

Explore Cultural Heritage with a Set-based Space-Time Cube

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Software Engineering & Internet Computing

eingereicht von

Bernd Landauer, BSc

Matrikelnummer 0825716

an der Fakultät für Informatik

der Technischen Universität Wien

Betreuung: Univ.Prof. Mag. Dr. Silvia Miksch

Mitwirkung: Dr. Paolo Federico

Wien, 5. März 2018

Bernd Landauer

Silvia Miksch

Explore Cultural Heritage with a Set-based Space-Time Cube

DIPLOMA THESIS

submitted in partial fulfillment of the requirements for the degree of

Diplom-Ingenieur

in

Software Engineering & Internet Computing

by

Bernd Landauer, BSc

Registration Number 0825716

to the Faculty of Informatics

at the TU Wien

Advisor: Univ.Prof. Mag. Dr. Silvia Miksch

Assistance: Dr. Paolo Federico

Vienna, 5th March, 2018

Bernd Landauer

Silvia Miksch

Erklärung zur Verfassung der Arbeit

Bernd Landauer, BSc
Sechshauser Straße 13/24, 1150 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen, Karten und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Wien, 5. März 2018

Bernd Landauer

Danksagung

An dieser Stelle möchte ich all jenen danken, die mich im Rahmen dieser Masterarbeit begleitet haben. Ganz besonders möchte ich meiner Betreuerin Univ.Prof. Silvia Miksch und Dr. Paolo Federico danken, die meine Arbeit durch ihre fachliche und persönliche Unterstützung begleitet haben. Darüber hinaus möchte ich mich bei meinen Eltern Richard und Annemarie bedanken, die mir durch ihre Unterstützung mein Studium ermöglicht haben. Danken möchte ich außerdem meiner Freundin Sabine, die mich mit viel Geduld moralisch unterstützt hat. Ebenso danke ich all meinen Freunden und Kollegen die mir während der Masterarbeit geholfen haben.

Vielen Dank!

Acknowledgements

At this point, I would like to thank all those who have accompanied me in the context of this Master's thesis. I would especially like to thank my advisors Prof. Silvia Miksch and Ph.D. Paolo Federico, for their unfailing support and assistance. In addition, I would like to thank my parents, Richard and Annemarie, for providing me with support and encouragement throughout my years of study. I would also like to thank my girlfriend Sabine, who supported me with patience. Likewise, I thank all my friends and colleagues who helped me during the Master's thesis.

Many Thanks!

Kurzfassung

Seit der Erstellung von digitalen Sammlungen kulturellen Erbes wie Europeana, Trove und Digital Public Library of America (DPLA) wurden viele Kunstobjekte Nutzer_innen über das Internet zugänglich gemacht. Dies gibt den Benutzer_innen die Möglichkeit, Gegenstände zu betrachten, die nicht im Museum ausgestellt sind, entweder weil diese in einem Safe aufbewahrt werden, sich in Restaurierung befinden oder einfach nicht in einer physischen Ausstellung sind. Die Frage, ob die Sammlung für alle Interessierten erreichbar genug ist, bleibt unbeantwortet.

Viele dieser Sammlungen sind schwer zu durchsuchen und zielen auf Expert_innen ab, die nach bestimmten Einträgen suchen. Europeana hat mehr als 50 Millionen Objekte in der digitalen Sammlung, was dieses Thema auch zu einer Big Data Herausforderung macht.

Diese Arbeit soll einen detaillierten Überblick über aktuelle Visualisierungstechniken im Bereich des kulturellen Erbes geben. Durch die Einbeziehung verschiedener mengenbasierter Visualisierungen werden passende Techniken evaluiert, um Kulturerbesammlungen zu visualisieren.

Danach wird eine Anwendung entworfen, die Benutzer_innen hilft durch diese Sammlungen zu navigieren und diese zu erkunden, wobei der zeitliche Aspekt in die Visualisierung einbezogen wird. Für die Einbeziehung des zeitlichen Aspekts greift diese Arbeit auf Einflüsse des Space-Time Cube Konzepts und der mengenbasierten Visualisierung zurück. Die Visualisierung basiert auf dem Polycube Projekt, das es bereits ermöglicht, Standortdaten in den Space-Time Cube zu visualisieren.

Das Ergebnis dieser Arbeit umfasst einen Prototyp, mit dazugehöriger Evaluierung, der mehrere Visualisierungsmethoden beinhaltet, die auf den von Europeana und anderen Sammlungen gesammelten Daten basieren.

