the object with transparency to decrease its visibility in favour of the new state caused by the transformation, much like the onion skinning method described in Section 2.3.3.2. Figure 2.10c features a visual aid to help understand how the transformation affected the object by mapping some of the vertices from the initial position to the corresponding vertices in the final position — a technique henceforward referred to as *vertex mapping*.



(a) Polygon translation followed by a rotation around a pivot [25]

(b) Sequence of geometric transformations [34]

(c) Vertex mapping [8]

Figure 2.10: Visual metaphors for geometric transformations

2.3.3.3 Providing context and detail

Visualizing multivariate data regarding both temporal and spatial domains is challenging, especially when dealing with large datasets. Nonetheless, some documented techniques minimize this problem by providing users with multiple levels of detail.

Base map Bertin also introduced the concept of *base map* as a design element, consisting of a set of known reference points, crucial for situating the overlaid elements. An example of a base map can be seen in Figure 2.11 The author defines two good practices to consider when drawing base maps:

1. the base map must include all elements of identification necessary for the construction and reading of the map;
2. the base map must be dominated by the new elements being mapped.

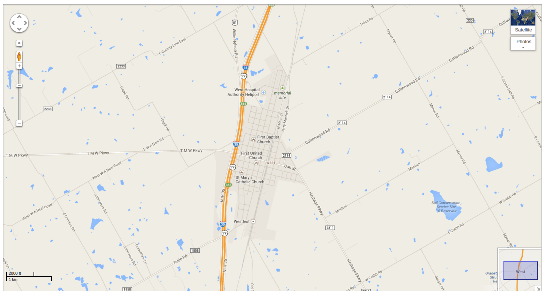
Despite the example shown being a geographical map, this concept should be thought of as a generic method for providing spatial context to the objects being rendered on top of it.

Information hiding Tominski et al. [50] used the concept of *information hiding* applied to 3D information visualization. By taking an event-based approach, the author allowed the users to

Figure 2.11: Base map of Paris in a study by Chombart de Lauwe [14]

specify which events were of interest, hiding information irrelevant to them and only visualizing the ones which are relevant. As a consequence of information hiding in some visualization methods, visual clutter is mitigated.

Multiple level-of-detail interfaces Cockburn et al. published an extensive review of interfaces which allow for multiple levels of detail, alleviating the previously discussed problem [15]. Four categories were distinguished by the authors: *Overview+Detail*, *Focus+Context*, *Zoom*, and *Cue*-based techniques. Overview+Detail is a design element of an interface characterized by simultaneously displaying both an overview and detailed view of an information space, each in a distinct spatially-separated presentation space. This technique can be seen in scrollbars (which represent the location in a document and the part that is currently visible), thumbnails (such as Google Slides which are also used for navigation), and maps (Figure 2.12a) or mini-maps (Figure 2.12b). Focus+Context is a technique similar to Overview+Detail as it also simultaneously displays focused and contextual views of the information space. However, it differs by seamlessly showing, in a single display, the focus view within the context view without spatial or temporal separation. A successful example of this technique is the folding features of code editors, like Visual Studio Code (shown in Figure 2.13a). In this example, there is one function whose body is being displayed (focused view) while the others are folded (context view). Zooming is a common interactive feature in interfaces. Like the other techniques, zooming supports both focused and contextual views. However, zooming temporally separates those views, meaning they cannot be seen at the same time. Instead, a user must interact with the visualization by magnifying (zooming in to focus) or demagnifying (zooming out for context). Animation is an important aspect with zoom to help users perceive the relationship between pre-zoom and post-zoom states. Finally, cue-based techniques modify how objects are rendered. Their main use case is in search boxes or filters. The data items that satisfy the criteria are rendered in a way that highlights them, as seen in Figure 2.13b.

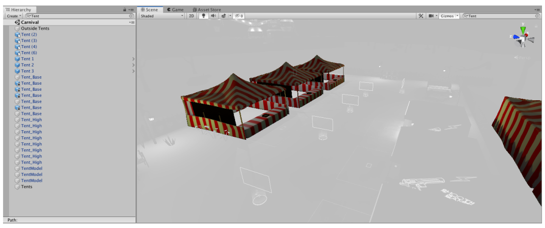


(a) Google Maps

(b) The Witcher 3: Wild Hunt

Figure 2.12: Examples of Overview+Detail. Figure 2.12a shows Google Maps with a detailed view of West, Texas, U.S.A. and an overview of the city in the right bottom corner. Figure 2.12b shows the video game *The Witcher 3: Wild Hunt* with a mini-map as overview in the right upper corner.

(a) Visual Studio Code



(b) Unity Engine

Figure 2.13: On the left, an example of Focus+Context in Visual Studio Code representing its folding features. On the right, an example of cue-based techniques in Unity, greying out the objects that do not match the search criteria and leaving the others fully rendered.

2.3.3.4 Visual clutter

Any information that is presented to viewers, creates cognitive load: requesting users to process that information with their brains. Visual clutter is a state in a visualization that creates excessive cognitive load, hindering the transmission of the message being conveyed by the visualization [30] or leading to a degradation of performance at some task.

Visual clutter can originate through the excess of items, their representation or organization [43], taking up space without an increase of understanding. There are two common techniques for addressing the excess of items to reduce visual clutter. First, allowing users to filter what information or objects are viewed. Secondly, designing visualization frameworks that implement multiple level-of-detail interfaces [43].

Besides suggesting the use of white space and contrast, Knaflic describes how the understanding of six of the Gestalt principles of visual perception can be applied towards reducing visual clutter due to item representation or organization [30]:

1. Proximity — People perceive that certain objects are part of a group if they are physically close together;
2. Similarity — If the objects feature similar colour, shape, size, or orientation, people consider them as belonging to a certain group;
3. Enclosure — Objects which are physically enclosed together are perceived to be part of a group;
4. Closure — People generally prefer simpler designs and constructs that are ubiquitous (thus already cemented in one's head);
5. Continuity — The eyes seek the smoothest path and naturally create continuity in what is seen even where it may not explicitly exist;
6. Connection — If objects are physically connected (e.g., through a line), people tend to understand that they are part of a group.

2.3.4 Visualization methods

The goal of visualization is to help people understand data by exploiting the human visual system's capabilities of uncovering patterns or identifying events. Visualization methods are the artefacts that define how the data is displayed, by making use of visual metaphors to convey meaning with signs (as seen in Section 2.3.3). This section focuses on the structure of visualizations as a means to represent spatial and temporal information.

2.3.4.1 Time-dependent data

Time-dependent data visualizations have been in use for centuries and for a wide range of industries, like science and engineering. Firstly, in order to define what would constitute an effective time-dependent visualization, MacEachren proposed the following questions which the visualization method should be able to answer, regarding the tasks to be performed [32]:

- *Temporal existence*: Does a data element exist at a specific time?
- *Temporal location*: When does a data element exist in time? Is there any cyclic behaviour?
- *Temporal interval*: How long is the time span from beginning to end of the data element?
- *Temporal texture*: How often does a data element occur?
- *Rate of change*: How fast is a data element changing or how much difference is there from data element to data element over time?

- *Sequence*: In what order do data elements appear?
- *Synchronization*: Do data elements exist together?

Secondly, various taxonomies regarding time-dependent data have been proposed. Frank proposes a taxonomy for the time axis [22]:

- *Discreteness*: Distinguishes between discrete time points and interval time. The former features no duration in time and can be seen as a timestamp, while the latter is an interval with an established time duration, specified by two time points.
- *Linearity*: Distinguishes between linear and cyclic time. The former defines a linear time axis with two specified extremities (start and finish time points), while the latter is able to define cyclic behaviour (e.g., seasons).
- *Continuousness*: Distinguishes between continuous and ordinal time. The former enables the quantification of time differences between points in the axis, while the latter can only establish relations of precedence between the time points.
- *Layout*: Distinguishes between ordered, branched and multiple perspective time. An ordered time axis considers events which occur in temporal order (one after the other). Branching the time axis allows for multiple alternative data values from one time point. Multiple perspectives allow the attribution of more than one data element per time step, defining parallel events in time.

Müller and Schumann distinguish the visual representations of time-dependent data between static and dynamic [39]. Static representations are not dependent on viewer time, meaning the visualization does not change over time automatically. Instead, modifications that occur to the visualization result from user interaction. Dynamic representations are dependent on viewer time, meaning the visualization changes as a function of time itself as they are being displayed without user interaction. Finally, some authors, like Heer et al. [26] and Müller and Schumann [39] collect several methods for representing this type of data. For static representations, there is a variety of charts and graphs: index charts, stacked graphs, horizon graphs, circle graphs, spiral graphs, etc. However, none of these variants is capable of incorporating the spatial dimension as a variable as well. Instead, some representations use the spatial dimension for contextualizing the data, as seen in a *Lexis Pencil* visualization [39]. Dynamic representations, on the other hand, make use of time itself to represent the data, liberating the canvas for data regarding a timestamp. Mapping the temporal aspect of the data directly onto the time control of the representation emerges as a natural approach. A common implementation of this idea is computer-aided animation, where each data element corresponds to a frame of the animation. Nonetheless, this type of representation leads to a possible problem: the time taken for the user to consume the information is directly mapped (even if not linearly) to the duration of the animation, which in turn relates to the time interval of the available data [39]. This time duration can either be too lengthy becoming unfeasible to detect slow-paced changes, or too short limiting the perception of fast-paced changes. The authors suggest adjusting a scale factor to speed up or down the animation, even though this may imply prior knowledge of the data. A more detailed definition of this problem can be found in Chapter 3.

2.3.4.2 Space-dependent data

Space-dependent data visualizations, similarly to their temporal counterpart, have been in use for centuries. With that said, it would be expected for the literature to be as extensive as it is regarding time-dependent data. Most authors direct the spatial nature of data to the more specific field of geographic data. As seen in the previous section, Heer et al. [26] compiled numerous visualization methods, categorized by the type of date or usage. Among the methods, the authors identify methods for geographical data, including multiple variations of maps. Nonetheless, maps are capable of other applications apart from geography, such as using a house map in architecture. The maps presented include *Choropleth Maps*, shown in Figure 2.14, *Graduated Symbol Maps*, and *Cartograms* which all aggregate data per geographic region, differing on the representation of the data.

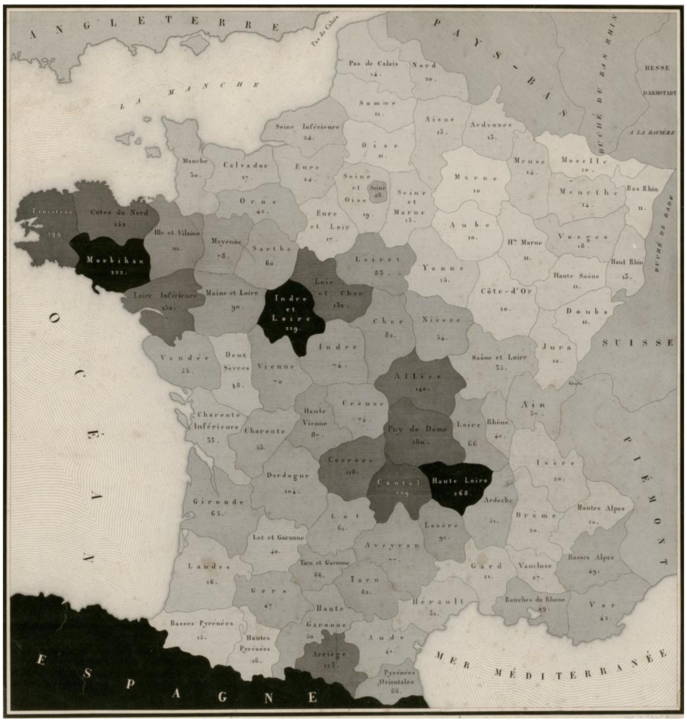


Figure 2.14: Choropleth map with shadings from black to white representing the distribution and intensity of illiteracy in France in 1826 by Charles Dupin

Perhaps the closest to representing spatiotemporal phenomena are the *Flow Maps*. Flow maps are capable of representing the movement of a quantity in space and time through the use of flow lines. Flow lines are a type of visual encoding used to represent multivariate information by varying direction, line thickness, colour and position. The most recognizable example of a flow map is the iconic representation of Napoleon Bonaparte's army through Russia in 1812 by Charles Minard. The map, shown in Figure 2.15, features multivariate data, including both temporal and spatial information and is considered the "best statistical graphic ever drawn" by visualization expert Edward Tufte [51]. However, while thriving in representing masses, this method is insufficient for representing geometry.

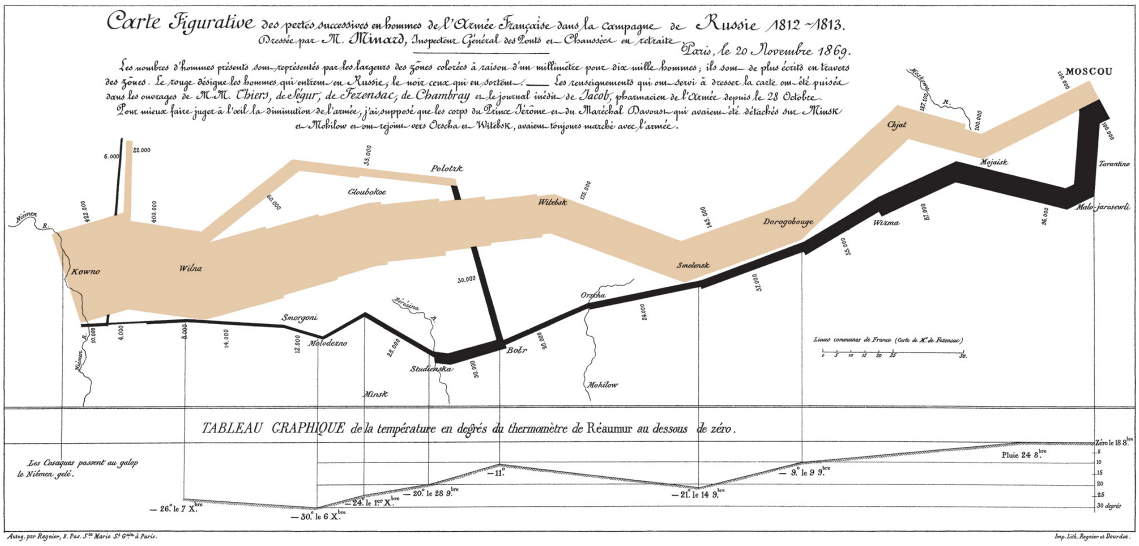


Figure 2.15: Flow map by Charles Minard of Napoleon's army campaign in Russia in 1812

On the other hand, *Isochrone Maps*, shown in Figure 2.16, are maps that, given a number of snapshots, draw an area for each timestamp (or time interval) with the same instance of the used visual variable (commonly texture or colour). Although its most common use case is to represent areas of equal travel times, by extending this concept, it is possible to represent object area changes between those snapshots with isochrone maps [5].

Geographical information systems (GIS) are computer systems that store, manage, analyse and visualize geospatial data [12] through the use of maps. These systems moved towards 3D [13], as some user needs are better fulfilled in that space, particularly when altitude, the third dimension, is required for such tasks. Others remain in the 2D space as it eases perception when another spatial variable is unnecessary (e.g., weather forecast). Despite its increasing usage, most tools either show static data without spatiotemporal phenomena transformations over time taken into account (Figure 2.17) or show dynamic data through animations (Figure 2.18) which can lead to possible viewing duration problems, explained in the previous section.

[1] ArcGIS. Geographic information system for working with maps and geographic information. Available at https://www.arcgis.com/. Accessed in 2020-01-24.

[2] Ventusky. Meteorological data visualisation application. Available at https://www.ventusky.com/. Accessed in 2020-01-24.

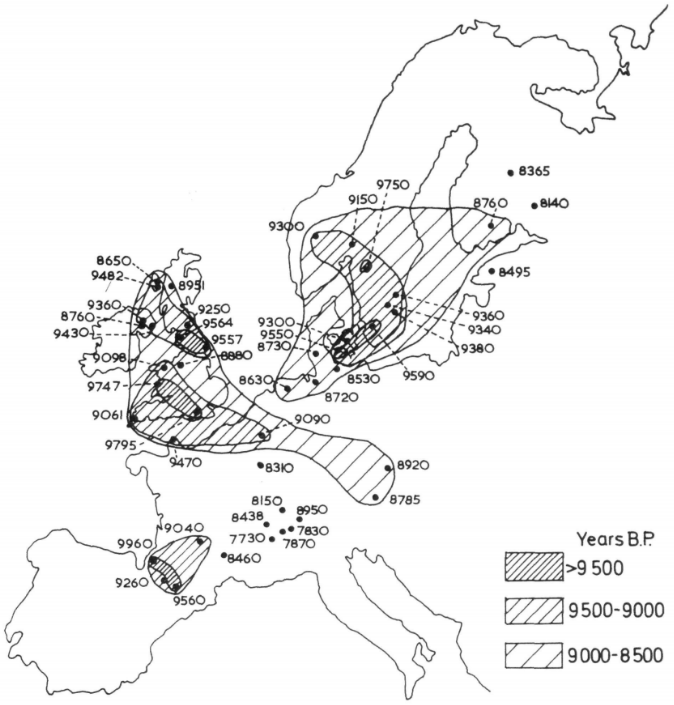


Figure 2.16: Isochrone map using texture as visual variable depicting possible regions of refugia and migration routes of *Corylus* [16]

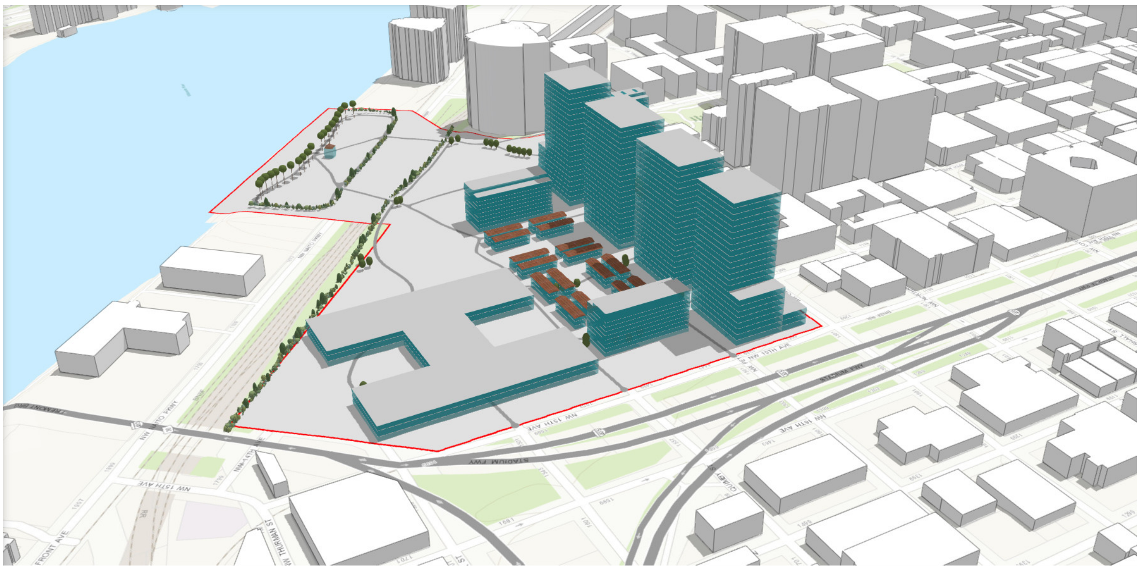


Figure 2.17: Static visualization of new buildings in ArcGIS [1]

2.3.5 Visualization evaluation

Defining and measuring the effectiveness of a visualization is a common challenge of the field. Ying Zhu contributed to this research by initially reviewing both the definition of effectiveness in

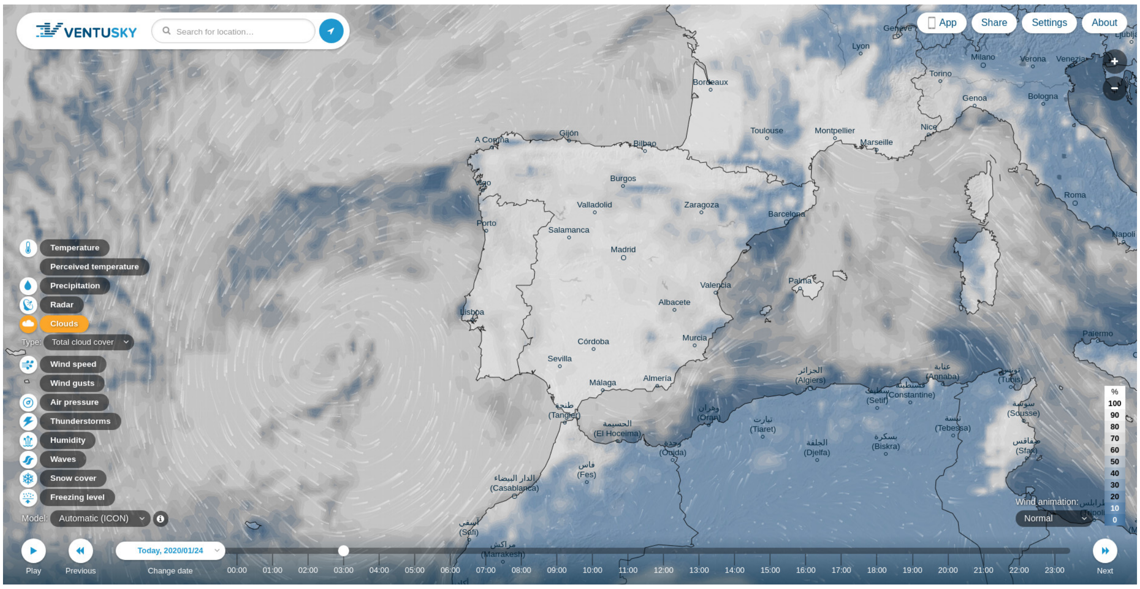


Figure 2.18: Dynamic visualization of clouds in Ventusky [2]

visualizations and its measures, and then proposing a solution as a framework [58]. The definition of *effective visualization* is inconsistent across the literature. Some authors, like Wattenberg and Fisher [55], have a data-centric view, suggesting that the structure of a visualization should match the structure of the data being represented. Others authors, like Amar and Stasko [1], have a task-centric view, believing that the effectiveness of a visualization is specific to a task.

Regarding the evaluation of the effectiveness of a visualization, there are generally two common methods: heuristic evaluation and user studies. The former evaluates effectiveness using principles from visualization experts. This method is therefore dependent on the rules postulated by the author of reference which can be rather subjective, ambiguous, or lack context and application procedures. The latter evaluates effectiveness through a more objective perspective by measuring some common metrics, such as task completion time and error rate, on actual users, typically with a significant sample, so that the results are expected to be more robust.

The biggest difficulties in performing user studies are the lack of standardization and the possible lack of details regarding what impacts those metrics. As a culmination of this, Zhu defines effectiveness in data visualization in terms of three principles (accuracy, utility, and efficiency) and identifies quantitative and qualitative measures to evaluate it. These can be seen in Table 2.2.

Other authors, such as Plaisant [41], also acknowledge there are challenges in information visualization evaluation. This author summarizes the current evaluation practices and describes some common problems. The practices identified consist of:

1. Controlled experiments comparing design elements;
2. Controlled experiments comparing different visualization tools;
3. Usability evaluation (studies conducted to receive feedback from users and the problems they encountered);

	Quantitative measurements	Qualitative measurements
Accuracy	Measure the number of interpretation errors	Interview Observation Expert-novice comparison
Utility	Measure the number of achieved benchmark goals Record the number of times a visualization design is selected by users to conduct a task	Interview Observation Expert-novice comparison
Efficiency	Record task completion time Record eye movements Measure the learning curve	Visualization complexity analysis Interview Observation Expert-novice comparison

Table 2.2: Measures of visualization effectiveness [58]

4. Case studies in realistic settings (studies with users performing real tasks in their ordinary environment).

The challenges are threefold:

1. Matching tools with users, tasks and real problems: the authors defend that utility should be demonstrated in a real setting with a set of users and given an application domain;

2. Improving user testing: the authors believe there is room for improvement when it comes to empirical studies such as including more complex tasks, using longitudinal studies, testing with domain experts, and measuring uncommon metrics (like learnability);

3. Addressing universal usability: it is challenging to make tools useful for the general public but techniques like multi-layer interfaces can aid adoption and comprehension by allowing users to find the right level of complexity for them.

2.4 Enabling technologies

Developing a working prototype not only requires high-level knowledge of the domain (spatiotemporal phenomena) and field (visualization) but also demands technical expertise. The following sections focus on addressing target programming languages and environments, 3D libraries and algorithms that were considered closely related to the development of the prototype that will be used to test the proposed solution.

2.4.1 Programming languages and environment

With the introduction of scripting engines in web browsers, it is possible to create applications that run on a browser without installing additional software. Web applications also have the benefit of ubiquitous execution, since most platforms support browsers (cross-platform and accessible minimum requirements) and browsers implement standards (browser-agnostic code). The technologies used in this environment are (1) HTML (for page document structure); (2) CSS (for presentation, formatting and layout); (3) JavaScript (for implementing functionality). In addition, containerization technologies, such as Docker[3], further enable application portability, eliminating concerns regarding run-time dependencies while maintaining performance requirements [2].

2.4.2 Existing libraries

The underlying process for generating and presenting images such as 2D polygons is referred to as *rendering*. Rendering, in the context of web browsers, can be classified into declarative or imperative [27]. Declarative rendering, like SVGs, defines how the geometry must look like, while imperative approaches, like WebGL, define steps for how an image would be displayed. Generally, most declarative methods lack dynamism and perform worse than imperative methods [28]. On the other hand, a common disadvantage of using a low-level API like WebGL is the increased *cost to the developer* (number of lines of code and time to develop) [28]. Due to this drawback, frameworks that abstract the low-level intricacies of WebGL are commonly used amongst developers. Many of these frameworks exist, but the most popular are:

1. *Three.js*[4] — an open-source JavaScript 3D library. It mainly focuses on WebGL-based 3D renderings, but it is also used for 2D visualizations and it supports custom shaders. Three.js also has very thorough documentation, API reference, and examples and is, by far, the most popular for graphics on the web.

2. *PixiJS*[5] — an open-source 2D JavaScript engine. It also supports custom shaders, but the documentation is based almost exclusively on the API reference and examples.

3. *Babylon.js*[6] — an open-source WebGL-based graphics engine. It supports custom shaders and features excellent documentation and examples.

4. *D3.js*[7] (Data-driven documents) — an open-source JavaScript library for manipulating documents based on data. It is mainly used as a visualization tool based on SVGs, limiting the performance factor and customization. Its documentation is also lacking and essentially based on the API reference and examples.

2.4.3 Algorithms

Polygon simplification When using vertex mapping, explained in Section 2.3.3.2 and shown in Figure 2.10, it is possible to come by the problem of visual clutter (Section 2.3.3.4) if applied to all the vertices. To pick a meaningful set of vertices for which to draw the mappings, one could use a polygon vertex reduction algorithm, like *Ramer-Douglas-Peucker* [19, 42], but instead of simplifying the polygon one would annotate which vertices to use. Hence, this problem lies in the

[3] Docker. Containerization technology. Available at `https://www.docker.com/`. Accessed in 2020-06-04.
[4] Three.js. JavaScript 3D library. Available at `https://threejs.org/`. Accessed in 2020-01-31.
[5] PixiJS. 2D JavaScript engine. Available at `https://www.pixijs.com/`. Accessed in 2020-01-31.
[6] Babylon.js. WebGL graphics engine. Available at `https://www.babylonjs.com/`. Accessed in 2020-01-31.
[7] D3.js. Visualization tool based on data documents. Available at `https://d3js.org/`. Accessed in 2020-01-31.

data enrichment/enhancement transformation of the visualization pipeline, described in Section 2.3.2.

The Douglas-Peucker algorithm receives a set of vertices as input. It then registers which of them to keep, resulting in the output for the algorithm. It starts by keeping both the first and last vertices since it is the most basic simplification possible. Then, it traces a line segment from the two current vertices (initially the first and last). After this, it measures the distance of each point to the line segment, registering the vertex furthest away from it, if the distance is higher the threshold parameter ε. The algorithm proceeds by recursively tracing two new line segments (one from starting point to the newly defined vertex and one from it to the end point) and repeating the process.

Transparency rendering Throughout the literature review, the importance of semi-transparent objects has been reiterated, for instance in onion skinning (Section 2.3.3.2). However, rendering semi-transparent is a problem in most computer graphics rasterization pipelines and, therefore, a problem in the *rendering* stage of the visualization pipeline described in Section 2.3.2. When alpha blending the pixel fragment for a semi-transparent object with the previous values of the depth buffer, the depth buffer will be updated with the depth value of the semi-transparent object. This has the consequence of subsequent semi-transparent objects with higher depth value (further away from the camera) to not be rendered behind the first semi-transparent object since it fails the depth test (the new object's depth is higher than the old one) even though it would still be visible because of the transparency of the objects [37].

The most well-known algorithm for this problem is *sorted transparency*. The algorithm is very basic: first, you would render all opaque objects, then sort all the semi-transparent objects by depth, and finally render the semi-transparent objects [37]. By this definition, the bottleneck is the sorting of the polygons, which is not a problem for a small number of polygons.

2.5 Summary

This chapter presents the outcomes of the literature review regarding spatiotemporal phenomena, visual narratives, visualization, and the enabling technologies available.

Concerning spatiotemporal phenomena, their features were analysed, classifying them into spatial, temporal, spatiotemporal, and non-spatiotemporal. It was explored how some of them could map directly to geometric transformations. An important concept established in this section is the events of interest which proposes the freedom of the user to select which of the features previously identified would be of interest and the thresholds it needed to meet before marking an event.

The visual narratives section presents definitions and good practices for visual storytelling with data. Throughout the section, it is reiterated the importance of interaction and a reader-driven approach towards engaging the viewers.

Concerning spatiotemporal data visualization, the most important visual metaphors, visual variables, and methods available were analyzed, as well as evaluation practices. In terms of depicting movement, it was considered that the onion skinning metaphor is very effective and broadly used in static mediums. For providing context, it was identified the base map principle as important for situating the overlayed objects, and multiple level-of-detail interfaces, like Overview+Detail, for giving a global and detailed view over the information space. Afterwards, there was a review of "out-of-the-box" methods for visualizing time and space-dependent data. While exploring temporal data visualization methods it was discovered the dichotomy between static and dynamic representations, with the latter suffering from problems related to the time required to consume the information and the former's lack of potential for representing the spatial dimension. In the spatial dimension, it was concluded that maps play a leading role in visualizations, with the literature delving more into the geographic aspect of them. For the spatial dimension, it was also seen in both static and dynamic approaches. The former with flow maps which could not quite represent most phenomena changes of the overlaid object over time. The latter suffering from the same problem as in the temporal domain which is the necessity of consuming the entire visual content.

Lastly, this chapter contains a brief presentation of the enabling technologies available to develop a prototype of the proposed solution. Some graphics libraries were reviewed and it was concluded that Three.js has the best feature support, well-structured documentation with extensive examples, and a large community due to high adoption and popularity.

Chapter 3

Problem Statement

This chapter thoroughly defines the problem addressed in the current work. Section 3.1 lists and describes the issues identified in the spatiotemporal visualization methods reviewed in Chapter 2. Afterwards, Section 3.2 provides a set of requirements which are considered essential for an appropriate solution. Then, Section 3.3 defines the scope of the work given the available resources and priorities. Finally, Section 3.4 presents the research questions that are at the core of the current work.

3.1 Current issues

As presented in Chapter 2, many visualization methods have been developed by other authors to provide solutions in terms of analysis, communication and perception of spatial, temporal and spatiotemporal data. However, from this study perspective, the presented methods were not able to effectively and efficiently tell the story of a spatiotemporal phenomenon in a general manner. As such, the problem identified for this work is defined as a union of the following issues in current visualization techniques:

1. **Linearity of time-mapping in dynamic visualizations**: Common implementations of dynamic visual narratives directly (or linearly) map the temporal aspect of the data into the visualization method. If mapped directly, data representing a time interval of one year would result, for example, in an animation with one-year duration. Nonetheless, this approach is seldom used, preferring a linear mapping that allows adjusting the duration of the resulting visualization as a linear function of the temporal data. Taking the previous example, the one-year data could be condensed into a one-minute animation, preserving the relative occurrences of transformations. It is, however, generally possible in these types of visualizations to modify the speed at which it is played.

   There are a few issues with these visualization methods. First, the user, not having prior knowledge of the phenomenon, must consume the entirety of the visualization in order to understand the evolution of the object. Second, there are three specific scenarios that can be problematic with these approaches: the occurrence of sudden transformations whose effects do not last might be imperceptible to the user, resulting in loss of perceived detail; the occurrence of fine-grained transformations that might not be perceived by the user, despite being of interest, due to their small changes; the existence of long periods of object immutability or slow occurring transformations, resulting in increased time spent analysing the behaviour of the phenomenon.

2. **Representing object dynamism in static visualizations**: Spatial and temporal data have been represented through static mediums of visualization, such as comics, for a long time. However, there has not been a predominant method for visualizing data with both spatial and temporal variables (spatiotemporal data). This fact comes from the inherited difficulty of conveying the dynamism of an object in a static fashion. This is a limitation that, although less problematic for displaying a small number of transformations (e.g., teaching geometric transformations in computer graphics books), it carries difficulties in the creation of a static narrative that explains the story of an object that goes through numerous distinct or compound transformations.

3. **Filtering through events of interest**: Being able to filter through visualization components is an interactive feature which arose with the *Digital Age*. Through filtering, a user can remove irrelevant information for a given task, data domain, or user preference to improve the efficiency and, sometimes, the effectiveness of the visualization tool.

   However, in the domain of spatiotemporal information, there has not been conceptualized, to the best of the researched knowledge, a framework that would allow users to filter through the transformations that occurred to the object. An example of a scenario where the lack of this feature becomes problematic is when the object is regularly performing a given transformation — translations, for example —, but the user is only interested in particular object scales, for instance. In this scenario, the viewer would have to consume the complete narrative, either dynamic or static, including the irrelevant translations, to observe the scale changes that occurred to the object.

3.2 Desiderata

A *desiderata*, Latin for "things desired", is, in literal terms, a set of needs or wants. In the context of scientific research, it is a collection of requirements essential for the proposed solution to solve an identified problem. In the case of this work, these requirements are the following:

1. **Use visual metaphors** that are effective representations for each object transformation. Each type of transformation should have different visual metaphors, so the user can easily and intuitively associate them with its corresponding transformation, improving perception.

2. **Automatically build a static visual narrative** to efficiently present the story of a spatiotemporal phenomenon, including visual aids, interactions, navigation and hierarchical levels of detail, thus providing the user with strong spatial, temporal and story contexts.

3. **Allow the parameterization of events of interest** so the user can define and specify which types of transformations are of interest and which are the parameters that define them as relevant. Additionally, hierarchical levels of detail for the narrative should be generated, possibly revealing more granular transformations.

4. **Develop an automated pipeline** starting with the request of spatiotemporal data, data transformation and filtering, and generating the corresponding visual narrative.

3.3 Scope

Since time is a limiting factor in developing a fully fledged solution, it is imperative to establish a realistic scope for this work. Accordingly, the choice of features to study, implement and test is given by their relative and empirical priority. As a result, the goals of this work are to (1) study, conceptualize, develop and test a framework that can automate the process of building static and interactive visual narratives for the summarization of a spatiotemporal phenomenon where (2) each transformation would be displayed with a number of visual metaphors designed to improve the viewer's perception of the object's story given that (3) the data is parametrizable to provide users with events that are of interest to the task, domain or preference.

Considering the highly exploratory nature of the current work towards spatiotemporal phenomena summarization through static visual narratives, the scope for this work is a first step that addresses a portion of the problem. Consequently, the scope of this work is set to (1) the observation of a single object's story which can suffer a (2) reduced set of recognized transformations for which there are custom visual metaphors with (3) limited possibilities of events of interest.

3.4 Research questions

Given the current visualization frameworks available for spatiotemporal phenomena, with this work the authors commit to contribute to the following research questions:

1. **Are static visual narratives effective in describing the evolution of a spatiotemporal phenomenon?** A visualization framework must foremost transmit the information to viewers, minimizing interpretation ambiguity or errors.

2. **Are static visual narratives efficient when compared to their dynamic counterpart?** Since there are dynamic alternatives to static visual narratives for spatiotemporal visualization, it must be assessed how they compare to those options in terms of visualization efficiency.

The above questions can be subdivided into two research scenarios. The first assessing the effectiveness of individual frames of a storyboard and the second evaluating the interpretation of an entire story, given the full static visual narrative.

3.5 Summary

This chapter defined the core problem of the current work. First, the issues with current spatiotemporal data visualization methods were identified, addressing both static and dynamic approaches. Static approaches lack effective visual metaphors to represent transformations, while dynamic approaches may not be efficient in terms of time to consume. Common to both approaches is the absence of parameterization to define what events are of interest to the user. Then, the requirements for a solution to this problem were established, underlying the usage of appropriate visual metaphors to represent transformations and a static visual narrative that can effectively display the story of an object, using numerous parameters to specify what constitutes an event of interest. Afterwards, the scope of this work was limited to the established goals in terms of features. Finally, the research questions the authors aim to contribute to were presented, emphasizing user interpretation of the phenomenon and the comparison with other approaches.

Chapter 4

Conceptual Visualization Framework

This chapter is devoted to documenting the proposed solution for the core problem addressed by this work, presented in Chapter 3. Section 4.1 proposes and describes a solution for those limitations by introducing a novel visualization approach for the spatiotemporal domain based on static visual narratives. Afterwards, in Section 4.2, the concept of events of interest is revisited and tied to the solution context. Section 4.3 decomposes the solution design into its visual elements, justifying its existence and representation based on the needs of communicating information and usability. Lastly, in Section 4.4 the conceptual visualization framework pipeline is presented, as well as each stage and interaction actions.

4.1 Proposed solution

Regarding the problem presented in Chapter 3, a hypothesis is stated and a solution is proposed with the goal of corroborating it. The hypothesis is that static visual narratives can improve efficiency in understanding the evolution of spatiotemporal phenomena. The solution proposed in the current chapter is a novel approach based on automated generation of interactive storyboards that summarize the evolution of a spatiotemporal phenomenon through a set of frames that represent the most relevant changes across all events of interest. Figure 4.1 shows an example of a spatiotemporal phenomenon visualized through the developed functional prototype of the proposed solution.

The above description of the proposed solution is lengthy and, at first, might be difficult to grasp. Therefore, for the remainder of this section, this description is explained in greater detail by decomposing it into key-parts. The solution description starts with the concept of automation — the automatic operation of a system —, which in this case refers to automatically generating the narrative with minimal effort from the user, abstracting the user from the creation of the visualization instance. The user should only be concerned about selecting the data and setting the

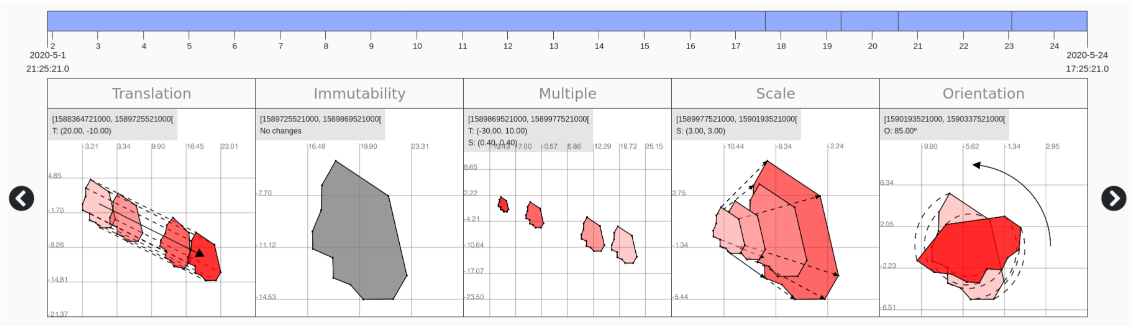


Figure 4.1: Functional prototype of the proposed solution

parameters that suit the current needs. Following this logic, it may seem counter-intuitive to mention interactivity. However, as an instance of static visual narratives, interactivity is important to update the contents being displayed, as explained in Section 2.2. The interaction from the user enables the adjustment of the level-of-detail, the filtering of the events of interest shown, and the navigation along the timeline. The storyboard aspect directs us to a mixture of common static visual narrative methods like interactive slideshows and comics. From comics, it is taken the idea of framing a varying length of time to tell a story. From interactive slideshows, it is realized the importance of user interaction to progress through the story. The storyboard is composed of frames where each visually represents a transformation of the object during an interval of time. Therefore, the set of frames that form the storyboard summarize the evolution of the spatiotemporal phenomena.

4.1.1 Fulfilling the requirements

The remainder of the current chapter addresses each of the requirements established in Section 3.2, albeit in a different order to facilitate understanding and accommodate their dependencies. Based on these requirements, a potential solution must:

1. Define suitable parameters for each spatiotemporal feature to allow specifying which events are of interest. This requirement is addressed in Section 4.2. Furthermore, these parameters are later further explored in Chapter 5 to describe how they are adjusted regarding user interaction. A user, having set the parameters, should be presented with a static visual narrative generated using a template that results from the following requirements.

2. Define the layout and the visual elements that compose the template of a static visual narrative. This requirement is addressed in Section 4.3 and describes possible layouts, navigation, and elements regarding spatial, temporal and story contexts. This requirement also encompasses the requirement of visually representing spatiotemporal features using visual metaphors, which is addressed in Section 4.3.2.

3. Define a visualization pipeline that inputs data and outputs the visual contents of the narrative through a series of stages and transformations that instantiate a static visual narrative

based on the previously defined parameters and the narrative template designed beforehand. This requirement is addressed in Section 4.4.

4.1.2 Solving the problem

The envisioned solution presented in this chapter intends to solve the current issues explained in Chapter 3. To that end, the conceptual visualization addresses them, respectively, as follows:

1. The visualization content is not linearly (or exclusively) proportional to the temporal horizon, as in dynamic visualizations. In contrast, the visual contents generated by the proposed solution are proportional to the number of detected object changes which is loosely dependent on the time interval and the events of interest parameters. This dependency means that more granular parameters might result in a larger and more detailed storyboard. Evidently, given the same parameters, a longer time interval in the observation of a spatiotemporal phenomenon might comprise more object transformations, resulting in more visual content. To some extent, the excess of visual content may result in visual clutter which is also addressed by this study.

2. Despite the static nature of each frame representation, the proposed solution can convey the dynamism of an object through effective visual metaphors and the juxtaposition of storyboard frames. Additionally, the issue of required space for the visual content, inherent in purely static mediums (e.g., paper), is circumvented through interactive navigation.

3. The proposed solution enables filtering the visual content by setting what events are of interest and the parameters that should be used to identify transformations, allowing the user to skip irrelevant transformations.

4.2 Events of interest

As presented in Section 2.1.2, the term *event of interest* was introduced by Tominski et al. by referring to the ability to define events in spatiotemporal context that are of interest to the user, given a series of parameters [50]. In the context of this work, the term *event* equates to an object change — or generically, a spatiotemporal feature — (e.g., a translation) that was identified and classified by an external framework. Object changes which are identified but are not able to be classified are, however, also considered an event — a fallback event with the name of *Unknown transformation*.

As stated in Section 2.1.2, an event is marked when it surpasses the parameters established for a certain transformation. Based on literature review, Tominski et al. identified two classes of parameters:

1. **Global parameters** that define a threshold for the value of a given feature. These parameters mark an event when the value of a feature in a certain timestamp exceeds the defined threshold: $V_t > \Delta$.

2. **Local parameters** that define a threshold on the variation of the value of a given feature between two consecutive time steps. These parameters mark an event when the difference in the value of a feature from one time step to the next surpasses the defined threshold: $V_t - V_{t-1} > \delta$.

Throughout the conceptualization stage of this work, it was pointed out that global parameters should be divided into two: one for transformations where each increment is towards the same *direction* as the previous increments and another parameter for *absolute* increments (disregarding direction changes). To illustrate this point, consider that the object suffers two consecutive symmetric orientation changes (rotation along the centroid): firstly, the object rotates 5° clockwise and, immediately after, the object rotates 5° anti-clockwise, resulting in a change of direction from one transformation to the other. A directed threshold of 10° for the orientation feature would not produce an orientation event since it would reset after the first transformation as the second changes direction. On the other hand, an absolute threshold of 10° would mark an event since the sum of both transformations would equal its value.

The concept of *Direction* is defined per transformation. In the example above, a change in direction for the orientation feature means going from a clockwise rotation around the object centroid to a counter-clockwise one, or vice versa. For translations, it indicates that the angle between translation vectors has reached an implementation-dependent value (e.g., 45° [11]), which could be considered another translation parameter. Concerning the scale feature, a change of direction is identified when a growing object starts shrinking, or vice versa.

Combining aspects from both approaches, it was decided that translations, orientation changes (and pivot rotations), and scales would utilize the following parameters:

1. **Delta threshold (δ)** — Based on local parameters, a *delta threshold* is a value that triggers an event when the absolute difference between feature values (e.g., translation vector magnitude) of two consecutive timestamps is greater than the delta threshold.
2. **Directed cumulative threshold (Δ_{dir})** — Branching from global parameters, a *directed cumulative threshold* is a value that triggers an event when its internal accumulator value surpasses the directed cumulative threshold. The internal accumulator is incremented with each timestamp with the difference in feature value from the last timestamp, only if the feature direction does not change in relation to the previous timestamp which would otherwise reset its value.
3. **Absolute cumulative threshold (Δ_{abs})** — The *absolute cumulative threshold* uses the same logic as the directed variant, differentiating in the way the internal accumulator treats values from new timestamps. Instead of incrementing with the difference in feature value from the last timestamp, it increments with the absolute value of that difference and completely disregarding changes of direction for the given spatiotemporal feature.

To define if these types of spatiotemporal events are of interest, the user would specify any of the three parameters while all of them being optional and the absence of every parameter meaning the object change itself is not of interest to the user.

However, not all spatiotemporal features can be parameterized by the aforementioned metrics. Others, like existential changes (e.g., appearances, disappearances, splits or merges) are inherently prone to a simple binary state thus mapping to a Boolean flag (i.e., the object either suffered a split or it did not). Therefore to specify if such events are of interest to the user a Boolean switch would suffice. On the other hand, features related to the temporal aspect, such as *immutability*, can be parameterized through a minimum time period (e.g., minimum object immutability period to mark an immutability event).

Since this work focuses on visualization and since the quantification of spatiotemporal transformations is out of the scope of this work, it is recommended to read the work of Carneiro [11] for more details on parameterization as the definition of these parameters was the result of a collaboration with the author.

4.3 Visual elements

This section decomposes the static visual narrative introduced in Section 4.1 into its visual elements and justifies the decisions behind their existence and design.

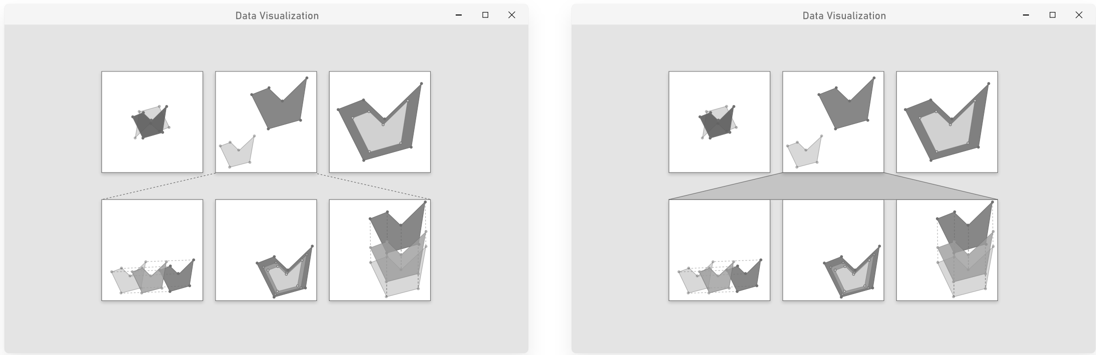
4.3.1 Layout

The visual narrative is organized as storyboards. There is a *main* storyboard, as shown in Figure 4.1, that considers the entire time interval of the dataset and displays all the object changes given the initial set of parameters. Any storyboard is composed of a set of static frames, where each frame visually represents an object transformation (or multiple overlapping transformations).

By clicking on any frame, a new storyboard is displayed beneath it, revealing which transformations occurred for the time interval of that frame, considering more granular parameters which should provide more fine-grained visual results. With these parameters, it may be possible to get more frames since these fine-grained parameters are easier to be reached thus revealing more granular transformations. The term that refers to this process is *zooming in on a frame*. This part of the layout is, therefore, hierarchical since it is possible to gradually zoom in on frames from storyboards that were also created with this form of interaction. Figure 4.2a documents an early version of the hierarchical layout with the use of dashed lines, which were replaced with a shadow area variant (Figure 4.2b) since it was too imperceptible.

Navigation across the narrative is also linked to the layout decisions. Since the size of the user's screen limits the number of frames that can be seen simultaneously, two arrows exist to traverse left (past, regarding the current temporal focus of interest) and right (future, regarding the current temporal focus of interest) through a storyboard. It is also possible to skip to a given frame by clicking on its corresponding item in the timeline, as explained in more detail in Section 4.3.3.

A more complex variant of this layout features the concept of Levels-of-Detail (LODs). Essentially, the storyboards in the hierarchy are gradually assigned lower LODs the closer they are from the root storyboard, providing a chain of Overview+Detail (see Section 2.3.3.3 for more information). In contrast, the storyboard further from the root should be displayed with the most



(a) Early hierarchy layout design with dashed lines (b) Hierarchy layout design with shadow area

Figure 4.2: Visual element for hierarchical layout

Detail. Figure 4.3 shows an early conceptual design of a hierarchical design using levels-of-detail. With this design, the main storyboard exclusively shows the first letter of the word of the transformation that occurs to the object. The second storyboard represents the transformation with the first and final states of the object. Finally, the third storyboard — the one which is currently focused — represents the transformation with the appropriate visual metaphors (e.g., onion skinning and vertex mapping). Nevertheless, this enhancement was not implemented in the prototype as it was impossible to conclude in the available time frame.