Die Evaluierung des Designs und des Prototyps wird durch Benutzerbewertungen und Interviews bereitgestellt. Das Feedback der Anwender_innen, die den Prototyp und die verschiedenen Visualisierungen erkunden, gibt ein Indiz dafür welche Visualisierungsmethode am erfolgreichsten war, um die Benutzer_innen zu unterstützen.

Abstract

Since the creation of digital cultural heritage collections like Europeana, Trove, and Digital Public Library of America (DPLA) many art objects become available via the internet to its users. This gives the users the ability to view items which are not displayed in the museum, either because they are stored in a safe, under reconstruction or simply not in any physical exhibition. The questions if the collection is accessible enough for all users is still unanswered.

Many of those collections are difficult to explore and targeted towards experts looking for specific items. Europeana, for example, has more than 50 million objects in its digital collection, which also makes it a big data challenge.

This work aims to provide a detailed overview of current visualization techniques in the domain of cultural heritage. By including different set-based visualizations in this research, fitting techniques will be evaluated to help visualize culture heritage collections.

Based on the research, an interface will be designed that helps users navigate and explore those collections, with the inclusion of the temporal aspect from the collection data. For the inclusion of the temporal aspect, this work takes influences from the Space-Time Cube concept and set-based visualization. The visualization will be based on the Polycube project, which can already visualize location data into the Space-Time Cube.

The results of this work include a prototype and its evaluation, capable of multiple visualizations, designed based on the data gathered by Europeana and other collections.

Evaluation of the designs and prototype will be provided by user reviews. The feedback of users exploring the prototype and on the different visualizations will be an indicator which visual encoding was most successful in supporting the users.

Contents

Kurzfassung	xi
Abstract	xiii
Contents	xv
1 Introduction	1
1.1 Motivation	1
1.2 Problem Statement	3
1.3 Research Questions	4
1.4 Methodological Approach	4
1.5 Structure of the Thesis	5
2 Related Work	7
2.1 Method	7
2.2 Casual Users & Exploration	8
2.3 Cultural Heritage visualizations	9
2.4 Abstract visualization of sets over time	15
2.5 Summary	21
3 Methodology	23
3.1 Nested Model	23
3.2 Design Triangle	24
4 Design	27
4.1 Design requirements	27
4.2 Analyze the datasets	28
4.3 The First Mockups	30
4.4 Polycube	32
4.5 Set Visualizations	32
4.6 Culture Cube Interface	36
5 Implementation	45
5.1 Technology	45
	xv

5.2	Set Module for Polycube	47
5.3	Culture Cube Interface	53
6	Evaluation	59
6.1	Feedback in the Development Process	59
6.2	Evaluation Method	60
6.3	User Study	62
6.4	Results	62
6.5	Summary	69
7	Conclusion	71
7.1	Conclusion	71
7.2	Future Work	73
	List of Figures	75
	List of Tables	76
	List of Algorithms	77
	Acronyms	79
	Appendix: User Tasks and Questions	81
	Treemap Style	82
	Matrix Style	83
	Rating Questions	84
	Open Questions	84
	Appendix: Answered Tasks	85
	Bibliography	89

Introduction

1.1 Motivation

What is Cultural Heritage (CH)? Heritage is typically something a person inherits from a relative or a family member upon the passing of this person. Now expanding this further, CH is something, we as a community inherit from previous generations. We inherit these objects, add to them and pass them on to the next generations. The focus on CH will grow in the next years, since the European Union launched the “European Year of Cultural Heritage”¹ in 2018. When thinking about CH often artifacts come to mind, these objects are tangible like paintings, sculptures, documents, buildings or archaeological sites. But Cultural Heritage includes much more; intangible elements like folklore, music, traditions, and crafting skills are also part of our heritage. Typically CH can be encountered in a museum, ancient places or performances in theaters and many more. But sometimes an object is not passed down to the next generation. Items can be destroyed through fires, places of importance can disappear through earthquakes, and even traditions can just be forgotten. Not everything is safe from the grips of time, the color of a painting can fade by sunlight itself. Even the attempt of reconstruction can forever destroy a valuable piece of history. With the goal of sharing and saving tangible and intangible objects, multiple digital cultural heritage collections were created in the past years.

The following is a short list of CH collections available via the internet for all users.

- Europeana² — Europe
- Trove³ — Australia
- Digital Public Library of America (DPLA)⁴ — U.S.A.