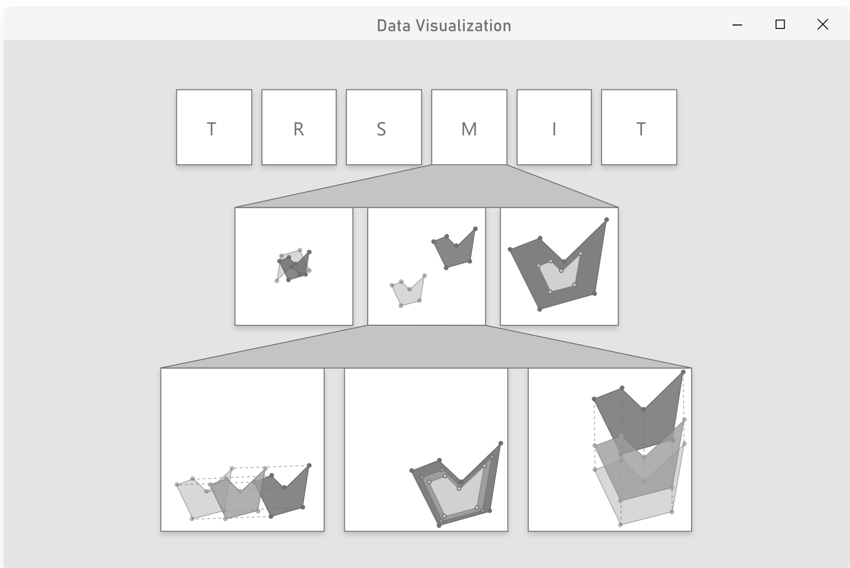


Figure 4.3: Design of a hierarchical layout with levels-of-detail

4.3.2 Transformations visual metaphors

The spatiotemporal domain encompasses phenomena from various areas. Therefore, the list of phenomena transformations that might be of interest is extensive. As hinted in Section 3.3, the scope of supported phenomenon transformations is limited to the spatial aspect of physical transformations and, more specifically, to geometric transformations. However, it is envisioned that the visual summarizing approach introduced by this work can be applied to other transformations, such as changes in velocity, cyclic events, and temperature variations.

The following subsections describe the visual metaphors used for each spatiotemporal feature addressed in the current work.

4.3.2.1 Translation

The visual metaphor used for translations is shown in Figure 4.4. Firstly, it implements onion skinning to denote acceleration, as seen in Section 2.3.3.2. This instance of onion skinning utilizes transparency to distinguish between states: older states have diminished visibility while more recent states are displayed with increased opacity. Secondly, to complement the visual metaphor, a translation vector is displayed to convey magnitude and direction. Lastly, vertex mapping (also explored in Section 2.3.3.2) is used to denote the translation of each vertex by mapping it from the initial state to the final state. Similarly to the proposed methods reviewed in the related work, this technique utilizes dashed lines to decrease its visual impact and relative importance towards the solid lines that represent the edges of the object. The translation vector and vertex mapping visual cues are components of the visual metaphor because it is predicted the possibility of the transformations containing noise (e.g., deformations to the object's vertices from the initial to final states). Without these, in such a case of noise, viewers might experience difficulty in creating a mental image of the transformation.

4.3.2.2 Orientation

The proposed visual metaphor for orientation changes — rotations around the object's centroid — is shown in Figure 4.5. To avoid visual clutter, onion skinning is not used as it is foreseen to cause major object overlapping in orientation changes. To convey the direction (clockwise or counter-clockwise) and orientation change angle, a circular path arrow is drawn. The other visual variable of this metaphor is vertex mapping. In contrast to the vertex mapping explored in translations, this instance does not utilize straight lines. Instead, as orientation changes form a circular path, circular vertex mapping is used to map the vertices from the initial to the final state of the object.

4.3.2.3 Scale

The visual metaphor for scales is shown in Figure 4.6. Scales, similarly to translations, also use the concept of onion skinning as a technique that displays intermediate states. However, overlapping is once more a concern in scales. To circumvent this possibility, the visual variable that changes in

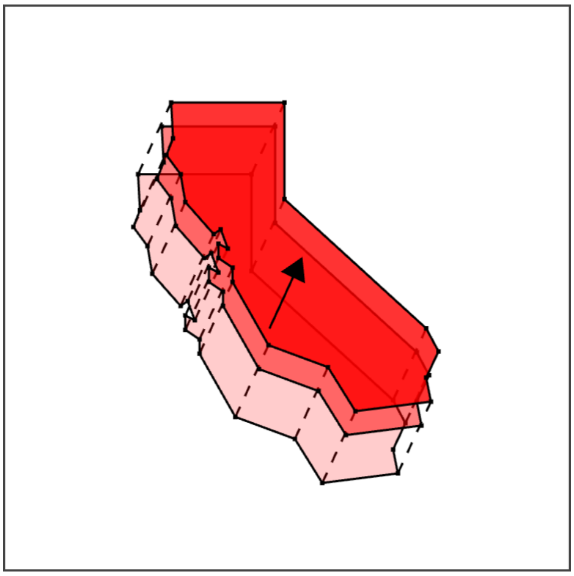


Figure 4.4: Visual metaphor for a translation

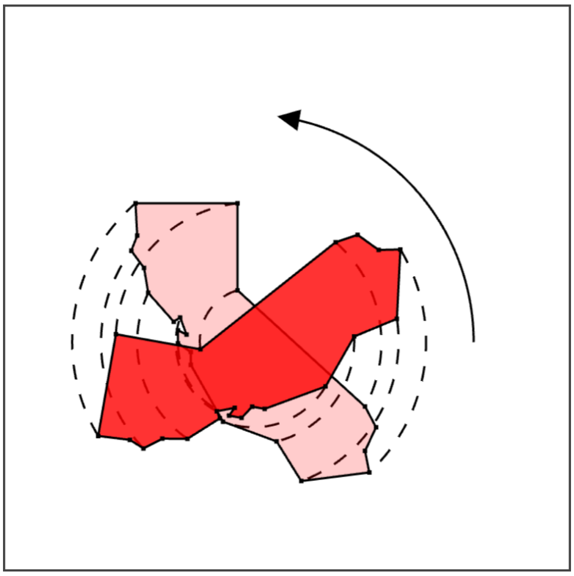


Figure 4.5: Visual metaphor for an orientation change

the sequence of states is the colour saturation instead of transparency. In other words, older states are drawn with decreased saturation while more recent states are rendered with higher saturation values. With inspiration from isochrone maps (see Section 2.3.4.2), the states are drawn in order of dimension: smaller instances of the object are drawn on top while larger areas are drawn from the back. This design allows for overlapping while still representing the increments or decrements of each state in terms of object dimension. Additionally, vertex mapping is merged with the scale vector of the transformation to form an auxiliary visual cue with a similar purpose to the previous transformations.

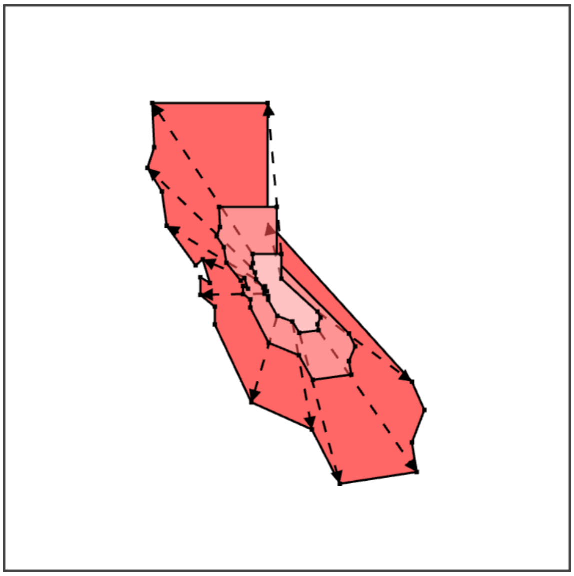


Figure 4.6: Visual metaphor for a scale

4.3.2.4 Rotation

The visual metaphor for rotations (around a pivot) is shown in Figure 4.7. Similarly to translations, onion skinning with transparency is used to denote acceleration. Given that rotations, like orientation changes, form a circular path, its vertex mapping visual cue also conveys this circularity of the transformation. Unique to a pivot rotation is the representation of the pivot point and angle of rotation.

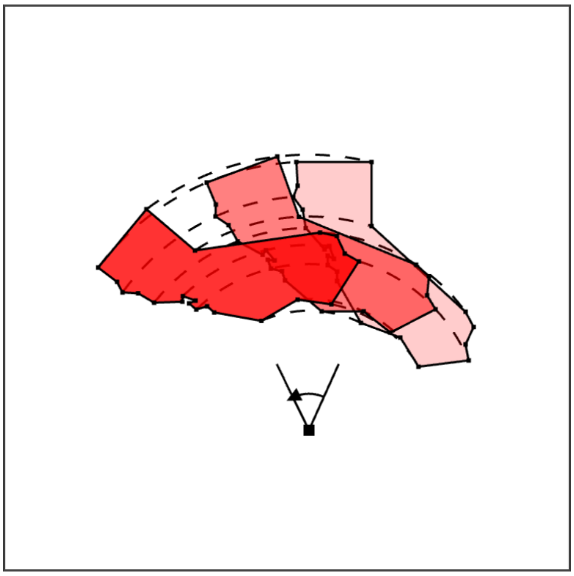


Figure 4.7: Visual metaphor for a rotation

4.3.2.5 Immutability

Immutability represents a period of time where no substantial transformations occurred to the object. Likewise, its visual metaphor, shown in Figure 4.8, represents this with the absence of colour saturation.

Figure 4.8: Visual metaphor for immutability periods

4.3.2.6 Multiple transformations

Multiple transformations can occur to an object, overlapping for a period of time. However, defining which visual aids to use could result in combinatorial explosion as more transformations (and visual cues) are supported. Therefore, given the scope of this work, the visual metaphor for multiple transformations, shown in Figure 4.9, consists solely of onion skinning using transparency.

4.3.2.7 Unknown transformations

The currently supported transformations may not fully explain the story of an object. Therefore, a fallback visual metaphor, shown in Figure 4.10, should be used in cases where there is no support for the transformation, either from the visualization framework or the transformation identification server. Similarly to an immutability period, this is represented with the absence of colour saturation. Conversely, since it is implied there were transformations that occurred to the object, onion skinning is used to represent intermediate states.

4.3.3 Temporal context

With the intent of providing temporal context, a *Timeline Object* was designed for the conceptual solution. The timeline object, shown in Figure 4.11, is composed of two distinct elements: a

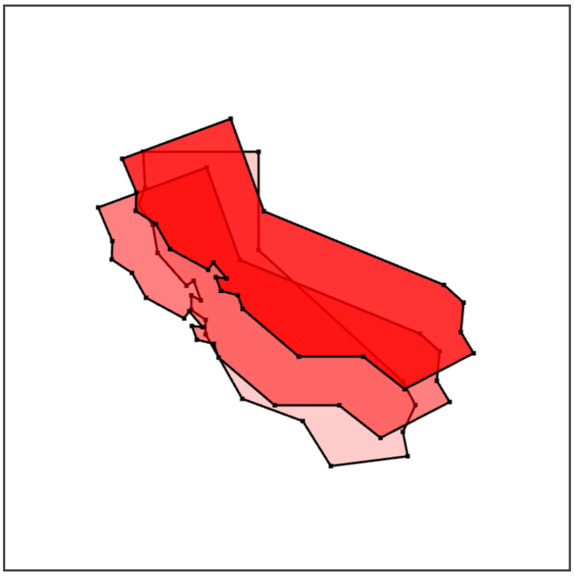


Figure 4.9: Visual metaphor for multiple transformations

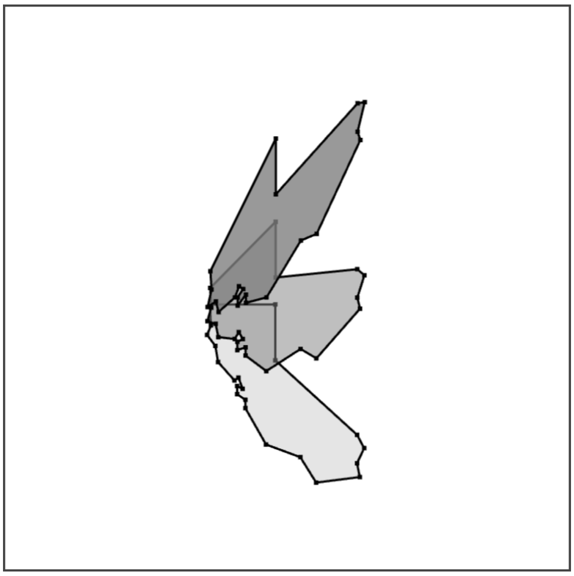


Figure 4.10: Visual metaphor for unknown transformations

narrative progress bar and a timeline. Since each storyboard can have a different number of frames and correspond to different time periods, a timeline object should be built per storyboard.

The narrative progress bar aids the user in contextualizing the frames being presented to the screen in relation to the complete narrative. It is composed of rectangles — progress items —, each representing a frame in the storyboard. In the example shown in Figure 4.11, the first three frames are being displayed to the user (represented by the blue colour) from a total of six (with the other three being greyed-out as they are not currently visible). The other primary responsibility of this element is to provide relative temporal context. The term *relative* refers to representing the duration of each frame *in relation to* the duration of the narrative. In the example, a user is expected to perceive that the fifth frame is the longest and that the fourth frame represents a time period around twice as long as the second frame. In addition, some progress items can be highlighted in certain situations:

1. A dark blue indicates the corresponding frame has been zoomed-in. A particular case exists where a zoomed-in frame may go out of scope, due to player navigation in the same storyboard, making the frame not visible. In those cases, the progress item displays a dark grey indicating that the frame has been zoomed-in, but is not currently visible.
2. Another shade of blue appears on the progress item that corresponds to the frame that the mouse pointer is hovering. The colour is slightly lighter than the aforementioned highlight but still darker than the normal visible frame colour. This feature eases the correspondence between what frame the user is analysing and its progress item in the timeline. Otherwise, the user would be required to manually find the respective progress item.

The timeline element temporally contextualizes the narrative frames in absolute terms, in contrast to the narrative progress bar which does so in a relative manner. Having both elements juxtaposed, the user can assert when a frame starts and ends, and, consequently, when the story begins and finishes. The timeline should also infer the most appropriate units for the timeline. In the example shown in Figure 4.11, the story has an approximate duration of 9 months, so the timeline uses *months* as the time unit.

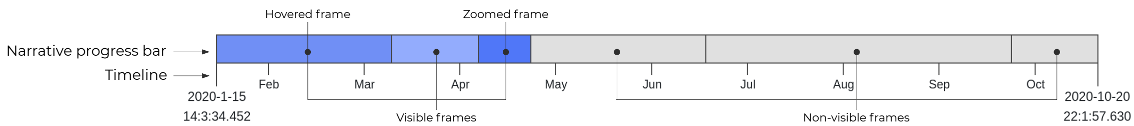


Figure 4.11: Timeline object with legend

4.3.4 Spatial context

As stated in Section 2.3.3.3, providing spatial context is essential in spatiotemporal visualization so users can spatially reference objects in space. Despite base maps being the ubiquitous choice for spatial contextualization in geovisualization, spatiotemporal visualization is used in various other areas. Therefore, to avoid tight coupling with geographical data, the envisioned solution utilizes a grid system to spatially contextualize the overlaid objects.

The grid is displayed behind the object and contains reference values agnostic to any unit of length. Additionally, as shown in Figure 4.12, the grid divisions are not constant throughout the storyboard to reflect scene dimension changes from one frame to another.

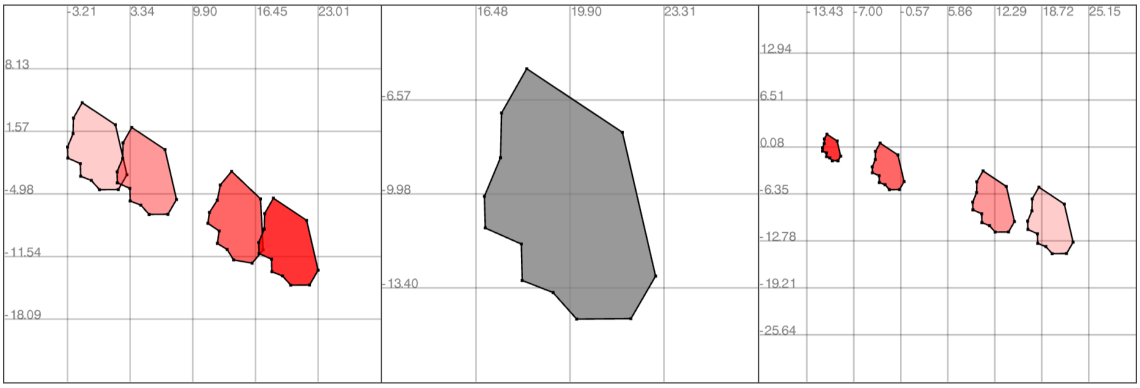


Figure 4.12: Relative grid divisions for different scene dimensions

4.3.5 Supplementary elements

Besides the previously discussed elements considered to be part of the visualization solution, there is another supplementary informative element: the overlay. The overlay, present in each frame, displays the frame's time interval and describes the transformation that occurred to the object and its units through a textual representation. The purpose of this element is to fulfil exact-value information needs of scientific visualization, such as "What was the angle of rotation of this orientation change?". Figure 4.13 shows the overlays of a given narrative. Analysing the overlay alone, a user can, for example, identify that in the third frame of the narrative multiple transformations occurred and, more specifically, a translation with vector $(-30.00, 10.00)$ and a uniform scale of 0.40 units.

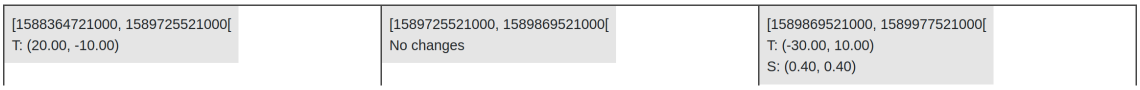


Figure 4.13: Frame overlays

4.4 Framework pipeline

A conceptual framework design is considered to be mandatory to orchestrate the elements discussed in the previous sections. The framework should take the spatiotemporal data as input and, through a series of transformations, output the visual narrative as presented in the current chapter. As seen in Section 2.3.2, Haber and McNabb have proposed a conceptual pipeline for computer-assisted visualizations. In this section, the authors' pipeline is adapted and extended to accommodate a solution based on static visual narratives. Figure 4.14 visually represents the framework's pipeline design.

As a starting point, the raw data must be input into the system by requesting the data from a spatiotemporal phenomena change representation and quantification server [11]. This requires a data schema and parameters to be defined. An example of a possible data format can be seen in the *Raw Data* stage in Figure 4.14 where the spatiotemporal phenomena are represented as a set of *object changes*, introduced in Section 2.1.2. For example, given a set of parameters, a translation was identified in the time interval from t_5 to t_{10} with a translation vector of $(-1,2)$.

The raw data is then pre-processed in the *Data Enrichment / Enhancement* transformation which essentially converts the data into the application's data structure and adjusting minor aspects. In the example, degrees are converted to radians and uniform scales — scales with equal units in all dimensions — are converted to general scales. The resulting data of this transformation stage is called *Derived Data*. This stage commonly encompasses filtering the data. However, in the case of the proposed conceptual framework, it is intended for filtering to be done alongside the quantification of object transformations (external to this framework). Otherwise, the resulting narrative could have gaps in time as frames might be removed altogether through filtering.

The *Visualization Mapping* transformation takes the derived data and defines an *Abstract Visualization Object*, as explained in Section 2.3.2. The abstract visualization object is, in essence, a visual metaphor for each object transformation, utilizing the visual variables introduced in Section 2.3.3. In this example, a translation is visually mapped to a visual metaphor that utilizes vertex mapping to denote the translation vector, and transparency to differentiate between the initial and final states.

Finally, these abstract representations are rendered, forming visible content — *Displayable Image*. The rendering stage encompasses computer graphics techniques, like vertex specification, model-view-perspective matrix calculation, and shading. These techniques are part of the rendering pipeline of a common WebGL application. Nonetheless, it is common to use a framework to abstract the low-level intricacies of WebGL, such as Three.js.

In parallel, interaction events by the user manipulate the various transformation stages. The user is able to select which dataset to visualize and adjust the parameters of each spatiotemporal feature. A representation and quantification server is therefore responsible for, given the dataset and parameters, returning the identified object changes. Users can also influence how to give visual meaning to an object change by enabling, disabling or adjusting the visual variables of a visual metaphor for an object change (e.g., disabling vertex mapping in the example shown in Figure 4.14). Finally, in the *Displayable Image*, the user should be able to navigate through the timeline in order to explore the different frames (and object changes) across time.

4.5 Summary

Towards solving the problem of efficiently understanding the evolution of spatiotemporal phenomena, this chapter presented a conceptual solution based on static visual narratives. Throughout the description, it was reiterated the importance of interactivity as a means for the user to freely explore the data. To that end, various parameterization approaches were documented for different spatiotemporal features to enable the concept of events of interest. Having set the parameters to retrieve the data, the focus shifted to its visual representation and thus the visual elements of the proposed visualization were presented, discussing the visual metaphors of some spatiotemporal transformations, spatial and temporal contextualization, and layout. Lastly, a conceptual framework pipeline was presented, tailored to the envisioned solution, describing each stage and interaction events.

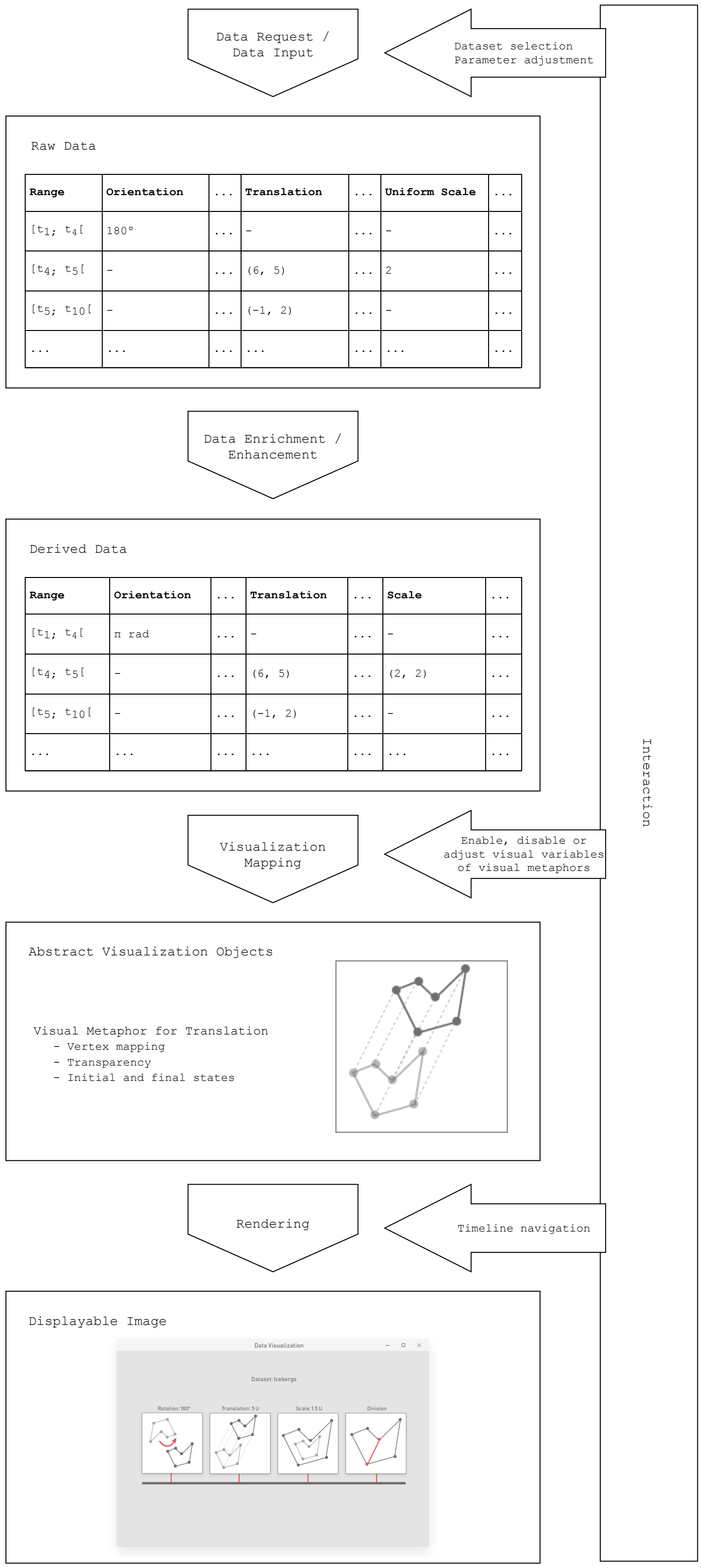


Figure 4.14: Framework pipeline

Chapter 5

Visualization Prototype

This chapter documents the implementation details of the developed visualization prototype, based on the conceptual solution proposed in the previous chapter. Section 5.1 contextualizes the system in which the application lies and its environment. Section 5.2 focuses on the logic of the developed application, its architecture, flow and implementation-specific details regarding spatiotemporal features, parameterization and visual elements. Lastly, Section 5.3 describes the technology stack used to develop the prototype.

5.1 System architecture

As hinted before, the prototype system developed in the current work is a component of a larger system. This overall system, shown in Figure 5.1, consists of two elements: a server and a Web application. The server represents a computational unit that stores and processes spatiotemporal data, quantifies spatiotemporal changes and organizes those into events of interest, according to user-defined criteria. The scope of this work is exclusively bounded by the Web application component which is a prototype of the conceptual solution proposed in Chapter 4. However, a communication interface was defined and agreed in collaboration with Carneiro (the author of the prototype server part of this system [11]). Nonetheless, these components are not tightly coupled, meaning any of the two can be replaced in the future for another upgraded component that further extends the work, as long as it implements the same communication interface. For more details regarding the communication interface, see Appendix A.1.

5.2 Client application

This section focuses on the Web application component of the system — the developed prototype —, illustrating its architecture and implementation details.