¹<https://europa.eu/cultural-heritage/>, Last checked Feb 15, 2018

²<http://www.europeana.eu/portal/en>, Last checked Feb 15, 2018

³<http://trove.nla.gov.au/>, Last checked Feb 15, 2018

⁴<https://dp.la/>, Last checked Feb 15, 2018

A digital CH collection is comprised of images, texts, videos, audios or 3D models and their metadata. Additional data about the object, its history, its cultural and historical context can be found in these collections. Historical context revolves around different entities (agents, events, places, time spans, ...) and relations (“has met”, “occurred at”, “was present at”, ...) between them. Collections can be created by various museums like the “Kunsthistorisches Museum” in Vienna or an aggregator like the Europeana project⁵. The project is co-financed by the European Union with the goal to make Europe’s cultural heritage accessible for all. In the paper by [Purday, 2009], the catalyst for creation of Europeana, was an announcement of Google to digitize the printed word with a partnership of US and UK libraries, which also raised the concern of moving public domain resources into the private sector. Additionally, this report specified five user groups the platform targets: general users, school students, academic users, expert researchers and professional users. Europeana provides a detailed data model which is called Europeana Data Model (EDM) to read information from the CH objects [europeana, 2016]. The data can be accessed by browsing the collection itself or applications and tools connected to the digital collection. For accessing the data, Europeana provides multiple Application Programming Interface (API) to search and display items from the collection.

Digital collections have grown over the years, Europeana alone aggregated more than 50 million artworks, artifacts, books, etc. from across Europe. With the number of items in collections, it is challenging to present all of them in a visually engaging way and let users explore. Additionally, the amount of data and the type variety makes it a big data challenge. Having the data is only the first step, one must also be able to interact with it. The most common way to interact with history is in a museum where customers can explore various collections and exhibition, which are carefully crafted by experts.

To let users explore the digital content is must be more than just a list of artworks or an endless wall of pictures on the website.

This challenge can be tackled in multiple ways one of which lies in the research area of Information Visualization (InfoVis). An appropriate visual representation of the collection gives the user overview and useful insight into the data. One way of providing an overview to such collections is achieved by set-based visualization. A set is defined as a collection of unique objects, and these sets can be based on the attributes of the objects. In the case of this work, the sets can be created based on the material, categories (pictures, sculptures, etc.) or epochs of the artifacts. Sets have the potential to overlap and form relations with other sets. Movie genres are another example for sets. The movie “Toy Story” is at the intersection between the “Animation” and “Comedy” sets/genres. Aggregating artifacts into sets and visualizing them could lead to a useful overview of the CH collection.

Since time is the most important information in cultural data, it should always be visible. This work focuses on a set-based visualization of a cultural heritage collection with the

⁵<http://europeana.eu> project, Last checked Nov 20, 2017

temporal aspect of their items.

The goal is to implement a prototype based on data gathered by museums or digital cultural heritage collections which helps users navigate within the collection.

As part of this work, the visualization will be integrated into the Polycube project, which already provides the possibility to visualize location data in a Space-Time Cube (STC). The integration will be a module for the polycube project focusing on visualization of sets within the STC. More details on the concept of this STC will follow in the related work part of this thesis.

A small subset of the “Kunsthistorisches Museum” in Vienna will be used as test dataset for the prototype. The prototype will be evaluated by user reviews and interviews. Based on the collected insight it can be determined if the created visualization is useful for explorers. The work will take a close look at the solution proposed in [Windhager et al., 2016a], in detail how to integrate the temporal aspect of data into a set-based diagram. This solution focuses on the design of a three-dimensional Space-Time Cube in combination with CH data to enable a visualization that covers all five design requirements for casual users in [Mayr et al., 2016].

In the work by [Walsh and Hall, 2015, p. 1] defines casual users as “These casual users frequently have no goal or topic in mind, but just want to have a look around what is available in the collection”. The users just want to have a look around and see what is available in the collection.

To target, a large group of users the prototype is developed as a web application.

1.2 Problem Statement

The current structure of CH data and interfaces is designed for researchers and users with a specific goal in mind. It is possible to find items when knowing what to look for. The first problem one encounters is the lack of overview in these collections, which stands in contrast to the “Overview first” point in the “Information-Seeking Mantra” by [Shneiderman, 1996]. A casual user or the general public access these collections without a goal and thereby only with the goal of exploration or entertainment. It is difficult to interact with the data in a search box way [Whitelaw, 2015]. Sometimes these search interfaces resemble more work than entertainment which most users seek. No guidance leads to frustration of the users and they end their exploration. Based on a survey that would lead to a loss of 21% of online traffic⁶. Because of the lack of overview in such collections, a set-based visualization could be a great fit [Alsallakh et al., 2014].

Another aspect that needs consideration, especially when dealing with cultural data, is the temporal information, “when did this happen?”. Many different types of data contain temporal information — from financial data to medical data to simple photography. Even

⁶<http://www.intk.com/en/ideas/visitor-motivation-in-online-museum-audiences>, Last checked Nov 20, 2017

a pie-chart with visible time encoding still shows a snapshot of a given point in time where the data was collected and visualized. Depending on the data and specific characteristics a dedicated visualization is chosen. More about various ways to visualize time and time-oriented data can be found in the work by [Aigner et al., 2011, chapter 4 & 7]. Also the article “It’s About Time”⁷ highlights, in multiple examples, how visualizations can help reveal patterns or simply tell stories. Based on this knowledge, the report in chapter 2 evaluates if a visualization can communicate the information of time or can be modified to do so.

1.3 Research Questions

The primary contribution of this research is a new visualization that is based on set-typed visualization and extending them with temporal information. The resulting visualization focuses on the needs of casual users and thereby should be understandable without a big learning curve.

- *How can an interactive visualization of cultural heritage collection support exploratory behaviors of casual users?*

This question is concerned with the creation of an interactive visualization with cultural heritage data. The focus is targeted towards casual users; therefore specific requirements have to be fulfilled. To answer these questions multiple visualization methods within a prototype will be designed and developed.

The main research question can be split into three detailed questions:

- *How can a set-based visualization be enhanced by the temporal aspect of data?*
What is an appropriate and intuitive way of communicating time in a set-based visualization without removing other information?
- *How can a set-based visualization be represented in a Space-Time Cube?*
Can a Space-Time Cube be used to communicate set-based information in a meaningful way?
- *How can a set-based visualization in combination with a temporal aspect be applied in cultural heritage collection?*
If a set-based visualization that works in a Space-Time Cube is chosen, does it work for a digital cultural heritage collection?

1.4 Methodological Approach

The methodological approach includes following steps:

⁷<https://uxmag.com/articles/its-about-time>, Last checked Nov 20, 2017

Research

- State of the Art research.
- Analyze various methods to visualize cultural heritage collections.
- Analyze various solutions of set-based visualizations, especially sets over time.

Design

- Evaluation of available technologies for the implementation.
- Creating interface mockups during the design process.
- This phase will focus on the simplicity of the visualization.

Prototype Development

- A prototype will implement many various visualizations based on the data from cultural heritage collections.
- The prototype will be developed as a client-side web application, with new web technologies Hypertext Markup Language (HTML5) and Web Graphics Library (WebGL).
- With the prototype, the casual user should have control over their way to explore the data.
- The prototype can be configured and visualizes JavaScript Object Notation (JSON) data by specifying the “time” information field and the “set” information field of data.

Evaluation

- The implemented prototype will be used as the main evaluation method of this work.
- The evaluation will be based on qualitative user feedback for usability and intuitiveness of the visualization.
- Potential issues with the prototype or design will be examined.

1.5 Structure of the Thesis

The structure of the thesis is as follows:

Chapter 2 Related Work

This chapter provides an overview of work related to this thesis. The first part focuses on requirements of information visualization for casual users. Further visualization techniques and work will be split into two categories one with the focus on visualization for cultural heritage. The other part focuses on set-based visualization over time. Presented techniques will be analyzed to fit the needs of the casual users.

Chapter 3 Methodology

This chapter contains the research methodology used to guide the design and development process of the prototype.

Chapter 4 Design

The chapter begins with the preparation of the data and a short description of tools used to analyze the information. Design requirements of the implemented visualization and the final user interface design with all its components will be introduced in this chapter. In addition to the design of the most important features of the prototype, the algorithm for stable Treemap visualization in the Space-Time Cube and layout techniques for the various visualizations will be discussed. Moreover, main interaction possibilities will be explained in detail.

Chapter 5 Implementation

This chapter contains a brief overview of the technologies and frameworks used in the prototype, followed by a detailed description of the project structure and technical documentation of the visual prototype.

Chapter 6 Evaluation

This chapter contains the evaluation of the prototype and the proposed visualization types. Which one is the most intuitive for the users will be discovered through interviews. The results of the interviews and answered questions will be listed here.

Chapter 7 Conclusion

This chapter provides a summary of the results gathered in the evaluation and discusses them in detail. Answers to the research questions will be provided in this chapter. Followed by some ideas for future improvements and features that have been collected over the development process and in the evaluation phase.

Related Work

This chapter contains an overview of research on casual user interfaces, visualization techniques of cultural heritage data and techniques to visualize set-based data. The method section describes how the literature research for the related work was performed. The second part of this chapter is looking at the requirements to assist users in exploring a new visualization. Since the visualization in this work is targeted towards casual users the resulting design must be usable, have intuitive controls and a soft learning curve. Section three and four are taking a more detailed look at the fields of cultural heritage visualization and set-based visualization.