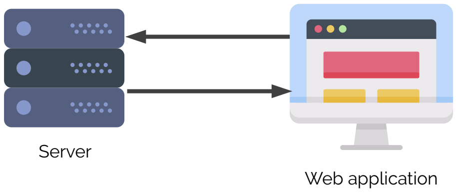


Figure 5.1: System architecture

5.2.1 Architecture and flow

The architecture of the client application loosely follows the Model-View-Controller (MVC) software architectural pattern as shown in Figure 5.2. The difference between the developed architecture and the traditional MVC pattern is the update bypass from the Controller to the View, enabling updating some aspects of the view that are not related to the model. For the remainder of this subsection, each module is detailed along with its interactions.

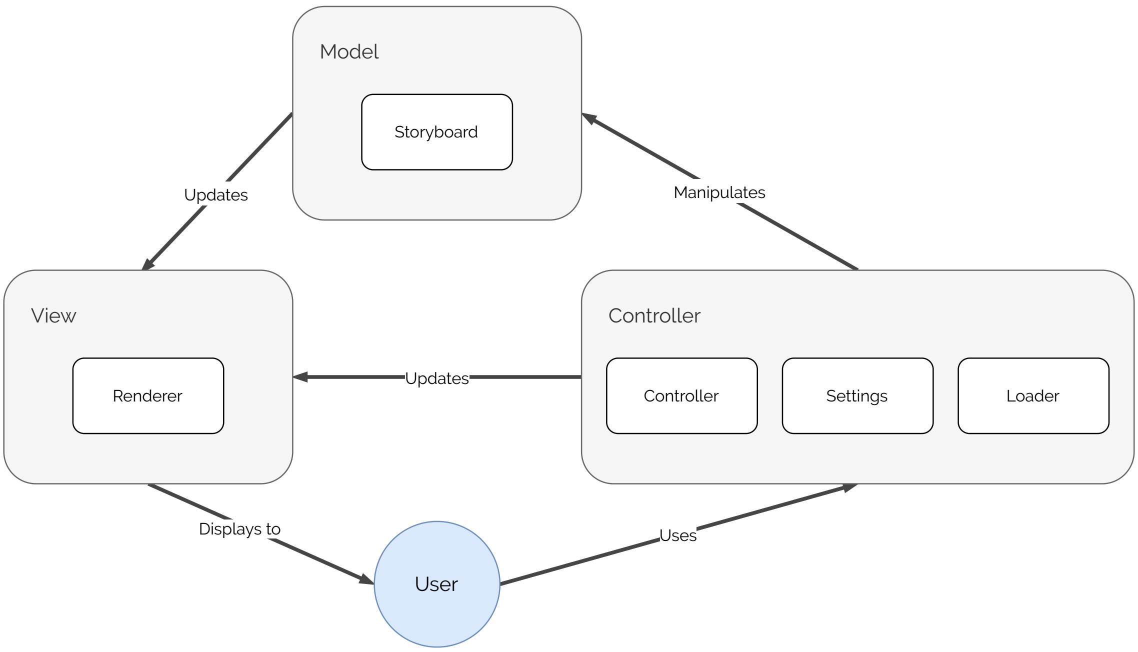


Figure 5.2: Application architecture

Following the flow of a regular usage of the visualization tool, the user would start by inputting the dataset key and the parameters for each spatiotemporal feature of interest, as well as define any other visual preferences. These interactions are all handled by the *Settings* manager (a controller). In this manager, each setting is part of one or more groups, each having a handler that reacts to changes in value(s). The settings manager ensures that each settings group handler is only invoked once, even when multiple settings with overlapping groups are simultaneously changed by the user. The *events* group deals with settings that change how events of interest are identified (e.g., the delta threshold of a translation) and its handler essentially requests a new storyboard from the server with the updated spatiotemporal feature parameters. The *scenes* is responsible for rendering a new scene, supported by Three.js, when, for example, elements of the visual metaphors are changed (e.g., disabling the grid). Finally, the *layout* group handles all changes to the layout by re-writing the page's content (e.g., disabling the overlay).

Whenever a storyboard is requested, either by changing the dataset or its spatiotemporal feature parameters, the *Loader* module handles the necessary logic. The loader reads the settings' values and sends the request to the server. The response is then parsed by the loader which then sets the new storyboard accordingly. These actions correspond to the first and second stages of the pipeline constructed in Section 4.4.

The storyboard keeps the data concerning all the frames retrieved from a given dataset and set of (initial) parameters. The storyboard's main data structure is a tree, as illustrated in Figure 5.3. The storyboard is initially loaded with all the frames, unbound in time (retrieving the full narrative from start to end), for the user-submitted dataset and parameters. This process results in *Row 0* of the storyboard (red row in Figure 5.3), hereby referred to as the *main storyboard*. As stated in Chapter 4, clicking on a frame — zooming-in — results in a request for more granular transformations for the time interval that frame represents. This logic is handled by the *Controller* and *Loader*, sending a request to the server for frames limited to the initial and final timestamps of the clicked frame and adjusted spatiotemporal feature parameters (see Section 5.2.3 for details on this adjustment). On receiving the response of this request, the controller sets the returned frames as the child frames of the frame that was clicked, forming a tree data structure. These child frames are cached so that subsequent zoom-in interactions on the same frame do not require a request to the server.

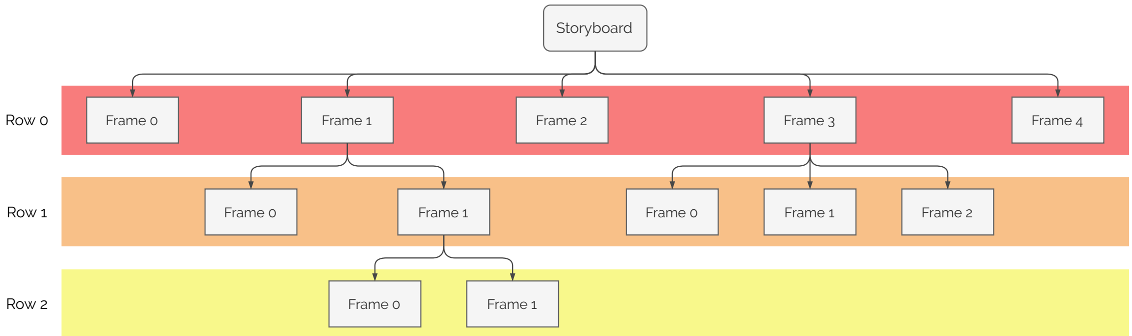


Figure 5.3: Example of a storyboard

Figure 5.4 presents the most important aspects of the Model component of the MVC architecture through a UML Diagram. Each *Frame* stores its time interval under the form of two timestamps and a reference to an *Object* and to a *Transformation*. The object is essentially a set of *ObjectStates* which represent the geometry (vertices list) of the spatiotemporal phenomenon at a certain timestamp within the frame's time interval. The *Transformation* is abstract and conceptually establishes a relationship of inheritance with each of the concrete transformations (e.g., rotation) and enables extracting common logic across transformations into the parent class. Furthermore, it allows supporting more transformations in the future without drastically changing the remaining architecture. The set of implemented transformations in the prototype are the ones presented in Chapter 4 with the addition of an *Unimportant* transformation (explained in Section 5.2.2). It is the responsibility of each transformation to populate the frame's graphics scene with the appropriate visual metaphors, seen in Section 4.3.2. Therefore, each transformation logic encompasses a distinct visual metaphor and therefore corresponds to the third stage of the visualization pipeline — visualization mapping, detailed in Section 4.4.

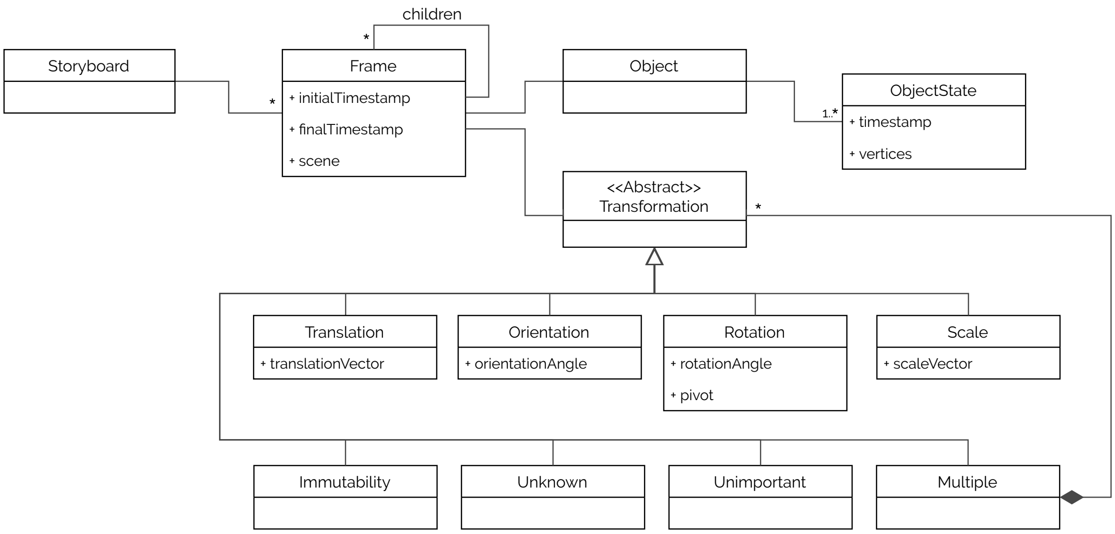


Figure 5.4: UML diagram of the Model component

After loading frames and populating their dependent objects, the *Renderer* renders each currently visible frame's scene to the screen, concluding the visualization pipeline presented in Section 4.4. An important detail of the prototype is that it uses a full-screen canvas to render all the visible frames, instead of a smaller canvas per frame. This is due to browsers limiting the number of available WebGL contexts (16 in Google Chrome, at the time of writing). Furthermore, it can be a memory inefficient solution as more resources need to be allocated per WebGL context. The renderer, when notified by any controller component, renders updates to reflect changes that happen due to user interaction. The user can navigate through the visual narrative using the arrows (sequentially) or by clicking on an element of the timeline (jumping to the selected frame). These navigation methods are handled by the *Controller* which notifies the *Renderer* so that it can update which frames are visible. The same logic also applies to zooming-in on a frame which, upon being notified, the *Renderer* updates the HTML content.

5.2.2 Supported transformations

The spatiotemporal transformations that were implemented are the ones presented in Section 4.3.2 (translation, orientation, scale, rotation, immutability, multiple transformations, unknown transformations), with two exceptions. First, even though the visual metaphor for pivot rotations was implemented, the current version of the communication interface does not include a schema for this spatiotemporal transformation since the prototype server part of this system does not support pivot rotations.

On the other hand, another type of frame was devised throughout prototype development and testing: *Unimportant* changes, meaning the lack of transformations encompassed by the features of interest. Before the introduction of Unimportant changes, "time gaps" existed between frames, as illustrated in Figure 5.5, leading to temporal gaps in the narrative. In the example of Figure 5.5, a scale from 0 ms to 30 ms and a translation from 40 ms to 70 ms were identified, leaving out 10 ms of the narrative (from 30 ms to 40 ms). The display of full temporal continuum was considered relevant to avoid lack of interpretation of what happened in-between some frames. The frames that fill in those gaps are referred to as *Unimportant* changes, having the same visual metaphor as an *Immutability*. However, it should not be considered an immutability because some changes might have occurred to the object, while not surpassing any of the established features of interest parameters.

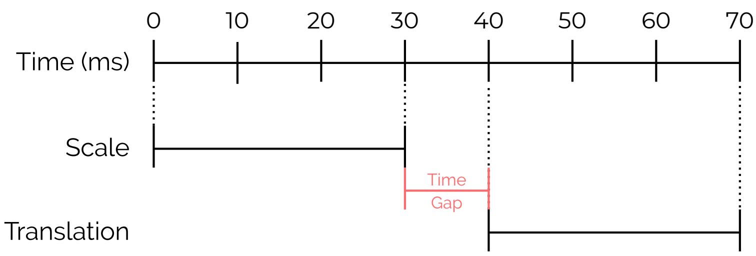


Figure 5.5: Example of a scenario where a "time gap" occurs

5.2.3 Parameterization

The prototype supports parameterization for the four primary spatiotemporal features — translation, orientation, scale, and immutability. As stated in Section 4.2, the first three utilize three parameters — delta threshold, directed cumulative threshold, and absolute cumulative threshold, while immutability is measured with a time-based parameter, named *Time threshold*. The full list of spatiotemporal feature parameters is shown in Figure 5.6. The list reveals two new types of parameters that were brought to light during tests with the prototype: *Noise epsilon* and *Detail threshold multiplier*.

Noise epsilon is a parameter that was defined out of necessity when testing the prototype connected to the change representation and quantification server [11]. In those tests, even when supplied with a fully immutable object, the server would respond with some form of transformation with very small values (e.g., orientation change of 0.0001°). Carneiro believes this issue is related to the selected algorithm for point set registration — the technique used by the author for spatiotemporal change identification. In order to overcome this noise, a noise epsilon value was

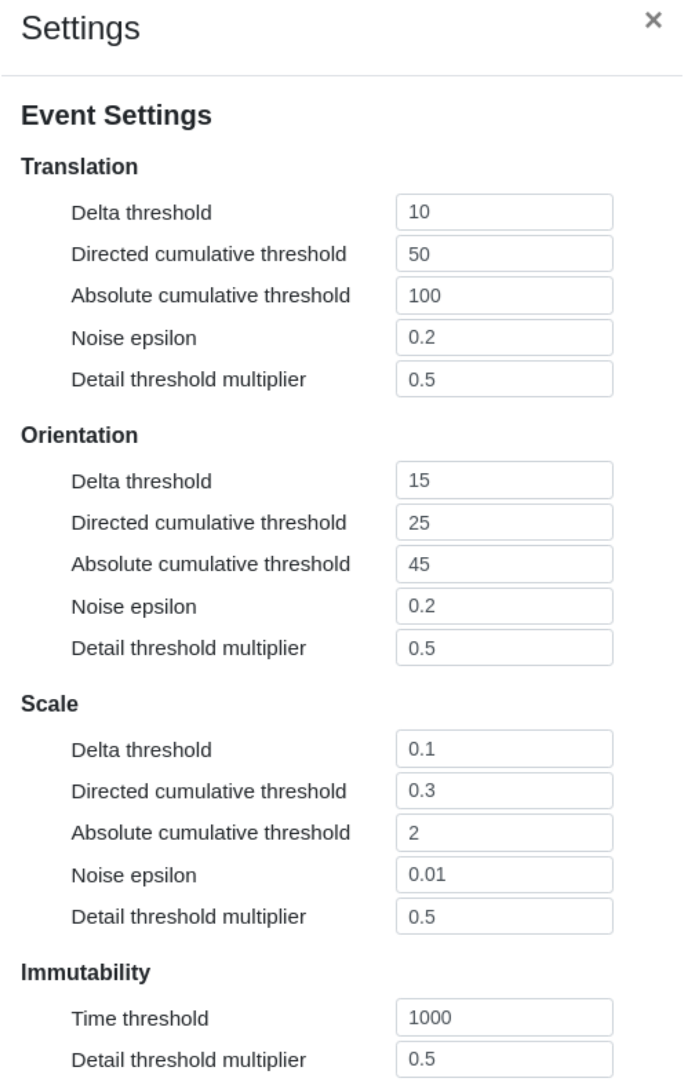


Figure 5.6: Parameter settings

added to the parameters of translations, orientations, and scales to mitigate the aforementioned sensitivity. The actual implementation of noise filtering is, as with the other parameters, dependent on the server component. A possible solution is to only start considering transformations when its noise epsilon is reached.

Section 4.3.1 describes how a user can zoom-in on a frame to see more granular transformations. It was hinted that the process that made this possible was based on lowering the parameter values so that, possibly, smaller changes are detected and more events are marked. However, it was not yet presented how those parameter values are adjusted. A simple solution would be to subtract a constant value to the parameters. However, doing so multiple enough times would result in the parameters reaching a value of zero which instructs the server to identify a transformation regardless of its units. The proposed solution is to have a function with a horizontal asymptote along the x-axis: $y = p \cdot m^x$, where x is the depth in the storyboard, y is the parameter value to be used to retrieve the frames, p is the initial parameter set by the user, and m is the detail threshold multiplier. Assuming a multiplier value in the interval $]0, 1[$, this approach prevents the parameters from ever reaching zero while decreasing on each depth. The depth of the storyboard corresponds to the rows shown in Figure 5.3 since it utilizes a tree data structure.

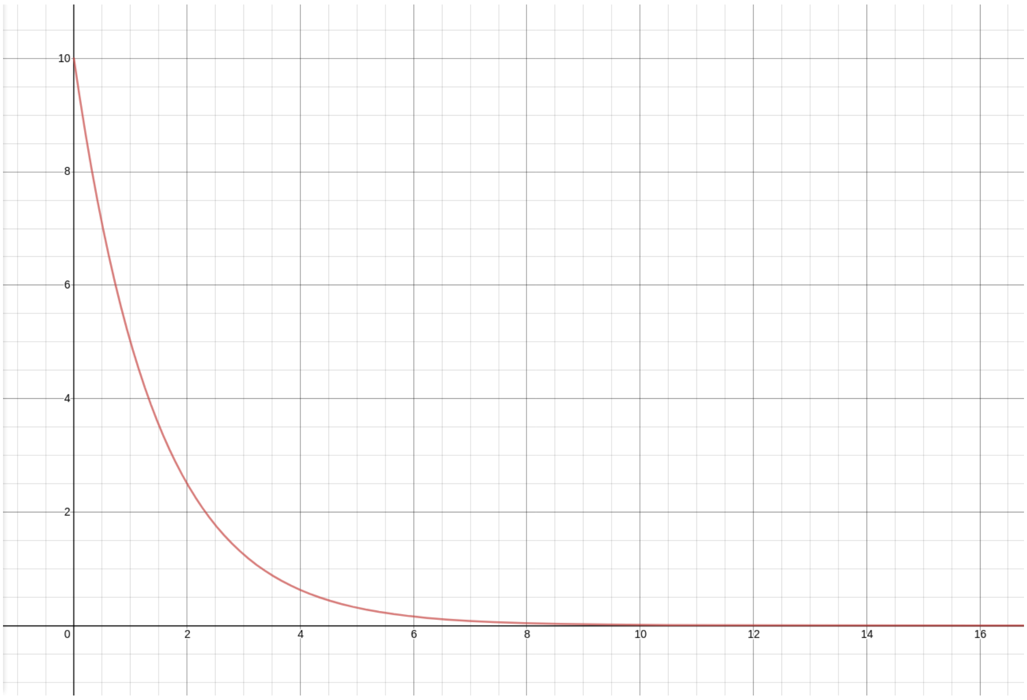


Figure 5.7: Instance of the proposed function for parameter value adjustment where $p = 10$ and $m = 0.5$

To illustrate the zooming feature, consider the example in Figure 5.8. Assume a multiplier of 0.5 for both transformations, and the only other parameters to be directed cumulative thresholds (20 units for translation, and 10º for orientation changes). The expected result is a main storyboard (row 0) with only one frame representing the translation. However, if clicked, the directed cumulative threshold for orientation becomes 5º ($y = 10 \cdot 0.5^1 = 5$), meaning the new row (row 1) of the storyboard will display both transformations (translation and orientation changes).

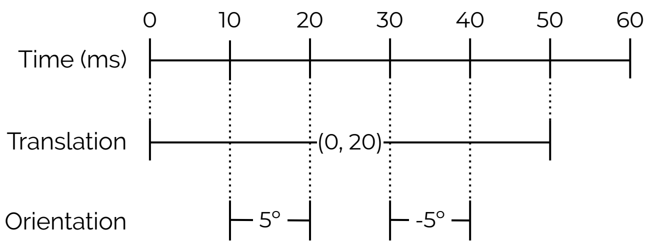


Figure 5.8: Zoom example phenomenon

5.2.4 Visualization customization

As mentioned in Section 4.4, user interaction can affect the visual mapping stage of the visualization pipeline. The developed prototype allows users to toggle or modify some elements of the visual metaphors or page layout, as shown in Figure 5.9. The majority of these settings simply enable or disable elements that were discussed in Chapter 4. The major additions to those were the polygon simplification feature and the limiting of intermediate (onion skinning) states, both addressing the problem of visual clutter.

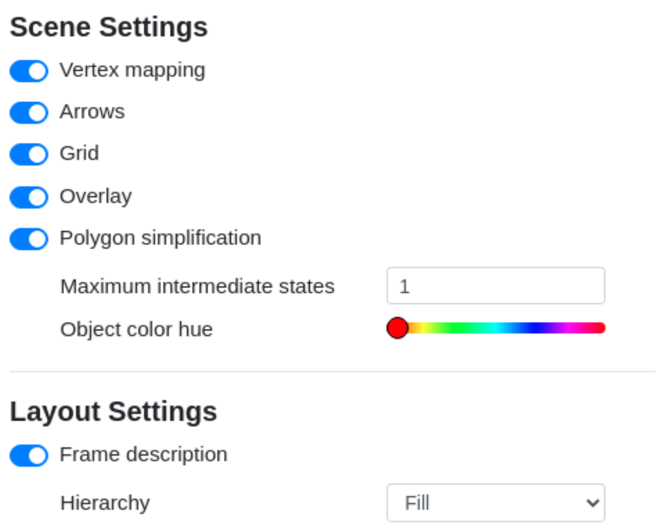
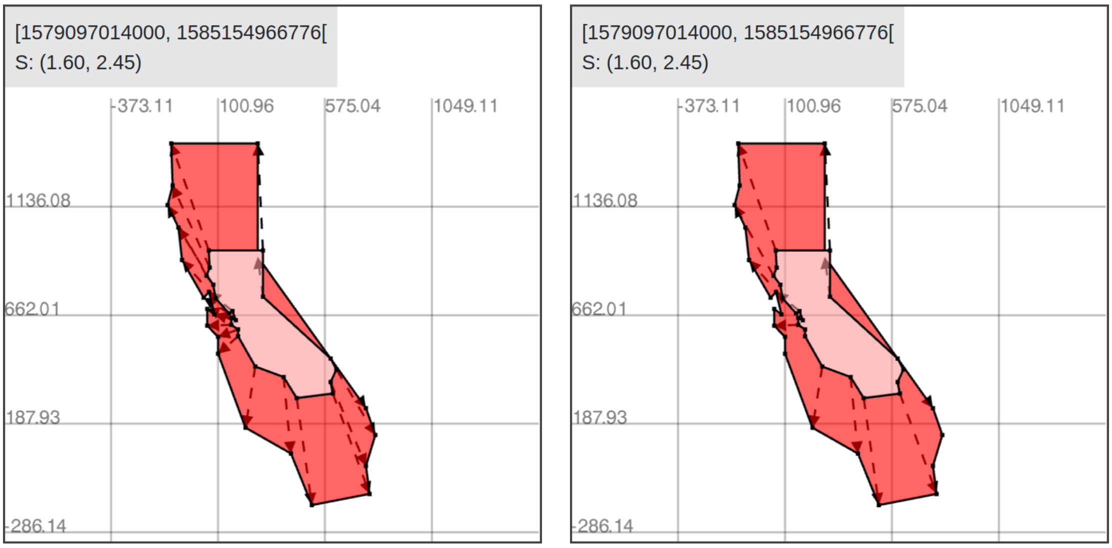


Figure 5.9: Customization settings

As predicted beforehand in Section 2.4.3, visual clutter is a problem for objects which have sections with a large density of vertices, as illustrated in Figure 5.10a. Through polygon simplification, it was possible to mark which vertices would be used for the visual metaphors. The result of applying polygon simplification is shown in Figure 5.10b. Analysing the left image, it is possible to see that the left side of the object featured larger vertex density resulting in multiple arrows overlapping each other and causing visual clutter. Looking at the right image, some of the visual clutter was mitigated. At an implementation level, the polygon simplification process utilizes a sequence of two algorithms: Radial Distance and Ramer-Douglas-Peucker. This combination of algorithms was based on the work of Vladimir Agafonkin[1], and leverages the performance benefit of a Radial Distance pre-pass and the finer-grained results of Ramer-Douglas-Peucker for the remaining vertices.

Another issue regarding visual clutter was discovered when testing with datasets which resulted in frames with numerous object states. In its most simple approach, onion skinning draws all the available intermediate states, causing visual clutter when multiple states overlap each other

[1] Simplify.js. Tiny high-performance JavaScript polyline simplification library. Available at `http://mourner.github.io/simplify-js/`. Accessed in 2020-06-17.