The chapter concludes with a summary and findings.

2.1 Method

Since this work incorporates three different fields of study. The research was split into three different categories, research in the field of casual users, cultural heritage visualization and set-based visualization. The literature research was mainly performed with the use of Google scholar, a search engine for scholarly literature. The search engine is connected to multiple digital libraries containing scientific papers, books, and other works. Additional references were discovered through searching the *ACM*, *IEEE Xplore* and *CatalogPlus* digital libraries. Performing the search with these tools makes it possible to look at references easily. While searching through the results, newer entries in each area were preferred.

In addition to the literature research, the same categories were used to perform a search on real application through the regular Google search engine.

Casual Users — Search in this area included the keywords *casual user visualization*, *casual user exploration*, *user exploration*, *casual users cultural heritage*.

Cultural Heritage — This search was started by using keywords like *cultural heritage*, *cultural heritage visualization*, *cultural heritage interfaces*, *cultural heritage collections*. Followed by searching the referenced work of the found papers. Research that focuses on displaying an overview on a collection or museums were included. On the other hand, work that focuses on techniques displaying single artifacts were excluded, like recreating a three dimensional model of a statue from images.

Set-based — This search was started with recent state of the report by [Alsallakh et al., 2014] followed by searching through the referenced work and search queries like *set-based visualization*, *collection visualization*, *set based visualization over time*.

2.2 Casual Users & Exploration

Casual Users in Information Visualization

To create a visualization targeted towards casual users and aimed at supporting their exploratory behavior, a couple of papers have been published. These papers focus on the needs of the users and recommended features an interface should have. Casual users search with no specific goal in mind or exploring a collection in search of entertainment. By exploring the behavior of casual users and how they interact with digital cultural heritage, five design requirements can be extracted [Mayr et al., 2016].

Those requirements are:

- **Providing Conceptual Orientation** — Before starting to search it is important to get an overview, like in a real museum where visitors get the information at the entrance. A map of the museum helps the visitors with their orientation.
- **Supporting Exploratory Search** — An interface has to support gaining overview, browsing through the topics and engaging with details.
- **Increasing User Experience** — Users should be provided with an entertaining learning experience that is as inviting, inspiring and informative as possible.
- **Keeping Cognitive Load Low** — The Visualization is easily understood, and users can orient themselves without much effort, enough cognitive resources are available to discover interesting things.
- **Enabling Intuitive Modes of Interaction** — Interactions should be simple and intuitive. If possible the modes of interaction should also be based on the prior knowledge of the users.

In addition to these requirements, one must not forget the “Information-Seeking Mantra” by [Shneiderman, 1996]. Any visualization should provide **overview first**, give the users option to **zoom and filter** the visual elements based on their needs. Later the users should have the possibility to request more **details on demand**.

How to arrange CH collections to offer the users the same exploratory possibilities as in a museum is a big challenge. Design patterns for interfaces range from close-ups, collection overviews to slideshows and grids. Those patterns are highlighted in [Windhager et al., 2016a] and some of the patterns are used in the following visualizations in this chapter.

From all the information three design recommendations for collection visualization have been introduced by [Windhager et al., 2016b] to help the casual user:

- **Orientation phase** — InfoVis interfaces should provide effective overviews and conceptual orientation of a collections extension, its major components, and arrangement.
- **Only limited cognitive resources** — InfoVis interfaces should offer a maximum of overview while keeping the cognitive load, required for cognitive information integration and orientation low [Dörk et al., 2011].
- **Support exploratory behavior** — New interfaces should support exploratory behavior, e.g., by offering multiple perspectives and a rich set of interaction methods to explore object collections on overview and detail levels.

Keep it simple

“Keep it simple” is not only crucial for developers but also for the resulting product. But what is the right level of complexity? In the research on the effect of different set-based visualizations by [Verbert et al., 2014] it is shown that it is vital to keep the visualization as simple as possible to trigger the exploratory behavior of users. Too much information at once can overwhelm the users. Venn diagrams are more intuitive than the cluster map visualization when exploring intersections of sets, especially if the users are in a free exploration setting without defined goals.

2.3 Cultural Heritage visualizations

There are many different ways to search and explore cultural heritage collections online. Starting from the simple website of local museums like the Kunsthistorisches Museum Wien (KHM) in Vienna to aggregated collectors like the Europeana project. In some cases, those websites already curated small parts of the collection to be browsed through. But the most common way to search a CH collection is by entering keywords in an input field and pressing a search button. A list of objects will appear as a result of this action.

This experience does not compare to a regular museum visit where users search a physical collection by walking. In a paper by [Whitelaw, 2015], he argues that this search box method is inadequate for the exploring user.

By changing the initial interaction with the collection, removing the “information need” aspect and opening with an overview of the collection itself a better start for exploration

is created [Walsh and Hall, 2015]. As a first step, this could be a simple tag-cloud with keywords or a display of an “interest map” resembling a physical map. Besides displaying a list of available collections like in Rijks Studio¹ or with the collection page of KHM² can help users.

As mentioned before, another common way apart from the regular search box interface is displaying a specific part of the overall collection in a separate area on the website. This approach gives the option to focus on a specific topic, for example, an artifact collection from early Egypt. The subset is typically displayed as part of the CH collection website³. Often such collections are aggregated into a separate website for exhibitions. As an example, Inpersepolis.org⁴ displays various information about the first Persian empire. Another example, “Inventing Abstraction”⁵ which explores the connections between artists.

Now to highlight some websites and interactive visualization in the area of cultural heritage and history. Some tools encourage the users to explore and other tools have a stronger focus on expert users.

MKG Hamburg Collection online

<http://sammlungonline.mkg-hamburg.de/de/search>

The “Museum für Kunst und Gewerbe” in Hamburg website contains a section that makes the collection available on the internet. As shown in figure 2.1a, the presentation of the data is very simple, and the main interaction happens with a search box. On the left side of the page it is possible to browse through the certain grouped properties of the objects within the collection, e.g., artists, locations, and so on. The result is displayed in images in the center of the page and is sortable by “relevance” or chronologically. Also, users can create their own collections and save them in their account on the website. All those features make it a usable and intuitive tool to browse through the collection.

Two small issues with this way of presenting the data are visible. Users never have a visual overview of the overall collection. Also, the time is not considered as a separate property, but epochs are selectable through a drop-down.

Google Arts & Culture

<http://www.google.com/culturalinstitute/beta>

Another example of such an interface was designed by Google as part of their Cultural Institute⁶, where collections can be browsed by artists, facts, locations or other topics.

¹<https://www.rijksmuseum.nl/en/rijksstudio>, Last checked Nov 20, 2017

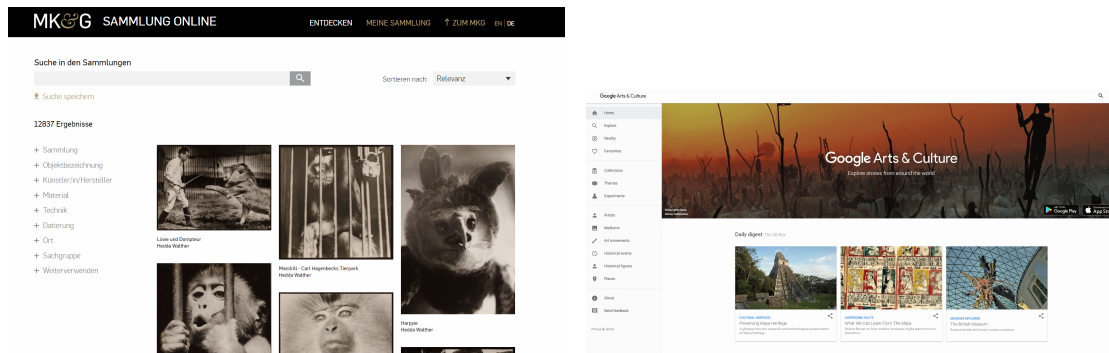
²<http://www.khm.at/besuchen/sammlungen/>, Last checked Nov 20, 2017

³<http://www.europeana.eu/portal/en>, Last checked Nov 20, 2017

⁴<http://www.inpersepolis.org/>, Last checked Nov 20, 2017

⁵<https://www.moma.org/interactives/exhibitions/2012/inventingabstraction/>, Last checked Nov 20, 2017

⁶<http://www.google.com/culturalinstitute/beta/>, Last checked Nov 20, 2017



(a) MKG Hamburg Collection online interface for search objects. (b) Google Arts & Culture website displaying various collections, themes, and locations.

Figure 2.1: MKG Collection & Google Arts & Culture

The overall User Interface (UI) is kept clean and simple but does not provide a complete overview of the whole collection that is available, see figure 2.1b. Museums and location can even be explored with a virtual tour through the location. This website is very similar to other museum websites. Under “Experiments” various interesting visualizations can be found, all of those experiments contain high definition images of artworks from various collections.