(a) Before polygon simplification (b) After polygon simplification

Figure 5.10: Same frame rendered without (left) and with (right) polygon simplification

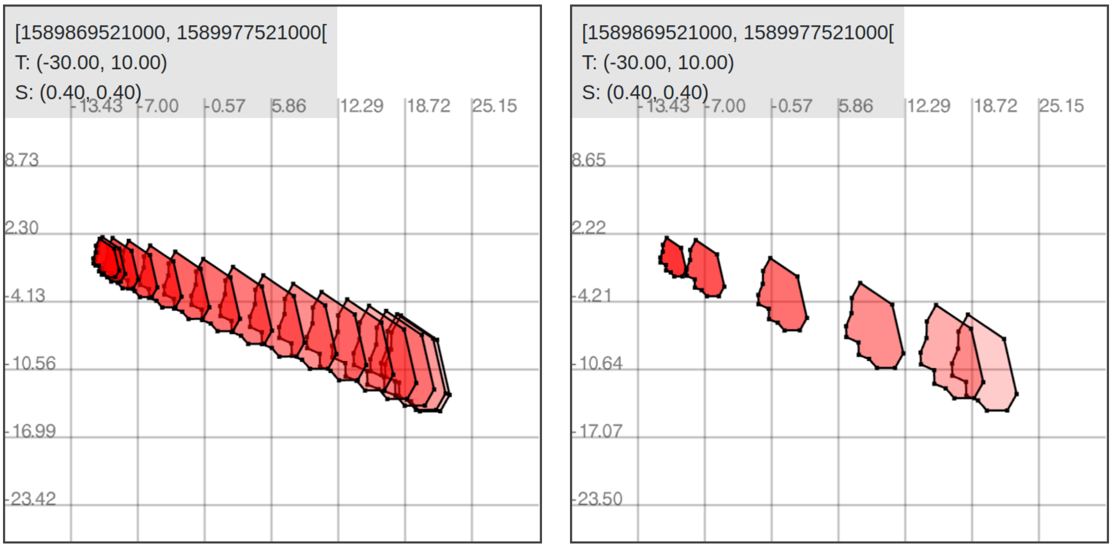
(Figure 5.11a exemplifies such a scenario). To mitigate visual clutter caused by onion skinning, a solution was devised by naively and manually limiting the number of intermediate states. This is far from being an ideal solution, however it can make a positive impact in certain scenarios, as illustrated in Figure 5.11b. In the prototype's settings, the user can set the value of the *Maximum intermediate states*. One of its major limitations comes from two constraints that were imposed on it: all displayed states must be equally spaced in terms of time, and the first and final states must always be displayed. Since it is expected that the states returned by the server to be equally spaced in time, it becomes a problem of retrieving, at most, n equally spaced items of the state array. However, as shown in Figure 5.12, it is not always possible for all combinations of the maximum number of intermediate states and the total number of states.

5.3 Technology stack

According to the conclusions taken from Section 2.4, the prototype that implements the solution introduced in Chapter 4 was developed using HTML (document structure), CSS (presentation, formatting and layout) and JavaScript (application logic) since these are prevalent in Web development. It is, however, common to use abstractions or extensions under the form of other programming languages, alternative syntaxes (e.g., Sassy CSS was used for more structured code), or libraries (e.g., Three.js was used for rendering each frame onto the HTML canvas).

Containerization technologies were also used for the prototype, namely *Docker* and *Docker Compose*[2]. The latter is used to orchestrate Docker containers. In the case of this prototype, it

[2]Docker Compose. Tool for defining and running multi-container Docker applications. Available at https://docs.docker.com/compose/. Accessed in 2020-06-15.



(a) Before intermediate state limitation

(b) After intermediate state limitation ($n = 4$)

Figure 5.11: Same frame rendered without (left) and with (right) intermediate state limitation

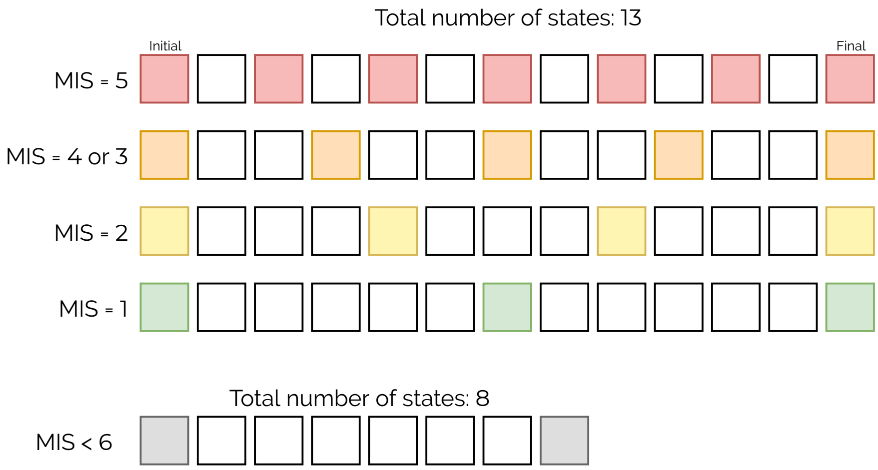


Figure 5.12: Limitations of the approach used in limiting intermediate states. Filled squares indicate those states have been selected for display. MIS standing for maximum intermediate states.

was responsible for setting up a *nginx*[3] server and the Web application's service. The application service lies in a *node*[4]-based environment which enables scripting for building — compiling all JavaScript and SCSS files into one file of each and minifying them for improved loading times — and serving the application. The *node* environment also enables the use of *yarn*[5] for package management, easing the addition of dependencies and scripting.

[3] Nginx. High-performance load balancer, Web server, and reverse proxy. Available at https://www.nginx.com/. Accessed in 2020-06-15.

[4] Node.js. Browserless JavaScript runtime environment. Available at https://nodejs.org/. Accessed in 2020-06-15.

[5] Yarn. Package and project manager. Available at https://yarnpkg.com/. Accessed in 2020-06-15.

5.4 Summary

This chapter presented implementation details about how the prototype interacts with a spatiotemporal feature quantification server while maintaining itself independent from specific implementations, defining a standard for communication. The application-focused section of this chapter discussed the aspects that were not conceptualized in the proposed solution, either because it was implementation-specific or due to the problems discussed only emerging during the actual development process. Amongst these problems, the most relevant to this work were the ones regarding the mitigation of visual clutter caused by numerous overlapping object states and polygons with a high density of vertices. Solutions for these problems were documented, even though they suffer from limitations. In addition, a new type of frame and a noise parameter were added to overcome limitations that can occur at server level and a multiplier parameter was defined to allow the user to customize how the parameters are adjusted when the frames are zoomed-in. Lastly, the technology stack used to develop the prototype was presented, highlighting the importance of cross-platform and containerization technologies.

Chapter 6

Evaluation

This chapter describes, in Section 6.1, the evaluation methodology adopted to assess the effectiveness of the developed visualization framework and its efficiency when compared to dynamic visual narratives. Afterwards, Section 6.2 presents and discusses the results of this assessment. Lastly, Section 6.3 answers the research questions presented in Section 3.4 with the results gathered from the evaluation process.

6.1 Evaluation methodology

To evaluate the effectiveness of the proposed solution, a user study was conducted. The main questions to be answered in the evaluation process were:

1. How is the proposed static visual narrative interpreted?
2. How does the proposed static visual narrative compare to a dynamic visualization approach?

To answer these top-level questions, a questionnaire was elaborated and sent to university computer science students who are more familiar with computer visual literacy. The following sections describe the evaluation artefact — the questionnaire — and the rationale behind its elaboration and structure as well as the questions.

6.1.1 Evaluation artefact

The evaluation process consists of user studies [58]. As reviewed in Section 2.3.5, another possible evaluation method is through heuristic evaluation. Through heuristic evaluation, user experience problems might have been identified and solved before conducting a user study, thus making it a helpful complement to the evaluation process. However, access to a specialist for heuristic evaluation was not feasible due to resource limitations. The user study, initially envisioned to be an in-person interview with hands-on access to the visualization framework, was based on a questionnaire that was elaborated and sent via e-mail due to the COVID-19 pandemic, removing the possibility for in-person interviews. The questionnaire, shown in Appendix B.1, was voluntarily answered by 33 students of the Integrated Master in Informatics and Computing Engineering at the Faculty of Engineering of the University of Porto.

It is important to note that the questionnaire results may be influenced by sampling bias since the subjects are voluntary (voluntary response sampling) and are all enrolled in the same college course (convenience sampling). This sampling method was adopted due to ease of access to subjects and resource limitations since, to reach a more diverse sample, helping guides would have to be developed to accompany the questionnaire for people with less experience with visualization tools. Being a non-probability sampling method, the generalization of results can be hindered [44]. Despite those conditions, the level of experience of the subjects that were sampled is representative of what to be expected from potential users. Hence, it provides important feedback and a first evaluation of the tool, its possible advantages and caveats, that can be used to guide further improvements.

6.1.2 Methodology rationale

The evaluation methodology follows the work of Ying Zhu [58] which, as explained in more detail in Section 2.3.5, proposes measuring visualization effectiveness in terms of accuracy, utility, and efficiency. Because of the medium adopted to conduct the user study, these three principles are not assessed evenly. Accuracy is assessed, for the most part, through the quantitative measurement of counting the number of interpretation errors. Utility can also be assessed with quantitative measurements, such as counting the number of successfully completed tasks. However, the questionnaire medium hinders the ability for test users to ask questions regarding the framework and therefore would impact the completion of these tasks. To overcome this difficulty, some interaction aspects are mocked beforehand and shown as images so the subject can formulate conclusions. Efficiency is being measured both quantitatively and qualitatively. In quantitative terms, it is of interest to know the time necessary to understand the evolution of a spatiotemporal phenomenon through the developed framework. In qualitative terms, it is important to compare the usage of static visual narratives as opposed to their dynamic counterpart. A limitation to this comparison is that the dynamic animation is not mapping the same time interval as the static version since the data represents almost a year which is reduced to around 12 seconds of video. An improved version of this comparison should use a dynamic visual narrative tool to view the data-driven animation with one-to-one temporal mapping and user-adjustable play speed.

6.1.3 Questionnaire

As stated, the artefact of the current evaluation process is a questionnaire. The following subsections describe the questionnaire's structure and questions.

6.1.3.1 Structure

The questionnaire is split up into 6 sections. The first section is an introduction to the work and to the questionnaire itself and contains some sociodemographic questions for statistical purposes.

The second and third sections evaluate the effectiveness of the visual metaphors used to visually illustrate each supported transformation. The difference between the two sections is that the former evaluates qualitative and relative aspects (e.g., what transformation is seen or how the object is accelerating) while the latter evaluates quantitative aspects (e.g., what is the translation vector for a given transformation). These two sections are also introductory to the full visual narratives of the next sections, meaning they are meant for the user to gradually and consistently acquire knowledge of the visualization framework to answer progressively more complex questions.

The fourth and fifth sections evaluate the effectiveness of the static visual narrative as a medium for conveying the evolution of a spatiotemporal phenomenon. The questions in these sections focus on assessing if a user can understand the story of the object through the static visual narrative and how the learning process compares to dynamic visual narratives — an animation, in the case of this questionnaire. The former, as an introduction to complete narratives, evaluates the effectiveness of some elements, such as the timeline. Most importantly, it assesses the interpretation of the object story using the testing prototype (Chapter 5) that instantiates the conceptual framework (Chapter 4). The latter focuses on comparing the visualization tool to a dynamic counterpart. The two sections also have different test scenarios, serving different purposes. The first dataset features a *simpler* story with no overlapping transformations and the parameters are mocked so that all transformations in the dataset form frames. The second dataset has the intent of evaluating the *frame zoom* interaction, thus enabling overlapping transformations and transformations which do not form frames for the initial set of parameters, requiring the user to click on a frame to *zoom in* the details through a set of new frames (controlled by more granular parameters). These details are hidden on the main timeline as the parameters controlling its frames are less granular. As a frame is zoomed for more detail, the parameters are loosened to allow for new frames to be drawn in a new storyboard row below the first, possibly revealing more information. The intent of this caveat is to leverage the ability of the framework to show transformations that might be missed from the animation, as they occur very rapidly or are too small to be recognized by the human visual and perceptual system.

Finally, a sixth section allows for optional observations regarding the questionnaire and the developed framework.

6.1.3.2 Questions

The first section, besides introducing the subject of the current work and questionnaire objective, gathers, non-obligatorily, two sociodemographic details — the subject's age group and gender (Questions 1.1. and 1.2., respectively).

The second section is composed of 10 multiple-choice questions. The first seven questions (2.1. to 2.7.) present one illustrative example of each possible frame type (translation, orientation change, scale, pivot rotation, unknown change, multiple changes, and immutability) and the user must identify, through exclusive selection over multiple-choice, the transformation considered to occur. The purpose of these questions is to evaluate how the visual mapping of each transformation type allows an effective interpretation from the user's point of view. Questions 2.8. and 2.9. assess how multiple states are perceived in terms of chronological order. Both these questions use symbols to refer to each state as numbers or letters induced some subjects to a preconceived order. Question 2.8. presents a frame with multiple transformations which uses transparency for onion skinning. Question 2.9. presents a frame with a scale transformation which uses colour saturation for onion skinning. The latter is challenging in the sense that the transformation itself does not follow an expected pattern of a scale — the object first shrinks from its original size and then enlarges to a size larger than the first state (simulating a frame triggered by an absolute cumulative threshold). Question 2.10. differs from the previous ones since it is a checkbox question, intended for users to select the options that are considered true statements. This question presents two figures side-by-side representing translations with the same initial and final states, but the objects having different accelerations — Figure 1 having constant velocity and Figure 2 having positive acceleration (increasing velocity). The purpose of this question is to assess the ability of onion skinning in conveying velocity and acceleration, at least in relative terms.

The third section features 4 similar questions. These questions (3.1. to 3.4.) present, like the previous section, a frame per question and ask the units of the transformation represented in the frame. In the case of translations and scales, it is asked the value for each coordinate (X and Y), while rotations and orientations inquire the angle and direction (clockwise or counter-clockwise), as well as the pivot point in the case of pivot rotation. The purpose of this question is to understand if subjects are able to infer the units and directions of a transformation, given the overlay and visual cues, such as arrows.

The fourth section is the first to evaluate the visual narrative as a whole and is composed of 6 questions. The first four questions (4.1. to 4.4.) assess the relevance of the timeline, querying the subject on how many frames the visual narrative has in total and, of those, how many are visible and also the duration of the object's story. Question 4.4. differs from the previous in the section since it is a checkbox question with true/false statements regarding the duration of frames. Question 4.5. is critical for the evaluation of this work. It requires the subject to textually and openly describe, given the full visual narrative, the transformations the object has suffered, its units and durations. This question was introduced to evaluate the subject's interpretation of the object's story in terms of transformations. The next question, and last of the section, queries the time it took for the user to understand the evolution of the spatiotemporal phenomenon, which is an important measure in order to compare with other visualization frameworks in the future.

The fifth section also considers the complete story of an object. It starts by requesting the subject to watch a video with the animation of an object over time. Then, question 5.1., similarly to question 4.5., requests the user to describe the object story in terms of transformations. The purpose of describing the evolution of the spatiotemporal phenomenon using a dynamic visual narrative is to highlight the limitations of perceiving small or fast occurring transformations.

Then, using the developed static visual narrative prototype, subjects are shown, in questions 5.2. and 5.3., *zoomed-in* frames of the initial storyboard, revealing transformations that might have been missed due to the common limitations of its dynamic counterpart, asking under the form of multiple-choice answers what the user understands given the new, more granular, storyboard. Lastly, question 5.4. requests users to compare both visual narratives (static and dynamic) in regards to:

1. General perception of what transformations the object suffered;
2. Time required to understand the evolution of the phenomenon;
3. Duration of the dynamic visual narrative (very long or very short videos);
4. Speed at which the object suffers its transformations (very rapidly and almost imperceptible to the human eye, or very slowly and consuming a lot of time to visualize);
5. The feature of filtering through events of interest.

This question was constructed to gather the subject's opinion, backed by the experience and ideas presented in the questionnaire, regarding both types of visual narratives.

The sixth, and last, section is an optional part of the questionnaire. It serves the purpose of gathering observations or comments regarding the questionnaire (6.1.) and the developed framework (6.2.).

6.1.4 Test scenarios

As hinted previously, test scenarios and technical demos were developed for the questionnaire. The technical demos are unitary examples with no subjacent narrative meaning, designed to showcase the various features independently of each other (useful in sections two and three of the questionnaire) and developed using prior knowledge of the application codebase. Nonetheless, for more structured cases (sections four and five), two test scenarios were developed under the form of frame datasets — complying with the data format expected to be received as input. The decision to build datasets as opposed to using existing real-world datasets was made to ensure full control over the evolution of the object, resulting in the desired complete knowledge of the spatiotemporal phenomenon.

For each dataset, a 2D polygon was required and, to remain agnostic to the object, random polygons were generated using a Python script developed by Mike Ounsworth[1]. Having the polygon's vertices, a *obj* file was written and loaded it into Blender[2]. This application allowed to create and edit animations having full control of all parameters. With Blender, a set of keyframes was created using the primitive transformations (translation, rotation, and scale). These keyframes, as the name indicates, are bound to a frame and Blender is then able to generate all other frames using interpolation. Blender supports exporting all frames to one *obj* file per frame. Then a Python

[1] Random 2D polygon generator. Available at https://stackoverflow.com/questions/8997099/algorithm-to-generate-random-2d-polygon/25276331#25276331. Accessed in 2020-05-11.

[2] Blender. Computer graphics software toolset. https://www.blender.org/. Accessed in 2020-05-11.

script compiled all frames into a dataset with the same schema as the expected application input, which was possible due to having prior knowledge of the transformations that occurred. Blender was also used to render the keyframes of each dataset as animations (in video format), which were later used for comparison with the developed static visual narrative counterparts.

6.2 Results and discussion

Since each part of the questionnaire serves a different purpose, the current section follows the same structure, presenting and discussing the results achieved, grouped by questionnaire section.

6.2.1 Introduction and sociodemographics

The first section of the questionnaire served as an introduction to the current work and gathers commonly asked personal details — age range and gender — which ultimately, as expected, did not reveal any important correlation. The age range selection was uniform, as predicted, due to the subjects being college students. The gender distribution also followed the norm of informatics and computing engineering students, with the female gender making up for 21.2% of the subjects.

6.2.2 Interpretation of visual metaphors

The second section of the questionnaire focused on assessing if users understood the transformations that occurred to an object at an individual level — one frame per question.

Results

Regarding the identification of transformations through its visual metaphors, questions 2.1. to 2.7., 95.7% of the answers were correct. The questions regarding translation and multiple transformations were answered with perfect accuracy (100%), while the most frequently wrongfully answered question concerned unknown transformations (which were considered by the subjects who answered wrongly as transformations successfully recognized by the program).

Questions 2.8. and 2.9., which regard the user's perception of the order of object states in a frame, featured opposing results between each other. Question 2.8. mostly got correct answers (87.9%) while question 2.9. representing a frame triggered with an absolute cumulative threshold, only had 30.3%.

Question 2.10. is a checkbox question and featured three correct options in a total of five. From the options selected by the subjects, shown in detail in Figure 6.1, 94.2% were correct while 57.6% of users answered the three correct options without selecting any wrong ones. The most frequently missed correct option was understanding that, in the second frame, the velocity was increasing and, accordingly, the only incorrect option selected was the exact opposite which stated that the velocity was decreasing in the second frame.

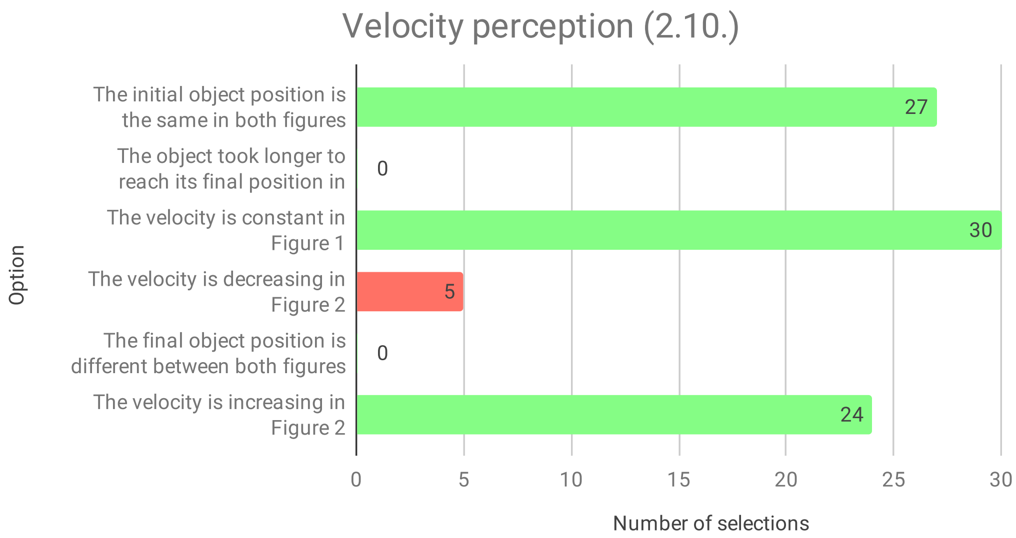


Figure 6.1: Results from question 2.10.

Discussion

The high rate of correct answers in the first seven questions indicates that users accurately identify the transformations of individual frames. However, it is possible that the overlay influenced the decision of some subjects towards the solution and thus the authors recommend, for future work, to query users on identifying transformations without the overlay element. The fact that 12.1% of the subjects mistook unknown transformations for multiple recognized transformations might have been due to the question regarding the latter, which got perfect accuracy, appearing after the former, and the subjects being asked not to revise past answers.

The discrepancy seen between questions 2.8. and 2.9. can be explained by the behaviour of the object shown in the frame. Question 2.8. shows multiple transformations (a translation accompanied by an orientation change) while question 2.9. shows a scale. However, the distinctive aspect is the fact that, in the first question, the intermediate state shown of the object smoothly follows the "expected" path indicated by the transformation. On the contrary, question 2.9. simulates a scale event triggered by a directed cumulative threshold on an object that shrunk and then grew, which might have been unexpected to some subjects since these types of events were not introduced before. This hypothesis is corroborated by the fact that the most frequently answered option indicates a continuous growth, from the smallest to the largest object state.

Question 2.10. suggests that the majority of users correctly perceive acceleration through onion skinning. On the other hand, 60% of the users that understood there was a decrease of velocity instead of an increase in the second figure also failed ordering questions. This may imply that there is a correlation between these questions which indicates the subjects might have perceived an incorrect order of intermediate states once more, leading to the opposite conclusion in terms of acceleration.

6.2.3 Interpretation of transformation units

The third section of the questionnaire served two purposes. The first was to give feedback to the subjects on their assumptions of identified transformations in the first section. This prevented further errors in identifying transformations in future sections by informing which transformation is occurring in each question of the third section. The second objective was to assess if users are able to understand the units of the transformation, possibly resorting to the overlay.

Results

Questions 3.1. to 3.4. feature similar results. Combining the four questions, subjects answered correctly with a rate of 90.9%. The question that stands out with the highest error rate (21.2%) is question 3.2. which regards a frame with multiple transformations (a translation and an orientation change). On the other hand, question 3.1., which queries the units of a translation, featured perfect accuracy.

Discussion

The results from the questionnaire's third section suggest that the majority of users understand the transformations' units, given the available visual and textual elements, despite a general difficulty when multiple transformation events overlap and, thus, are shown in the same frame.

6.2.4 Interpretation of the object story

The fourth section of the questionnaire evaluates the users' interpretation of the object's story using the developed prototype, as well as the effectiveness of temporal and narrative context elements.

Results

Questions 4.1. and 4.2. concern the interpretation of the narrative context — the narrative progress bar presented in 4.3.3, with the first regarding the total number of frames and the second the number of visible frames. Both questions expected short textual answers and, after discarding improper answers (e.g., negative numbers from subjects who later admitted to not having understood the concept of frames), the subjects showed a large correct rate discrepancy between these questions. Question 4.1. had little over half correct answers (51.6%) while question 4.2. fetched a correct rate of 80.6%. The most frequent wrong answer given in question 4.1. was that there were three frames in total, which matches the number of visible frames which, in turn, is the correct answer for question 4.2.

Question 4.3. regards the duration of the narrative in quantitative terms. As seen in question 4.1., the results were once again evenly distributed between correct and incorrect answers (correct rate of 51.6%). On average (and discarding answers with no units of time), the wrong answers represent an observable duration of 6 months, which is half the correct duration of the narrative.

Question 4.4. explores both narrative and temporal context interpretation under the form of a checkbox question. From all the options considered to be correct by the users, shown in detail in Figure 6.2, 89.5% were true statements. On the other hand, 4 subjects admitted to not having understood the question, its options or the concept definitions required to answer. The most frequent false statement that the users considered to be correct states that the narrative ends at the end of December, while in fact it is ending December 15th.

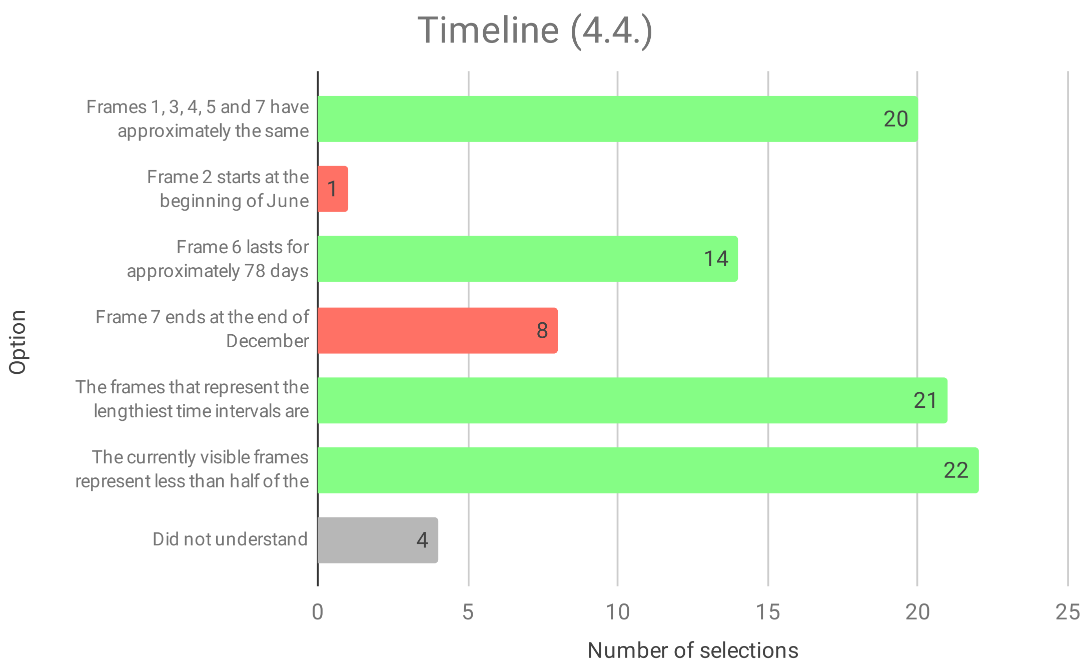


Figure 6.2: Results from question 4.4.

Question 4.5. addressed the interpretation of the object's story with a static visual narrative. Due to the unsupervised nature of the evaluation artefact, most of the inquired people did not answer with the appropriate requested detail (transformation names, units and duration) or skipped the long textual answer altogether, as seen in Figure 6.3. Nonetheless, the subjects who answered according to the requested details all accurately explained the story of the object. From the users which either answered with just transformation names and units (disregarding duration) or only the transformation names, most interpreted the narrative correctly although there is a significant portion that left out the immutability periods from their answers. From the 7 users who answered according to the requested details, 3 revealed they understood the narrative in less than 10 seconds, 2 between 10 and 30 seconds, while the remaining 2 between 30 seconds and 1 minute. When considering all the subjects, the percentage of time intervals selected deviate significantly, having a percentage of 30.3%, 39.4% and 18.2%, respectively. Amongst the 33 subjects who answered the questionnaire, only 1 admitted to not having understood the object's story.

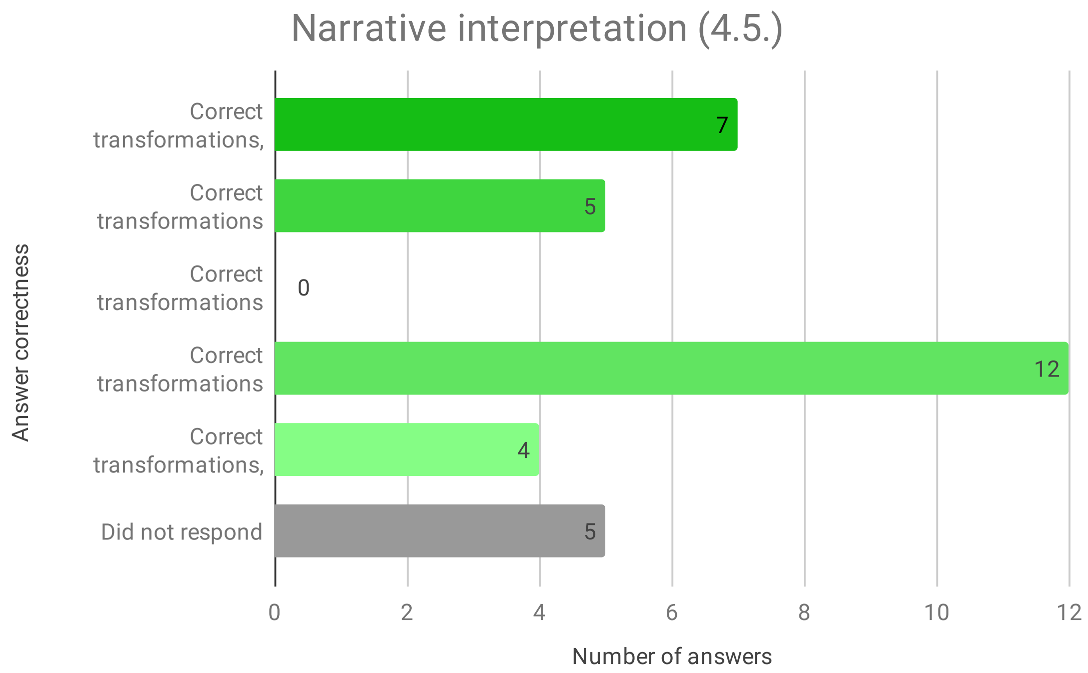


Figure 6.3: Results from question 4.5.

Discussion

From the first questions of this section, there seems to be a common difficulty in understanding the timeline element. On the other hand, the results of object story interpretation with a static visual narrative are promising since most subjects understood the transformations that occurred to the object, its units, duration, and order. The lack of user supervision led to inappropriate levels of details in long textual answers although the users who were thorough in their answers perfectly succeeded in explaining the phenomenon's story. The omission of the immutability periods shown by some of the users might have been the consequence of how the question was phrased as it specifically requests transformations while expecting all the frames to be explained.

6.2.5 Comparison between static and dynamic visual narratives

The fifth section of the questionnaire compares the developed static visual narrative with a primitive dynamic approach — a video animation. This section utilizes a dataset which contains some transformations that occur very fast, are very small, or occur simultaneously with other transformations. The goal behind the usage of this dataset is to assess the advantages and disadvantages of the solution proposed in Chapter 4 in a more realistic scenario.

Results

The first question requires the subjects to watch a video animation of the dataset and, afterwards, to describe the evolution of the object in terms of the transformations. The diagram in Figure 6.6 represents the dataset used for this section. In the diagram, the transformation units of the red events are so small that they might be missed when watching the video. As expected, 76.2% of the subject missed the translation from frame 225 to frame 255. Unsurprisingly, all of the users missed the small uniform scale of 0.95 units. On the other hand, 90.5% of the subjects correctly identified the orientation changes that occurred in simultaneous with the first translation. As seen in Question 4.5. and despite the question description explicitly requesting immutability periods to be documented, 90.5% of the users failed to recognize there was an immutability period between the first and second translations. The rest of the transformations were, in general, correctly identified with the most frequently missed of those transformations being the uniform scale of 0.4 units that occurred from frame 210 to frame 225.

Animation video interpretation (5.1.)

	Continuous translation with punctual orientation changes				Translation and scale simultaneously		Scale and translation simultaneously		Orientation and scale simultaneously	
Transformation	**Translation**	**Orientation**	**Orientation**	**Immutability**	**Translation**	**Scale**	**Scale**	**Translation**	**Orientation**	**Scale**
Hit	100.00%	95.24%	90.48%	9.52%	100.00%	85.71%	90.48%	**23.81%**	95.24%	**0.00%**
Miss	0.00%	4.76%	9.52%	90.48%	0.00%	14.29%	9.52%	**76.19%**	4.76%	**100.00%**

Blue: transformation should be **easily** identified
Red: transformation should be **difficult** to identify using dynamic visual narratives

Figure 6.4: Results from question 5.1.

Questions 5.2. and 5.3. both assess the interpretation of zoomed frames. The first as a checkbox question while the second is a multiple-choice question. Question 5.2. had three true statements and three false statements. In total, 54 true statements were checked while the subjects also considered 26 false statements as correct. The most selected option, by 75.8% of the subjects, was a true statement that said the object suffered two distinct orientation changes. On the other hand, the second most selected option, by 54.6% of the subjects, was false, stating that the translation showed in the frame that got zoomed-in was in fact multiple distinct translations. Closely following these options were the remaining correct statements, leaving a large gap to the rest of false statements. From the 33 subjects that responded to the questionnaire, only 3 admitted to not having understood enough of the concept to reply to this question, in spite of not confessing this statement in the following question. Question 5.3. followed a similar pattern to the previous question in terms of the logic applied by the users. The correct option, which stated there was only one orientation change and a scale that overlapped the orientation change at some time interval, was the least selected with a percentage of 18.2%. Similarly to the previous question, the majority of users (57.6%) selected the option that stated the same as the correct one with the nuance that the object suffered in fact three distinct orientation changes, where one of them featured a scale simultaneously. The remaining option, chosen by 24.2% of the subjects, implied there were one orientation change and one scale which fully overlapped in time.

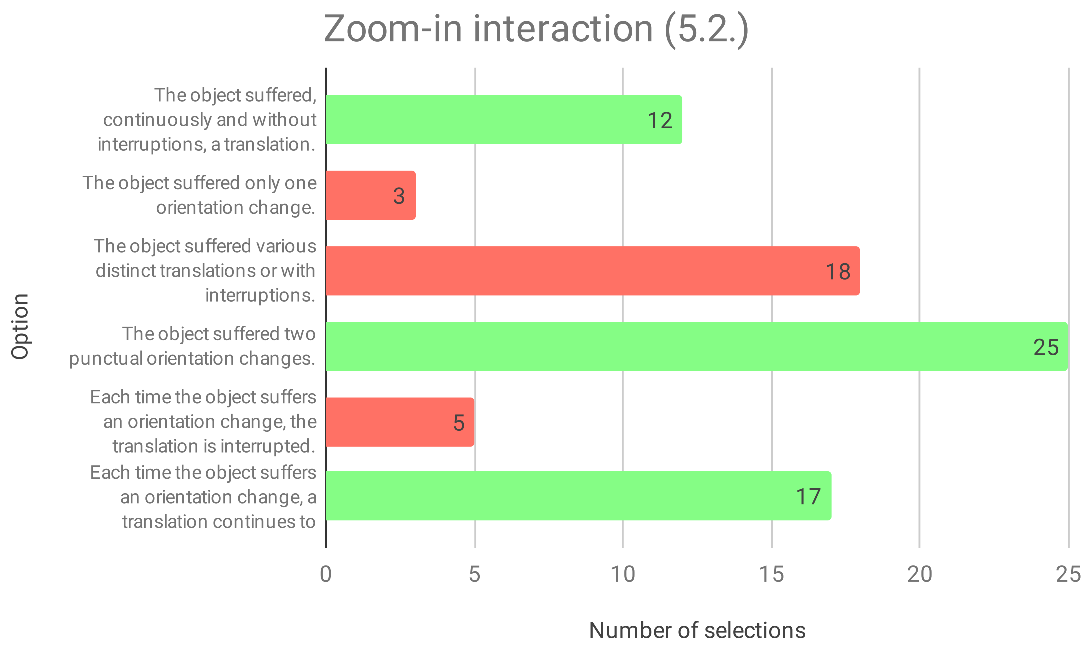


Figure 6.5: Results from question 5.2. In red are the statements that should have been considered false and in green are the correct statements

The final question of this section is opinative in nature. Nonetheless, these opinions are relevant since they come from potential users who have literacy on visualization tools and have been in contact with other data visualization solutions before due to the field of their college degree. The most noted advantages identified by the subjects regarding static visual narratives were the following:

1. **Easier identification and quantification of transformations.** The majority of users reveals that the static approach was more effective at understanding the phenomenon's story. Furthermore, some subjects say that indicating transformation units is only possible in the static version — making it the most suited for delivering information — since the animation video lacks the visual aids or user tools that complement with more information.

2. **Improved accuracy.** Most subjects point to improved perception of events that might be missed to the naked eye due to being small transformations, too complex, or for occurring too fast. Moreover, users admit static visual narratives reduce interpretation errors in those scenarios.

3. **Improved efficiency.** Concerning spatiotemporal phenomena that, when represented as video animation, result in long videos, most subjects believe static visual narratives would

be advantageous since the time to consume would be shorter. Moreover, videos require more attention to note all the transformations that occurred to the object, its units, duration, and relevant timestamps, making the experience tiresome. On the other hand, users acknowledged that the proposed solution showed multiple storyboard frames at the same time which helped in summarizing the whole story. This summarizing aspect solves some of the users' critiques towards long videos — the difficulty in recalling what happened after watching the video. Building on that problem, some users reveal forgetting what happened in the video, leading to re-watching it multiple times (especially the fastest bits).

4. **Easier to explore.** The majority of users say that having an approach based on levels-of-detail is beneficial to reveal details that could have been missed in the dynamic visualization by analysing a transformation in isolation. The users find advantageous to be able to filter the content and consider it easier to access specific moments in time or discovering what happened in a certain time period — an action that would otherwise require users to look back and forth through the video.

On the other hand, users also pointed out some disadvantages with the proposed solution:

1. **Animations are more intuitive and effortless.** A minority of users reveals that analysing the static version of the narrative required more time to process and understand the transformations due to having to create a "mental video" of the phenomenon.

2. **Animations are more suited for simple scenarios.** Most subjects admit that the static approach has a comprehension overhead that might not be worth it to deal with in simple phenomena that could be represented as a short animation.

3. **Increased level of indirection and segmentation.** A user suggested that the segmentation of events through frames can lead to a disconnection with the phenomenon as a whole. This idea is further reiterated by another subject who states that the proposed solution hinders the perception of time. Furthermore, a few users mention as a disadvantage the increased level of indirection to understand a period of time with more detail, showing a common difficulty in understanding which events happened simultaneously.

Discussion

In general, the results from the first question corroborate some of the problems identified in Section 3.1 regarding dynamic visual narratives — the difficulty in perceiving small or fast transformations. The fact that, from the transformations that users were expected to identify, the transformation most frequently missed was a scale that more than halved the size of the object suggests that multiple simultaneous transformations might be difficult to perceive through dynamic visual narratives. The overall omission of the immutability period might either have been caused by subjects not acknowledging its importance to the question or missing it entirely due to being a short period relative to the dynamic visual narrative duration.

The results from questions 5.2. and 5.3. suggest that users might not perceive the continuity of transformations when presented as different frames, which is present in both questions. To describe the segmentation of frames, consider the third question where a scale transformation occurs (begins and ends) while there is already an orientation change happening, as shown in the dataset diagram of Figure 6.6. In this figure, the events in blue appear in the main storyboard while the events in red appear in a separate storyboard row when the frames corresponding to the overlapping blue events get zoomed in. The adopted approach to handle such cases is to divide into three frames: one frame before the scale, indicating there was already an orientation happening; another frame with multiple transformations, depicting the scale and the continuation of the orientation that it overlapped; one last frame afterwards that completes the orientation change. This segmentation strategy may have hindered the immediate perception of the phenomenon's evolution. On the other hand, a more far-fetched possibility is that users correctly considered that there was only one orientation change in the new storyboard row of question 5.3. but, seeing the other two from question 5.2., chose, in the third question, the option that stated that there were a total of three orientation changes.

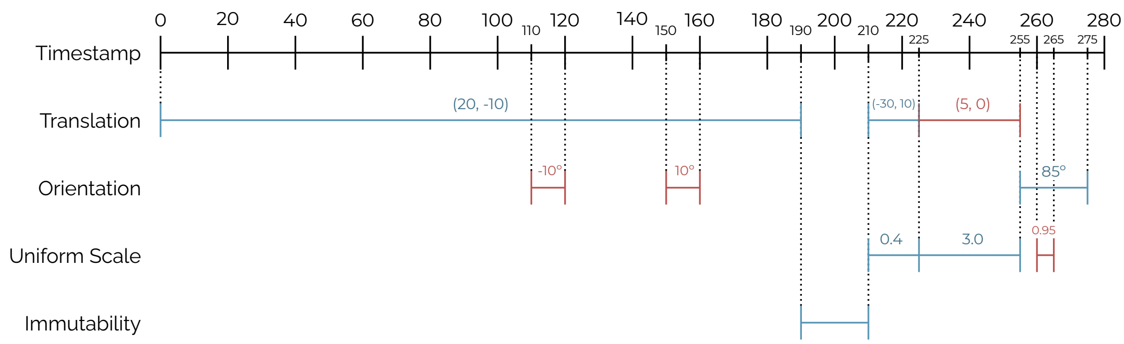


Figure 6.6: Diagram of the dataset used in the fifth section of the questionnaire

As stated in Section 1.3, a single visualization framework is seldom the irrefutable choice for every possible scenario. From the feedback gathered in question 5.4., users prefer the effortlessness and intuitiveness of an animation video for simple and short spatiotemporal phenomena as long as no fast or small-to-detect transformations occur. On the other hand, the majority of subjects affirm the visualization framework is more suitable for conveying scientific information due to better accuracy and efficiency at telling the story of the object. Users state that the proposed method should be applied to cases with complex phenomena, fast occurring or potentially imperceptible transformations, or stories lasting for long periods of time. There seems to be another dichotomy in the users' opinions: those who value freely exploring the data and those who consider it a level of indirection that leads to a disconnection of the event itself. It is essentially the dichotomy seen in Section 2.2 between *Reader* and *Author*-driven visual narratives, with some users preferring to discover the history behind the data and others being delivered the story effortlessly, respectively. The proposed solution is a hybrid approach regarding this taxonomy since its author is the framework itself which generates the initial storyboard and then allows the reader to freely interact with the visual contents. Therefore, the aforementioned advantages and disadvantages are to be expected. Lastly, the difficulty of perceiving time is a justified limitation of the solution proposed in Chapter 4 as should be addressed in future work, for example by resizing the width of the storyboard frames proportionally to their duration.

6.2.6 User observations section

The final section of the questionnaire allowed users to express their opinion on the questionnaire and regarding the developed prototype.

Concerning the questionnaire, the main critique was its length: considered to be too long to complete. Some subjects mentioned that questions 5.2. and 5.3. should be more explicit as some options were partially right and selecting the "most correct" ones was considered a difficult task.

Regarding the visualization prototype, users, in general, found it interesting and useful and left some improvements to be made in the future:

1. Easier parameter configuration;
2. Being able to filter a transformation altogether via a switch instead of nullifying all the parameters of that transformation;
3. Dividing the settings modal into tabs instead of having to scroll down through the different settings sections;
4. Disabling the navigation arrows when reaching either bound of the narrative.

Hence, the improvements mentioned by the users are related to the user experience of the experimental prototype and do not regard the conceptual solution presented in Chapter 4. The last item had been identified by the authors and was fixed after the evaluation process. Nevertheless, these improvements could have been identified earlier if a heuristic evaluation was conducted by a visualization specialist.

6.3 Research questions assessment

This section answers the research questions presented in Section 3.4 with the knowledge gathered throughout the study, conceptualization, implementation and testing of the visualization framework.

1. **Are static visual narratives effective in describing the evolution of a spatiotemporal phenomenon?** According to the results from Section 6.2, static visual narratives are effective at telling the story of an object. It was considered by potential users to be suitable for scientific visualization since its individual frames contain the necessary information to identify transformations, its units and duration. Considering the complete narrative of a phenomenon, users were also capable of describing it with high accuracy with the exception of interpreting the meaning of zoomed-in frames.

2. **Are static visual narratives efficient when compared to their dynamic counterpart?** The evaluation stage revealed that static visual narratives can outperform dynamic approaches in scenarios with potentially imperceptible transformations — either due to its fast pace or small units of transformation — or for data that spans for long periods of time. These advantages match the limitations of dynamic visualizations presented in Section 3.1. On the other hand, dynamic visual narratives such as animated videos are still relevant and preferred by some users for simple and short spatiotemporal phenomena.

6.4 Summary

This chapter documented the evaluation of the proposed solution with the goal of answering the research questions presented in Section 3.4. The evaluation process consisted of user studies through a questionnaire. The questions were developed according to the work of Zhu [58], evaluating accuracy, utility, and efficiency to the extent at which is possible to assess these principles in a questionnaire. Concerning the structure of the questionnaire, it was divided into six sections, each serving a different purpose. Besides the introductory and observations sections, the other four can be seen as two separate parts of the questionnaire with two sections each. The first part concerning storyboard frames as individual elements and the second part addressing the narrative as a whole. The first part analysed the identification of transformations through the visual metaphors in individual frames and the interpretation of transformation units. The second part assessed the interpretation of complete narratives and compared its usage with dynamic visual narratives. The research questions presented in Section 3.4 could then be answered in light of the results obtained in the questionnaire. It was concluded that static visual narratives are effective at transmitting spatiotemporal information with good precision and few interpretation errors, in spite of overall user difficulties in understanding zoomed-in frames and, to a lesser extent, the timeline element. The comparison between static and dynamic approaches towards spatiotemporal phenomena visualization resulted in the conclusion that static visual narratives are more suitable in scenarios with potentially imperceptible transformations or for data that represents long periods of time. On the other hand, dynamic approaches are still preferred by users for its simplicity and intuitiveness in simple and short spatiotemporal phenomena scenarios.

Chapter 7

Conclusions

This chapter presents the conclusions drawn from the current work. Section 7.1 summarizes the conclusions taken from each research stage. Afterwards, Section 7.2 retrospects the scientific contributions of this work, answers the research questions presented in Section 3.4, and re-analyses the hypothesis in light of those answers. Lastly, Section 7.3 proposes future work towards improving the presented solution.

7.1 Conclusions

The current work reaffirmed the importance of information visualization towards the analysis of data in an age where enormous amounts of data are gathered. Regarding the spatiotemporal domain, it was concluded that the current approaches towards the visualization of spatiotemporal phenomena are limited and unsuitable for certain scenarios. Dynamic visualization approaches were considered inappropriate for scenarios where objects suffered fast or small transformations that could be missed by the user and scenarios with long periods of object immutability or slow occurring transformations. Previous static approaches, on the other hand, had not been developed towards representing spatiotemporal phenomena. Finally, neither type of visualization methods allowed the definition of events of interest.

The outcomes of the literature review regarding spatiotemporal phenomena visualization, presented in Chapter 2, mentioned a vast set of object features concerning spatial, temporal and spatiotemporal aspects. The common principle to those features was the ability to identify changes by defining events of interest through a series of documented parameters. This concept was not found when reviewing visualization methods for spatiotemporal data, encompassing a promising research gap. The state of the art study of visual narratives and visualization enabled the authors of the current work to later design and assess a static visual narrative template and visual variables suitable for the representation of the various object transformations.

With the knowledge from the literature review and the definition of the problems seen in current information visualization approaches, the authors of this work decided to postulate the hypothesis that static visual narratives could efficiently represent the evolution of spatiotemporal phenomena. Towards corroborating this hypothesis, a novel visualization framework was envisioned based on automated generation of interactive storyboards that summarize the evolution of a spatiotemporal phenomenon through a set of frames that represent the most relevant change across all events of interest.

Towards assessing the feasibility of the presented visualization framework, a functional prototype was developed as part of a system that was composed of an object change quantification server and a client visualization application. The developed prototype addresses the client component of the system and mapped the visualization pipeline that was earlier presented into functional application modules. Throughout the development and testing of the prototype, new parameters were added to address information noise and parameter adjustment on requesting more detailed frames. It was also concluded that visual clutter is a problem that affects the effectiveness of visualization frameworks and, despite mitigating it with polygon simplification algorithms and limiting the number of visible intermediate states, is still present in certain situations.

In order to evaluate the efficiency and effectiveness of the solution proposed in Chapter 4, a questionnaire was conducted with engineering students. The questionnaire addressed the interpretation of the static visual narrative (i.e., whether the users were able to understand the evolution of the spatiotemporal phenomenon) and the comparison between static and dynamic visualization approaches. The results from the evaluation process allowed the authors to conclude that static visual narratives are effective at transmitting spatiotemporal information with adequate precision, although some difficulties were noted in understanding zoomed-in frames. The comparison between static and dynamic approaches concluded the former is more suitable in scenarios with potentially imperceptible transformations or for data that represents long periods of time. Nevertheless, it was concluded that users prefer the simplicity and intuitiveness of dynamic approaches for short scenarios that encompass only transformations that are easily perceivable.

7.2 Research results

The current work aimed to add scientific value to the field of spatiotemporal visualization through the following contributions:

1. **Systematic literature review** of spatiotemporal phenomena, visual narratives, and information visualization. Reviewing current visualization methods for spatiotemporal data allowed for the formalization of the core problem addressed by this work and it may be useful for the definition of other problems in future work. Moreover, the state-of-the-art chapter provides guidelines on designing, developing and testing a visualization framework.

2. **Conceptual visualization framework** based on the automated generation of static visual narratives that describe the evolution of a spatiotemporal phenomenon in terms of object

transformations. Towards solving the identified problem, a visualization framework was conceptualized, remaining agnostic to implementation specifics or environments. Hence, the proposed solution chapter provides a conceptual description on how this problem could be solved in terms of defining events of interest, visually mapping spatiotemporal transformations and other elements, and adapting the visualization pipeline to the current domain.

3. **Visualization prototype** that implements the conceptual visualization framework. Having a materialized version of the proposed solution is valuable since it enables testing and, ultimately, becomes a viable tool for spatiotemporal visualization. The prototype is also freely available as an open-source project (see Appendix A).

4. **Evaluation results** can be considered a scientific contribution since they allow researchers to find aspects that can be improved, resulting in more scientific work. The evaluation results are also publicly available (see Appendix B.2).

As a result of the aforementioned contributions, the authors of the current work were able to answer the research questions presented in Section 3.4:

1. **Are static visual narratives effective in describing the evolution of a spatiotemporal phenomenon?** According to the results from Section 6.2, static visual narratives are effective at representing the story of a spatiotemporal phenomenon in terms of events of interest.

2. **Are static visual narratives efficient when compared to their dynamic counterpart?** The evaluation stage revealed that static visual narratives are more efficient than dynamic approaches in scenarios that either span for long periods of time or contain very granular or fast-occurring transformations. On the other hand, short phenomena that are composed of easily perceivable transformations are considered to be more efficiently represented by dynamic visualizations.

Considering the answers given to the research questions, the authors conclude that summarizing spatiotemporal phenomena through static visual narratives is an effective and efficient method for understanding the evolution of these objects in terms of transformations for a subset of possible scenarios.

7.3 Future work

Given that the current work represents a novel approach to spatiotemporal visualization through static visual narratives, there is a vast set of visualization elements and features to be explored or enhanced. Therefore, the future work identified by the authors of the current work encompasses:

1. **Exploring different layouts.** As presented in Section 4.3.1, the hierarchical layout could gradually decrease the detail of the visual metaphors and the frame's relative size for the storyboard rows closer to the root. Such a layout would increase the focus on the row that

is currently being analysed. An improved version of the layout would feature more fluid navigation (possibly by dragging through rows to see other frames or through horizontal scrolling), replacing the strict navigation which replaces the page's content with no animated transitions.