“Freefall” displays a collection of images without particular order in various views⁷. With “The Curator Table”⁸, users have the option to see the data on a virtual surface and filtered by a search box. Initially, users are presented with a visual overview of the collection, based on the color of the artworks. After a search for an artist or other criteria, the items are pulled from the table and presented in a separate space. Here the options for displaying the items as grid or timeline become available for this subset. The interactiveness of this experiment encourages users to explore. “T-SNE Map”⁹ is a visualization created with machine learning and creating a 3D landscape based on the similarity of the images. Here the users choose a guided tour or explore freely.

The focus of these experiments is not the temporal information, but rather the number of objects and possibility to examine every object in high detail.

Wikigalaxy

<http://wiki.polyfra.me/>

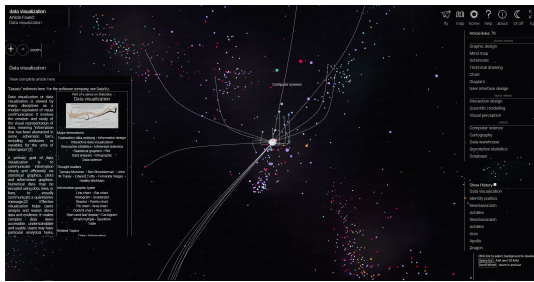
When it comes to data, Wikipedia is a collection of all kind of information also cultural data. Based on this data many visualizations have been created, which gives users a unique way of exploring this collection. “Wikigalaxy” is a visualization that displays the

⁷<https://artsexperiments.withgoogle.com/freefall/timeline>, Last checked Nov 20, 2017

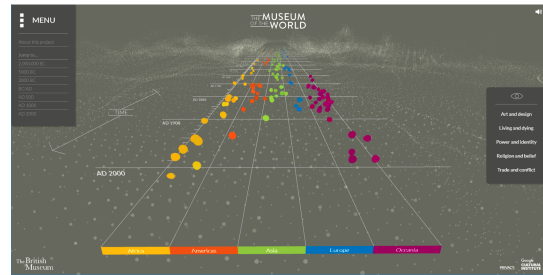
⁸<https://artsexperiments.withgoogle.com/curatortable/>, Last checked Nov 20, 2017

⁹<https://artsexperiments.withgoogle.com/tsnemap/>, Last checked Nov 20, 2017

2. RELATED WORK



(a) Wikigalaxy shows a three-dimensional visualization of Wikipedias content and connections.



(b) Museum of the World visualizes events and objects along a timeline.

Figure 2.2: Wikigalaxy & Museum of the World

collection through the metaphor of a galaxy which triggers the fun in exploration and gives a feeling for the scale of this huge collection. Each star in figure 2.2a represents a Wikipedia article, and when selecting it, users see links to related articles and routes to those in the galaxy. Users can explore the galaxy with a search box in the top left corner. For more visualizations, Hatnote¹⁰ curated a list of visualizations based on the information collected on Wikipedia¹¹.

The galaxy does not contain a way to interact with the time information in the articles.

The Museum of the World

<https://britishmuseum.withgoogle.com/>

“The Museum of the World” is a project partnership with the **British Museum** and **Google Cultural Institute**. The goal is to explore objects from prehistory to the present day along a timeline. Jump through time, explore objects and listen to curators sharing their insights. It is possible to filter objects based on a certain category and through clicking on objects users uncover connections between objects. The objects are split by continents, visible on the bottom of the timeline. The user can traverse time either through pressing the arrow button on the keyboard, scrolling movement with the mouse or by clicking on predefined spots in time in the menu. And all while listening to relaxing music.

As shown in figure 2.2b, the main focus is the visualization of the timeline.

Histography

<http://histography.io/>

¹⁰Hatnote is a volunteer-run design studio, organized around Wikipedia as a social and data platform.