2. **Studying methods for mitigating visual clutter.** Section 5.2.4 explained how visual clutter was addressed in the current work. It was pointed out that the presented solutions have limitations and should, therefore, be further studied. A more intuitive approach to limiting the number of intermediate states could be based on a new numeric setting that would represent the time between states. In contrast, a solution based on heuristics regarding intermediate state density (or state overlap) could be devised, similarly to the implementation of polygon simplification presented in Section 5.2.4. Regarding vertex density and its impact on visual clutter in some visual metaphors, the polygon simplification approach could also be improved since, for example, when the scene camera is far away from the objects, the dashed lines that compose vertex mapping introduce visual clutter as they are condensed to the same area due to the camera's zoom level.

3. **Studying methods for visually representing the temporal dimension.** As discussed in Section 6.2, some users refer a disconnection between the transformations that occurred to the object and their duration. A possible study could be resizing a frame's width proportionally to the frame's time interval. However, this approach is non-trivial since if this relationship is linearly proportional, some frames could be too narrow to view and others could be too wide to fit the user's screen.

4. **Studying multi-layered user interfaces.** Accommodating requirements from both a general public and from industry specialists is not a trivial task in any type of scientific visualization. In the case of the current work, the parameterization of spatiotemporal features can be complex and non-intuitive for the general public while being desirable customization aspects for data analysts. Therefore, exploring multi-layered user interfaces — interfaces that have multiple layers of complexity — could improve user experience across all levels of expertise.

5. **Improving the visual metaphors.** Despite the largely positive results in transformation identification, some visual metaphors could be improved, especially concerning *Multiple transformations*. For multiple transformations, the visual metaphors of its individual transformations could be combined to form a new visual metaphor that takes into account the transformations that occurred. Furthermore, the visual metaphors for transformations could also depend on the type of threshold that triggered the event (delta or cumulative, directed or absolute). A communication design study addressing the visual metaphors would corroborate the selected metaphors, further improve them or select more efficient metaphors.

6. **Adopting a hybrid approach.** Even though this work focused on static approaches for visualizing spatiotemporal events, a hybrid approach was considered for exploration in future

work. This hybrid approach could consist, for example, in allowing for either a frame or a storyboard row to be visualized as an animation of smoothly interpolated object states. Such an approach could potentially fulfil the preference that the questionnaire subjects revealed for dynamic visualizations regarding simple and short phenomena.

7. **Conducting improved evaluation.** Since the conditions and resources for evaluation were limited, the results might not be generalizable. Therefore, there are opportunities for improvement in the evaluation stage. To avoid sampling bias, the authors propose conducting evaluation using probability sampling methods. Moreover, the testing methodology could be further enhanced by allowing subjects to interact directly with the prototype. Finally, testing with individual users would ensure more direct communication between the authors and the user performing the test, allowing the user to expose doubts and learn more about the visualization framework throughout the process.

Other aspects of the overall system could be improved by extending the change quantification server component which, in turn, would result in more topics to be explored for the visualization framework. This is the case for supporting more spatial transformations (e.g., affine transformations), dependent variables (e.g., object temperature changes), temporal features (e.g., cyclic events and patterns), and derived variables (e.g., changes of velocity). Regarding the concept of events of interest, various aspects could significantly improve the user experience, such as automatic tweak of the parameters based on the spatiotemporal data or artificial intelligence approaches that consider the user's behaviour and interests. Another topic for future work would be to further extend the events of interest notion into aggregations of transformations (e.g., a user might not be interested in punctual scales or translations, but events where a scale and translations simultaneously occur might be of interest to the user).

Appendix A

Prototype

The prototype's source code is available at `https://github.com/rendoir/feup-thesis`.

A.1 Communication interface

The communication between the system components shown in Section 5.1 is conducted via HTTP requests. For the scope of this work, the most important routes are related to retrieving storyboards and dataset metadata. For more details consult the API documentation [1].

- GET */storyboard/metadata/{id}*: Retrieves metadata regarding the dataset with key *id*. The expected response for this request is a JSON object as Listing A.1 exemplifies. For the moment, it contains a *timePeriod* (time in milliseconds between snapshots), *startTime* (timestamp in milliseconds of the beginning of the dataset), and *name* (dataset name). This metadata schema can easily be extended in the future if there are information needs for author name, versioning, or others.

```
{
  timePeriod: 5,
  startTime: 1441670400000,
  name: "Iceberg Dataset"
}
```

Listing A.1: Example of a metadata response

- POST */storyboard/{id}*: Submits the parameters to process the dataset and returns the frames of the storyboard. The request supports two query parameters: *initialTimestamp* and *finalTimestamp*. These are useful when requesting a zoomed storyboard, limiting the initial and final timestamps to the ones from the zoomed frame. Listing A.2 demonstrates an example

[1] API Documentation. Available at `https://app.swaggerhub.com/apis-docs/EdgarACarneiro/thesis/`. Accessed in 2020-06-15.

of the request's *body*. The request sends the parameters as a JSON object for each supported spatiotemporal feature as discussed in Section 4.2. Listing A.3 illustrates an example of a storyboard response, with a frame where a translation event was detected via directed cumulative threshold with a translation vector of $(80.0, 90.0)$ lasting for 100 milliseconds.

```
{
  parameters: {
    translation: {
      delta: 10.0,
      directedAcc: 100.0,
      absoluteAcc: 200.0,
      noiseEpsilon: 0.5
    },
    rotation: {
      delta: 5.0,
      directedAcc: 10.0,
      absoluteAcc: 20.0,
      noiseEpsilon: 0.1
    },
    scale: {
      delta: 1.1,
      directedAcc: 1.5,
      absoluteAcc: 2.0,
      noiseEpsilon: 0.05
    },
    immutability: 1000
  }
}
```

Listing A.2: Example of a storyboard request's body

```
[
  // Other frames...
  {
    events: [
      // Other events...
      {
        threshold: "DIRECTED_ACC",
        type: "TRANSLATION",
        trigger: {
          transformation: [ 80.0, 90.0 ]
        }
      }
      // Other events...
    ],
    temporalRange: [ 1593734400000, 1593734400100 ],
```

```
    phenomena: [
      // Other object states...
      {
        representation: [
          [ 0, 5 ],
          [ -5, -5 ],
          [ 5, -5 ]
        ],
        timestamp: 1593734400000
      }
      // Other object states...
    ]
  }
  // Other frames...
]
```

Listing A.3: Example of a storyboard response

Appendix B

Evaluation

B.1 Questionnaire

To evaluate the solution proposed in Chapter 4, a questionnaire (shown below) was conducted and is available at `https://forms.gle/9oAzsXEKff681tKe8`.

Spatiotemporal data visualization

In the context of the Master dissertation "Spatiotemporal Phenomena Summarization through Static Visual Narratives" in Informatics and Computing Engineering, we ask for your collaboration in this questionnaire.

The developed application, shown in the figure below, summarizes the story of an object that represents a spatiotemporal phenomenon in terms of geometrical transformations. This summary (visual narrative) is composed of various frames which visually represent the transformations that occurred to the object. The transformations are dependent on a set of user-introduced parameters which decide to what values a transformation is considered "relevant". By clicking on a frame, a new summary is shown bellow it with more detail regarding that frame. In other words, the transformations are finer-grained due to less restrictive parameters.

The questionnaire is anonymous, no personal data is gathered (apart from age range and gender) and the obtained results will be used exclusively for academic purposes. It is desirable that you answer each question spontaneously, sincerely and keep the answer even if the solution is presented later in the questionnaire. If you are unsure about a question or consider the correct answer to be missing from the options, please resort to the "Other" option, if available. For any doubts or comments, please contact the author via email up201503822@fe.up.pt. The estimated duration of the questionnaire is 15 to 25 minutes.

We profoundly thank your collaboration.
* Required

1.1. Age range

Mark only one oval.

- < 18
- 18-30
- 31-50
- 51-65
- > 65

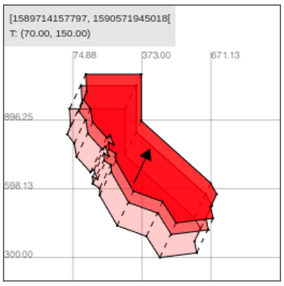
1.2. Gender

Mark only one oval.

- ⬭ Female
- ⬭ Male
- ⬭ Other: ____________________

Visual metaphors	This section evaluates the effectiveness of various visualization elements individually in qualitative terms.

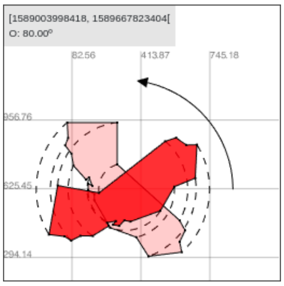
2.1. What transformation occurred to the object? *



Mark only one oval.

- ⬭ Translation
- ⬭ Scale
- ⬭ Orientation (rotation around the centroid (central point of the object))
- ⬭ Pivot rotation
- ⬭ Multiple transformations
- ⬭ Other: ____________________

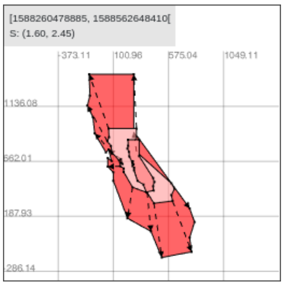
2.2. What transformation occurred to the object? *



Mark only one oval.

- ⬭ Translation
- ⬭ Scale
- ⬭ Orientation (rotation around the centroid (central point of the object))
- ⬭ Pivot rotation
- ⬭ Multiple transformations
- ⬭ Other: ____________________

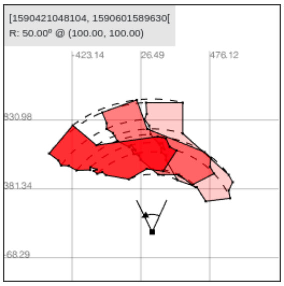
2.3. What transformation occurred to the object? *



Mark only one oval.

- Translation
- Scale
- Orientation (rotation around the centroid (central point of the object))
- Pivot rotation
- Multiple transformations
- Other: ______________________

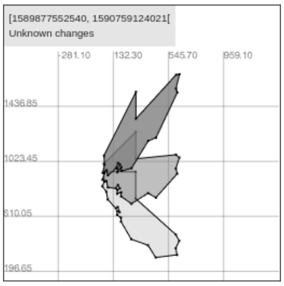
2.4. What transformation occurred to the object? *



Mark only one oval.

- Translation
- Scale
- Orientation (rotation around the centroid (central point of the object))
- Pivot rotation
- Multiple transformations
- Other: ______________________

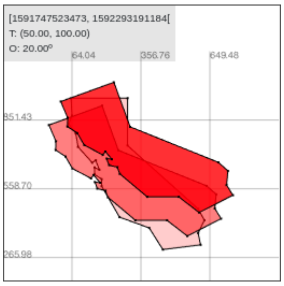
2.5. What transformation occurred to the object? *



Mark only one oval.

- ◯ The object did not suffer any transformations
- ◯ The object suffered a transformation that the program could not qualify
- ◯ The object suffered multiple transformations recognized by the program
- ◯ Other: ____________________

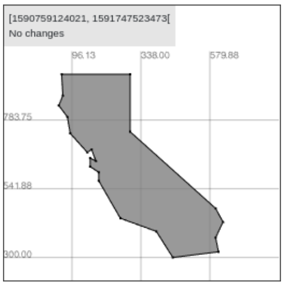
2.6. What transformation occurred to the object? *



Mark only one oval.

- ◯ The object suffered a transformation that the program could not qualify
- ◯ The object did not suffer any transformations
- ◯ The object suffered multiple transformations recognized by the program
- ◯ Other: ____________________

2.7. What transformation occurred to the object? *

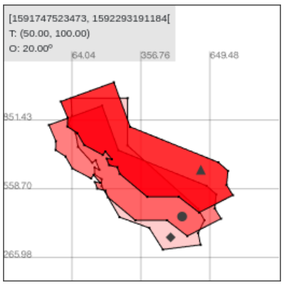


Mark only one oval.

- ◯ The object suffered a transformation that the program could not qualify
- ◯ The object suffered multiple transformations recognized by the program
- ◯ The object did not suffer any transformations
- ◯ Other: ____________________

2.8. Chronologically order the object states *

Use the symbols shown in the figure to order the object states from the oldest to the most recent.

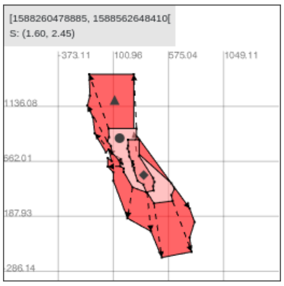


Mark only one oval.

- ◯ ▲ → ● → ◆
- ◯ ▲ → ◆ → ●
- ◯ ● → ▲ → ◆
- ◯ ● → ◆ → ▲
- ◯ ◆ → ▲ → ●
- ◯ ◆ → ● → ▲
- ◯ Other: ____________________

2.9. Chronologically order the object states *

Use the symbols shown in the figure to order the object states from the oldest to the most recent.



Mark only one oval.

- ◯ ▲ → ● → ◆
- ◯ ▲ → ◆ → ●
- ◯ ● → ▲ → ◆
- ◯ ● → ◆ → ▲
- ◯ ◆ → ▲ → ●
- ◯ ◆ → ● → ▲
- ◯ Other: ______________________________

2.10. What do you perceive in these images in terms of velocity and acceleration? *

Select the options that you consider to be correct.

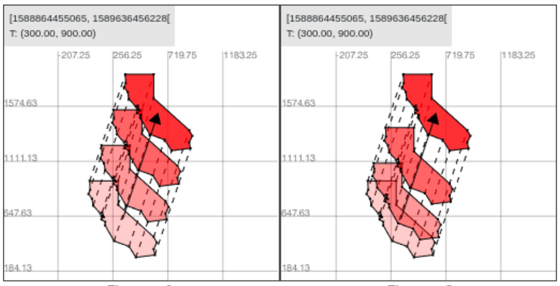


Figure 1 Figure 2

Check all that apply.

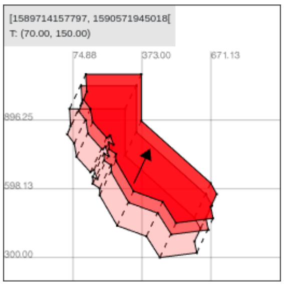
- ☐ The initial object position is the same in both figures.
- ☐ The object took longer to reach its final position in Figure 2.
- ☐ The velocity is constant in Figure 1.
- ☐ The velocity is decreasing in Figure 2.
- ☐ The final object position is different between both figures.
- ☐ The velocity is increasing in Figure 2.

Other: ☐ ______________________________

Visual metaphors	This section evaluates the effectiveness of various visualization elements individually in quantitative terms.

3.1. Describe the translation *

This frame represents a translation. Describe the movement units for each axis. Consider the positive X-semiaxis to be along the right direction and the positive Y-semiaxis along the upwards direction.

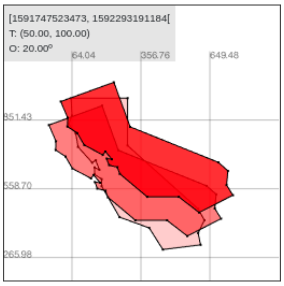


Mark only one oval.

- The translation vector is X=70 and Y=150
- The translation vector is X=150 and Y=70
- The translation vector is X=-70 and Y=150
- The translation vector is X=-70 and Y=-150
- The translation vector is null
- Other: ______________________________

3.2. Describe the transformation *

This frame represents a transformation composed of multiple transformations. Identify what are those transformations and their units.

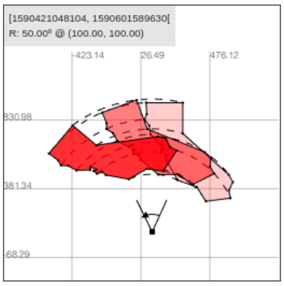


Mark only one oval.

- Translation of 50 units in X and 100 units in Y, accompanied by an orientation change of 20º in the counter-clockwise direction
- Rotation of 20º in counter-clockwise direction centred around the point X=50, Y=100, accompanied by other transformations unrecognized by the program
- Translation of 50 units in X and 100 units in Y, accompanied by an orientation change of 20º in the clockwise direction
- Rotation of 20º in clockwise direction centred around the point X=50, Y=100, accompanied by other transformations unrecognized by the program
- Other: ______________________________

3.3. Describe the pivot rotation *

This frame represents a pivot rotation (rotation around a pivot). Identify the angle and direction of rotation and its pivot. Consider the positive direction as the counter-clockwise direction.

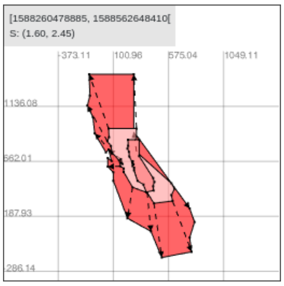


Mark only one oval.

- ( ) 50º in clockwise direction around the centroid of the object
- ( ) 50º in counter-clockwise direction around the centroid of the object
- ( ) 50º in clockwise direction around the point X=100, Y=100
- ( ) 50º in counter-clockwise direction around the point X=100, Y=100
- ( ) Other: ____________________

3.4. Describe the scale *

This frame represents a scale. Identify the units of this transformation, considering that values inside the range [0, 1[represent a decrease in object size while values above 1 represent an increase in size. Consider the positive X-semiaxis to be along the right direction and the positive Y-semiaxis along the upwards direction.

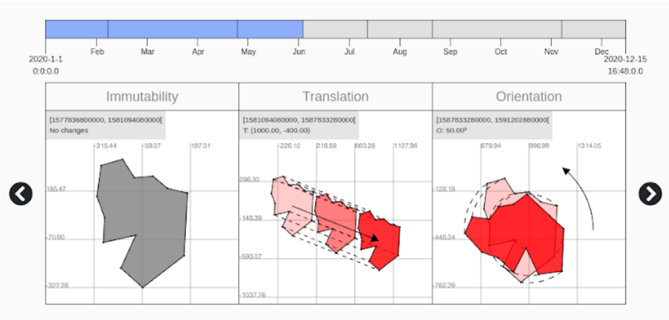


Mark only one oval.

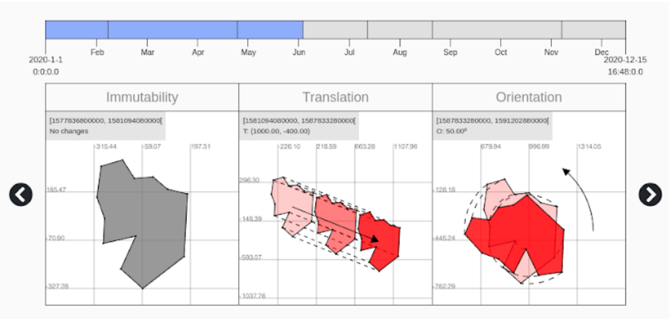
- ( ) The object size increased with a scale of X=1.6 and Y=2.45
- ( ) The object size decreased with a scale of X=1.6 and Y=2.45
- ( ) The object size increased with a scale of X=2.45 and Y=1.6
- ( ) The object size decreased with a scale of X=2.45 and Y=1.6
- ( ) Other: ____________________

Visual narrative	This section evaluates the effectiveness of the static visual narrative as a means to convey a story based on spatiotemporal data. The visual narrative is composed of various frames. Each frame visually represents one or more transformations that occurred to the object during a certain time interval. The displayed frames are the result of a computing system which detects what transformations happened to the object and if they are relevant, with the term "relevant" being the defined by user-introduced parameters for each spatiotemporal feature.

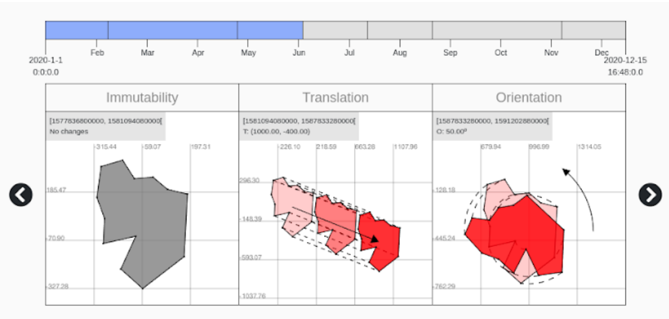
4.1. How many frames does the visual narrative have, in total? *



4.2. How many of the visual narrative frames are visible in this example? *

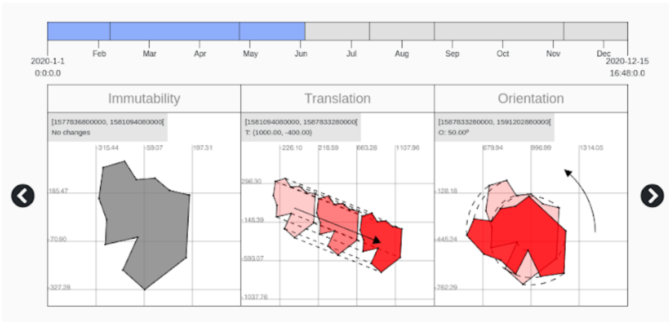


4.3. What is the approximate duration of the complete visual narrative? *



4.4. Select the true statements, regarding the duration of the frames *

Assume the frames are numbered from left to right, starting at 1.



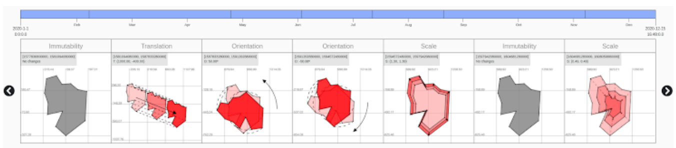
Check all that apply.

- [ ] Frames 1, 3, 4, 5 and 7 have approximately the same duration.
- [ ] Frame 2 starts at the beginning of June.
- [ ] Frame 6 lasts for approximately 78 days.
- [ ] Frame 7 ends at the end of December.
- [ ] The frames that represent the lengthiest time intervals are Frames 2 and 6.
- [ ] The currently visible frames represent less than half of the time length of the narrative.

Other: [ ] ______________________________

4.5. Consider the entire visual narrative. Describe the transformations that occurred to the object, its units, and approximate duration. *

Utilize the nomenclature adopted thus far: translation, scale, orientation change (rotation along the object's centroid), rotation (around a pivot), immutability, and multiple (simultaneous) transformations (identifying which). A better quality image can be found at https://i.imgur.com/urk4BzB.png



____

____

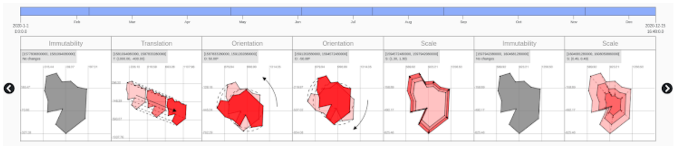
____

____

____

4.6. How much time did it take you to get a "mental image" of the story presented in the visual narrative? *

A better quality image can be found at https://i.imgur.com/urk4BzB.png



Mark only one oval.

- Less than 10 seconds
- Between 10 and 30 seconds
- Between 30 seconds a 1 minute
- Between 1 minute and 2 minutes
- More than two minutes
- I could not understand the evolution of the object

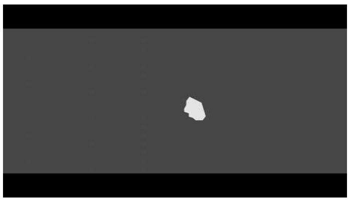
Visual narrative

This section evaluates the effectiveness of the static visual narrative as a means to convey a story based on spatiotemporal data.

As stated, the frames result from a computing system which detects what transformations happened to the object and if they are relevant, with the term "relevant" being the defined by user-introduced parameters for each spatiotemporal feature.

Building on that mechanism, the visualization tool allows users to click on a frame to "loosen" those parameters and show new frames that represent the same time interval as the frame that was clicked and those frames show object transformation with the adjusted parameters. This mechanism is a way of obtaining more details regarding a specific frame.

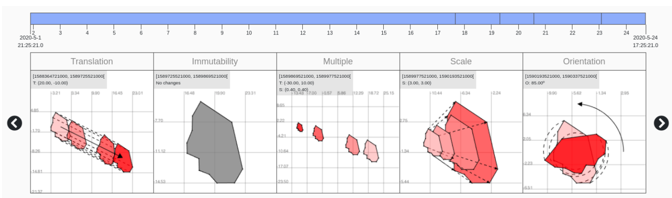
Watch the following video (12s)

 http://youtube.com/watch?v=kDBO0l1rrmw

5.1. Describe the transformations that occurred to the object, based only on the above video. *

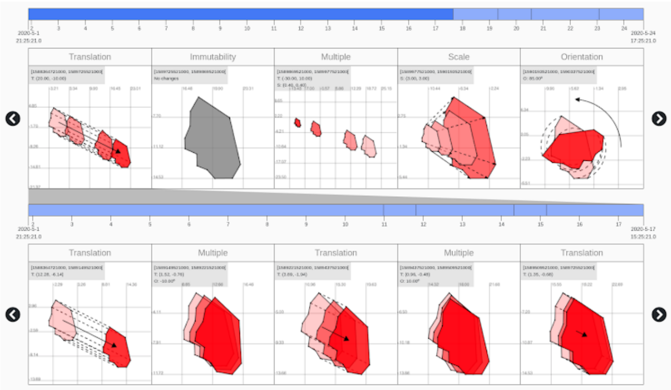
Utilize the nomenclature adopted thus far: translation, scale, orientation change (rotation along the object's centroid), rotation (around a pivot), immutability, and multiple (simultaneous) transformations (identifying which).

Consider the following visual narrative. A better quality image can be found at https://i.imgur.com/C3ujOhs.png



5.2. Consider that the first frame was clicked to obtain more detail regarding the frame's time interval. What are the true statements concerning the object, based on the new narrative row? *

A better quality image can be found at https://i.imgur.com/jQ17Ahp.png



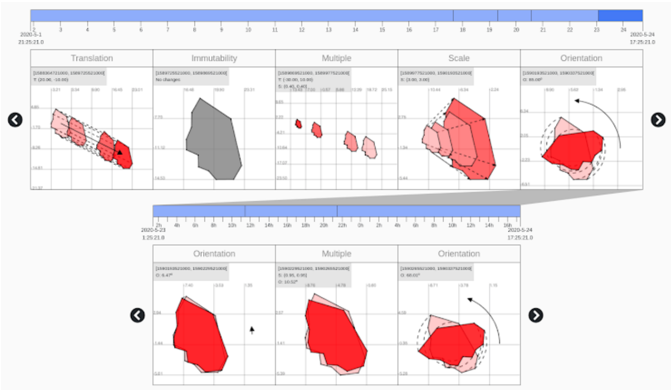
Check all that apply.

- [ ] The object suffered, continuously and without interruptions, a translation.
- [ ] The object suffered only one orientation change.
- [ ] The object suffered various distinct translations or with interruptions.
- [ ] The object suffered two punctual orientation changes.
- [ ] Each time the object suffers an orientation change, the translation is interrupted.
- [ ] Each time the object suffers an orientation change, a translation continues to occur.

Other: [ ] ____________________

5.3. Consider that the fifth frame was clicked to obtain more detail regarding the frame's time interval. What are the true statements concerning the object, based on the new narrative row? *

A better quality image can be found at https://i.imgur.com/IjcopxV.png



Mark only one oval.

- ◯ The object suffered, simultaneously, an orientation change and a scale.
- ◯ The object continuously suffered only one orientation change and, simultaneously during a certain time interval, a scale.
- ◯ The object suffered three orientation changes and, simultaneously during a certain time interval, a scale.
- ◯ Other: ____________________

5.4. Considering that the video represents the same data as the visual narrative, what are the aspects that you consider to be advantages and disadvantages of the visual narrative when compared to the video? *

Consider, non-exclusively, the following comparison criteria: (1) interpretation of the transformations that occurred to the object; (2) time necessary to comprehend the evolution of the phenomenon; (3) video duration (consider what would happen if the video was very long or very short); (4) speed at which object transformations can occur (very rapidly and almost imperceptible to the human eye, or very slowly and consuming a lot of time to visualize); (5) the feature of filtering the content by what is of interest to the user (for example, being able to only see what scales occurred disregarding translations, or filtering by rotations as long as their angle surpasses 20º).

Observations	Thank you very much for your collaboration. In this last section, you can give us your opinion and suggestions regarding the questionnaire and the developed visual narrative tool. For further questions or comments, please contact the author via email up201503822@fe.up.pt.

6.1. Observations about the questionnaire

6.2. Observations about the visualization tool

Your observations can be based on the images and descriptions supplied throughout this questionnaire or, if you prefer, the tool is available to interact at http://fctmost.inesctec.pt:8080/visualization/1

B.2 Results

This section presents complementary questionnaire result graphs. The raw questionnaire answers can be consulted at https://drive.google.com/file/d/1mOYBB5Tm9ivOrVID8AnF_cVX-rIphrM5.

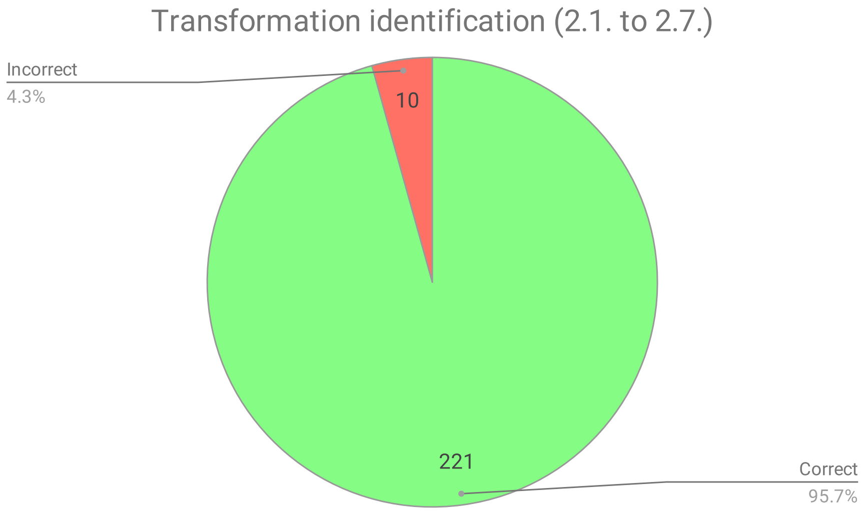


Figure B.1: Aggregated results from questions 2.1. to 2.7.

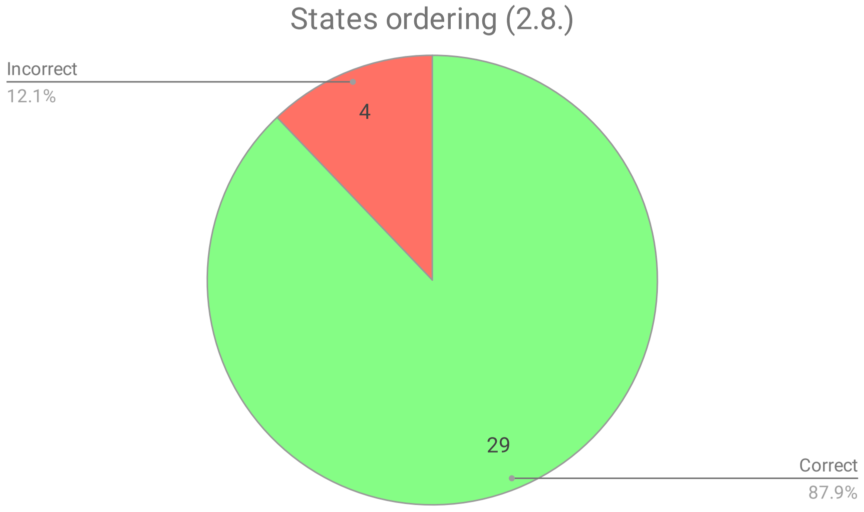


Figure B.2: Results from question 2.8.

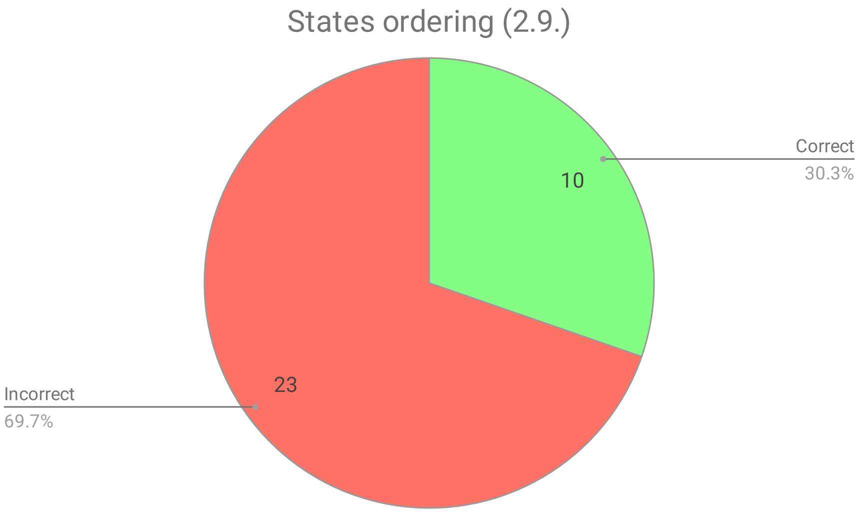


Figure B.3: Results from question 2.9.

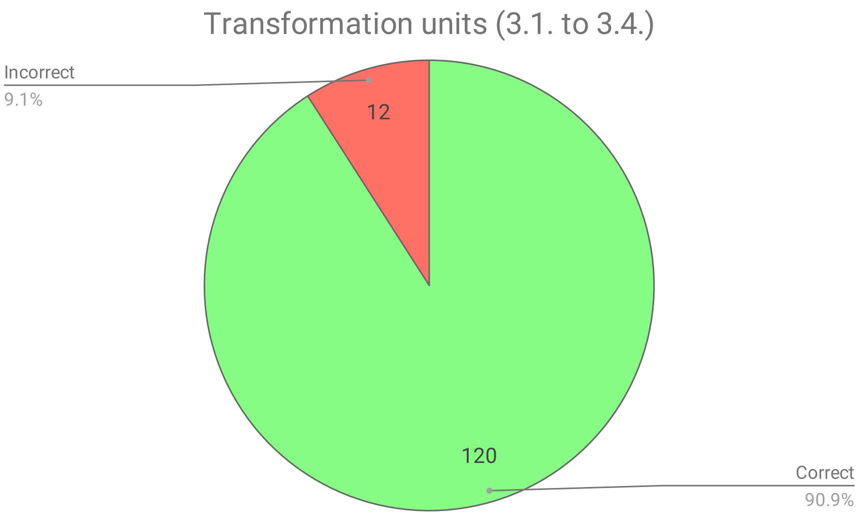


Figure B.4: Aggregated results from questions 3.1. to 3.4.

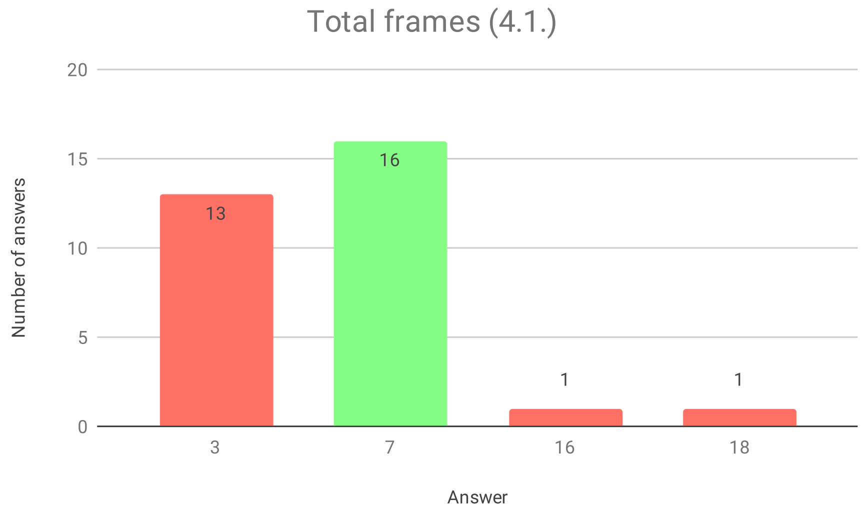


Figure B.5: Results from question 4.1.

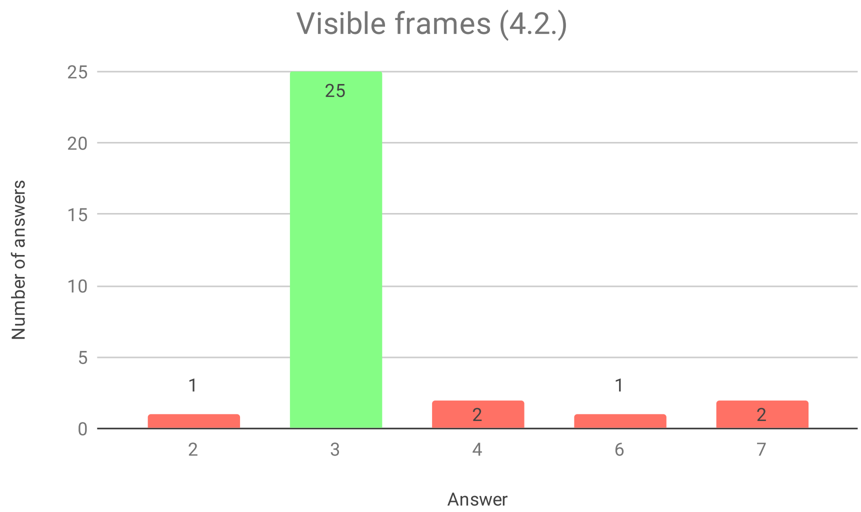


Figure B.6: Results from question 4.2.

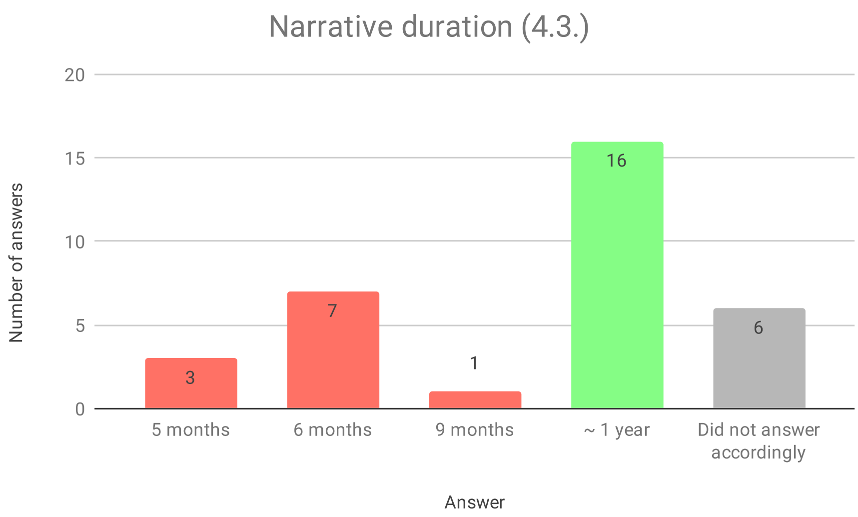


Figure B.7: Results from question 4.3.

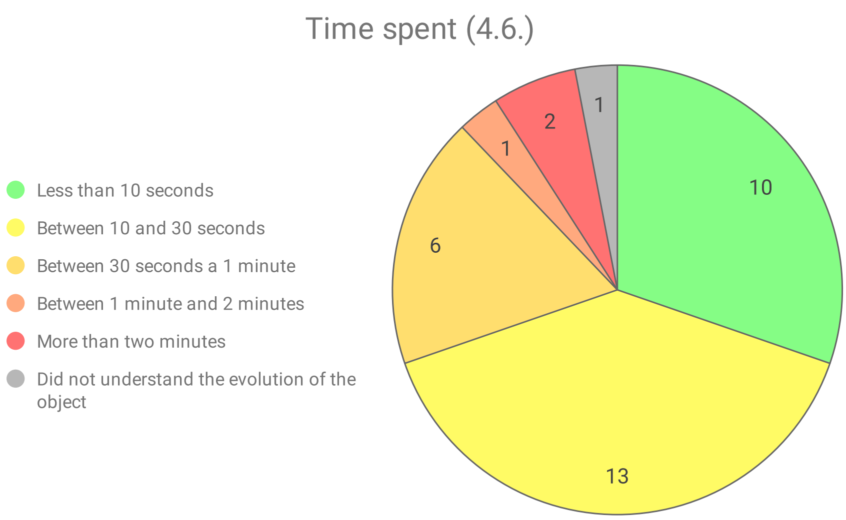


Figure B.8: Results from question 4.6.

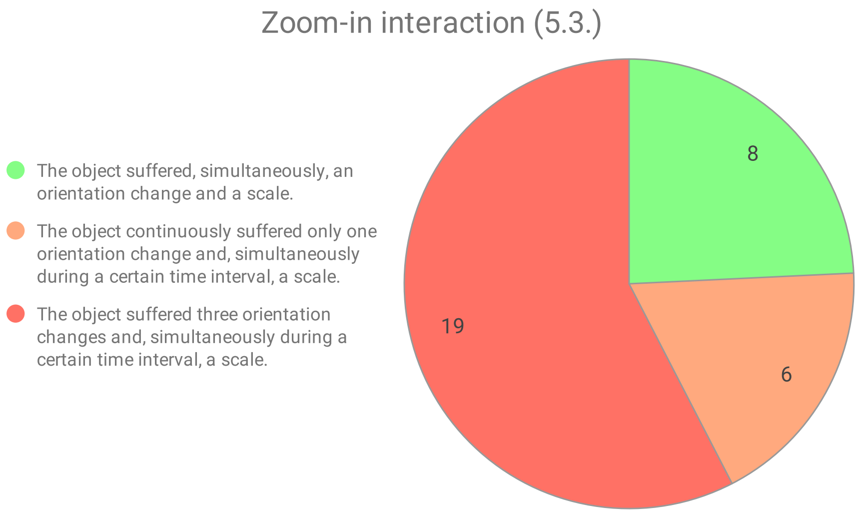


Figure B.9: Results from question 5.3.

References

[1] Robert A. Amar and John T. Stasko. Knowledge precepts for design and evaluation of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):432–442, July 2005.

[2] C. Anderson. Docker [software engineering]. *IEEE Software*, 32(3):102–c3, 2015.

[3] Gennady Andrienko, Natalia Andrienko, Peter Bak, Daniel Keim, Slava Kisilevich, and Stefan Wrobel. A conceptual framework and taxonomy of techniques for analyzing movement. *J. Vis. Lang. Comput.*, 22:213–232, 06 2011.

[4] Matti Anttonen, Arto Salminen, Tommi Mikkonen, and Antero Taivalsaari. Transforming the web into a real application platform: New technologies, emerging trends and missing pieces. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, SAC '11, page 800–807, New York, NY, USA, 2011. Association for Computing Machinery.

[5] J. Bertin. *Semiology of Graphics: Diagrams, Networks, Maps*. ESRI Press, 2011.

[6] Connie Blok. Monitoring change: Characteristics of dynamic geo-spatial phenomena for visual exploration. In *Lecture Notes in Computer Science*, volume 1849, pages 16–30, 01 2000.

[7] Andy Buchanan. Real-time? reframing temporal consciousness in time-based and interactive media. *Technoetic Arts*, 16:53–62, 03 2018.

[8] S.R. Buss. *3D Computer Graphics: A Mathematical Introduction with OpenGL*. Cambridge University Press, 2003.

[9] S. K. Card and J. Mackinlay. The structure of the information visualization design space. In *Proceedings of VIZ '97: Visualization Conference, Information Visualization Symposium and Parallel Rendering Symposium*, pages 92–99, Oct 1997.

[10] Stuart Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization: Using Vision To Think*. Morgan Kaufmann Publishers, 01 1999.

[11] Edgar Carneiro. Representation and quantification of the evolution of spatial temporal phenomena. Unpublished Master Dissertation, 2020.

[12] K.T. Chang. *Introduction to Geographic Information Systems*. McGraw-Hill Higher Education, 2006.

[13] Ruzinoor Che Mat, PhD, rashid shariff, Biswajeet Pradhan, and ahmad rodzi Mahmud. A review on 3d terrain visualization of gis data: Techniques and software. *Geo-spatial Information Science*, 15, 06 2012.

[14] Paul Henry Chombart de Lauwe. *Paris et l'agglomération parisienne*. Presses universitaires de France, 1952.

[15] Andy Cockburn, Amy Karlson, and Benjamin B. Bederson. A review of overview+detail, zooming, and focus+context interfaces. *ACM Comput. Surv.*, 41(1), January 2009.

[16] Joy Deacon. The location of refugia of corylus avellana l. during the weichselian glaciation. *New Phytologist*, 73(5):1055–1063, 1974.

[17] Somayeh Dodge, Robert Weibel, and Anna-Katharina Lautenschütz. Towards a taxonomy of movement patterns. *Information Visualization*, 7(3):240–252, June 2008.

[18] Selan dos Santos and Ken Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3):311 – 325, 2004.

[19] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973.

[20] Elisabeth El Refaie. Understanding visual metaphor: The example of newspaper cartoons. *Visual Communication*, 2, 02 2003.

[21] Steven Feiner and Clifford Beshers. Worlds within worlds: Metaphors for exploring n-dimensional virtual worlds. *Proc. of the 3rd Annual ACM SIGGRAPH Symposium on User Interface Software and Technology*, 09 2000.

[22] Andrew Frank. Different types of "times" in gis. *Spatial and Temporal Reasoning in Geographic Information Systems*, pages 40–62, 01 1998.

[23] Robert Haber and David McNabb. Visualization idioms: A conceptual model for scientific visualization systems. *Visualization in scientific computing*, 74:93, 1990.

[24] Christopher Healey. Perception in visualization. Available at https://www.csc2.ncsu.edu/faculty/healey/PP/, 2008. Accessed in 2020-01-29.

[25] D. Hearn and M.P. Baker. *Computer Graphics, C Version*. Prentice-Hall international editions. Prentice Hall, 1997.

[26] Jeffrey Heer, Michael Bostock, and Vadim Ogievetsky. A tour through the visualization zoo. *ACM Queue*, 8:20, 01 2010.

[27] Jacek Jankowski, Sandy Ressler, Kristian Sons, Yvonne Jung, Johannes Behr, and Philipp Slusallek. Declarative integration of interactive 3d graphics into the world-wide web: Principles, current approaches, and research agenda. In *Proceedings of the 18th International Conference on 3D Web Technology*, Web3D '13, page 39–45, New York, NY, USA, 2013. Association for Computing Machinery.

[28] Daniel E. Kee and Liz Salowitz. Comparing interactive web-based visualization rendering techniques. In *VisWeek 2012*, 2012.

[29] Andy Kirk. *Data Visualisation: A Handbook for Data Driven Design*. SAGE Publications, 2016.

[30] C.N. Knaflic. *Storytelling with Data: A Data Visualization Guide for Business Professionals*. Wiley, 2015.

[31] Ralph Lengler and Martin J. Eppler. Towards a periodic table of visualization methods of management. In *Proceedings of the IASTED International Conference on Graphics and Visualization in Engineering*, GVE '07, page 83–88, USA, 2007. ACTA Press.

[32] A.M. MacEachren. *How Maps Work: Representation, Visualization, and Design*. Guilford Publications, 2004.

[33] Axis Maps. Visual variables. Available at https://www.axismaps.com/guide/general/visual-variables/, 2019. Accessed in 2020-02-05.

[34] S. Marschner and P. Shirley. *Fundamentals of Computer Graphics, Fourth Edition*. Taylor & Francis, 2015.

[35] Scott McCloud. *Understanding Comics: The Invisible Art*. Harper Collins & Kitchen Sink Press, 1994.

[36] S. McKenna, N. Henry Riche, B. Lee, J. Boy, and M. Meyer. Visual narrative flow: Exploring factors shaping data visualization story reading experiences. *Computer Graphics Forum*, 36(3):377–387, June 2017.

[37] T. McReynolds and D. Blythe. *Advanced Graphics Programming Using OpenGL*. The Morgan Kaufmann Series in Computer Graphics. Elsevier Science, 2005.

[38] K.L. Murdock. *3ds Max 2011 Bible*. Bible. Wiley, 2010.

[39] Wolfgang Müller and H. Schumann. Visualization methods for time-dependent data - an overview. In *Proceedings of the 2003 Winter Simulation Conference*, volume 1, pages 737 – 745 Vol.1, 01 2004.

[40] Sherline Pimenta and Ravi Poovaiah. On defining visual narratives. *Design Thoughts*, 3:25–46, 2010.

[41] Catherine Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '04, page 109–116, New York, NY, USA, 2004. Association for Computing Machinery.

[42] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer graphics and image processing*, 1(3):244–256, 1972.

[43] Ruth Rosenholtz, Yuanzhen Li, Jonathan Mansfield, and Zhenlan Jin. Feature congestion: A measure of display clutter. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, page 761–770, New York, NY, USA, 2005. Association for Computing Machinery.

[44] Mark N.K. Saunders, Philip Lewis, and Adrian Thornhill. *Research Methods for Business Students*. Pearson Education Limited, 2015.

[45] E. Segel and J. Heer. Narrative visualization: Telling stories with data. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1139–1148, Nov 2010.

[46] Shashi Shekhar, Zhe Jiang, Reem Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. Spatiotemporal data mining: A computational perspective. *ISPRS International Journal of Geo-Information*, 4:2306–2338, 10 2015.

[47] S. S. Stevens. On the theory of scales of measurement. *Science*, 103(2684):677–680, 1946.

[48] A. Taivalsaari, T. Mikkonen, D. Ingalls, and K. Palacz. Web browser as an application platform. In *2008 34th Euromicro Conference Software Engineering and Advanced Applications*, pages 293–302, 2008.

[49] F. Thomas and O. Johnston. *The illusion of life: Disney animation*. Hyperion, 1995.

[50] Christian Tominski, Petra Schulze-Wollgast, and Heidrun Schumann. 3d information visualization for time dependent data on maps. In *Proceedings of the Ninth International Conference on Information Visualisation*, IV '05, page 175–181, USA, 2005. IEEE Computer Society.

[51] Edward R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, USA, 1986.

[52] Edward R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, USA, 1997.

[53] David Vronay and Shuo Wang. Designing a compelling user interface for morphing. In *Conference on Human Factors in Computing Systems - Proceedings*, pages 143–149, 01 2004.

[54] Colin Ware. *Information Visualization: Perception for Design*. Interactive Technologies. Elsevier Science, 2004.

[55] Martin Wattenberg and Danyel Fisher. Analyzing perceptual organization in information graphics. *Information Visualization*, 3(2):123–133, June 2004.

[56] Leland Wilkinson. *The Grammar of Graphics*. Springer, 2005.

[57] Michael Worboys. Event-oriented approaches to geographic phenomena. *International Journal of Geographical Information Science*, 19:1–28, 01 2005.

[58] Ying Zhu. Measuring effective data visualization. In *Advances in Visual Computing*, pages 652–661, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.