¹¹<http://seealso.org/>, Last checked Nov 20, 2017

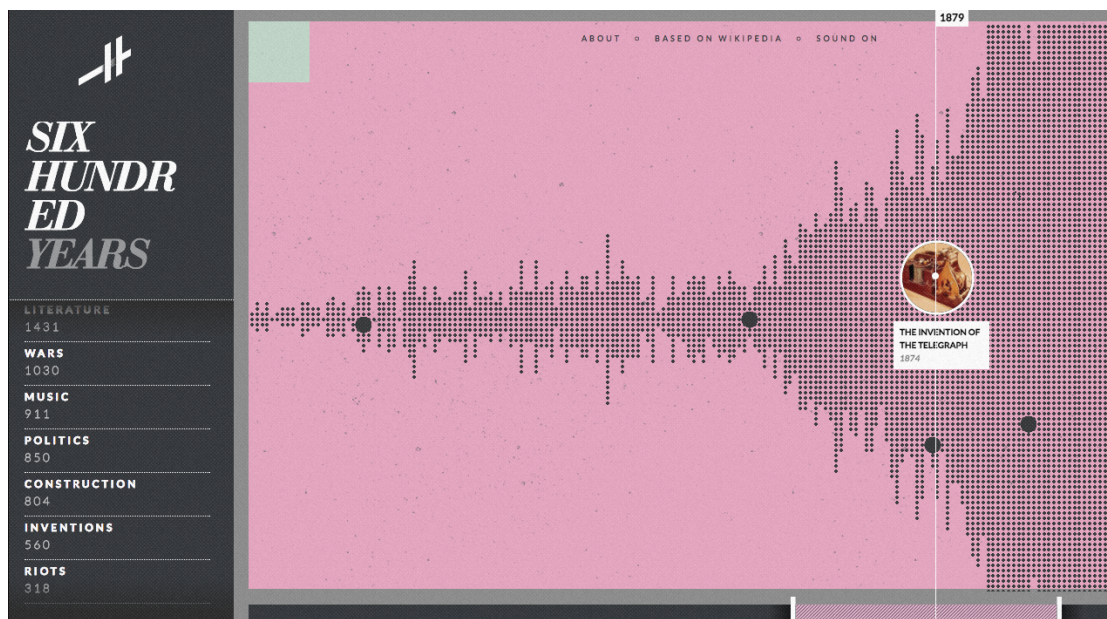


Figure 2.3: Histography by Matan Stauber, displays events as little dots on a timeline, which can be manipulated. Certain categories of events can be selected on the left side of the interface.

One of the most visually appealing visualizations is Histography created by Matan Stauber, which is also built around the data of Wikipedia, shown in figure 2.3. Visualizing dots on a timeline, every dot represents a historic event from Wikipedia. Users can move and resize the bottom bar to view any time period. Upon clicking on one of the dots, an image is displayed and it is possible to highlight related events or to read the Wikipedia article for more details. Some of the dots even have a video linked which is watchable from the website. In addition, it is possible to filter the dots based on the type of event, e.g., “Music”, “Wars”, “Discoveries”, and so on. Upon selection, only the dots representing this type of event are visible in the timeline. By selecting another type of event, one can compare both sets, the second set of dots is visible in different color.

Histography is a simple and intuitive view of historical events with an easy to navigate timeline.

German digital library visualized

<https://uclab.fh-potsdam.de/ddb>

The work by [Dörk et al., 2017] introduces a visualization “Deutsche Digitale Bibliothek visualisiert” containing multiple views on the same digital collection, each of those with a different focus. These views contain information about places, networks and a word cloud based on keywords of the collection. For this work, the most interesting view is the “Periods & Sectors”, which focuses on the overview of the collection, shown in figure 2.4.

2. RELATED WORK



Figure 2.4: Deutsche Digitale Bibliothek visualized, shows the content of a digital collection in multiple visualizations on one page. Word-clouds, line and bar charts are all visible.

The view aggregates its data from different sectors which are represented in line-charts. On the bottom, the visualization has a timeline, which the user can interact with to select a time period. On selecting a time period the user gains insight into the data inside this period. Below the actual chart, there are three word-clouds for keywords, places, and people in this time period. On the right side of the line charts, there are horizontal bar charts representing the distribution of the artifacts by the sectors and the media type of the items. By selecting one of the bars the user will open a new browser tab where it is possible to explore the items the user selected, with additional fine grain options like filters. This feature is functional, but since it comes without warning, it rips the users out of the experience.

Also in this example, we see the focus on time and providing an overview of the collection.

Past Visions

<https://uclab.fh-potsdam.de/fw4/en/vis>

In the work by [Glinka et al., 2017] a visualization is presented that takes the drawings by Frederick William IV and shows them in a timeline. The user can zoom to all the individual drawings, which have high-definition quality. It is also possible to set one or more filters by selecting tags on the top of the website, see figure 2.5a. On setting a filter, the items that match the criteria will stay visible. All the other not matched images will move below the timeline and get slightly transparent.



(a) Past Visions shows various drawing by Frederick William IV, organized similar to a histogram.

(b) Library Observatory shows the content of a digital collection as a treemap.

Figure 2.5: Past Visions & Library Observatory

Library Observatory

<http://www.libraryobservatory.org/>

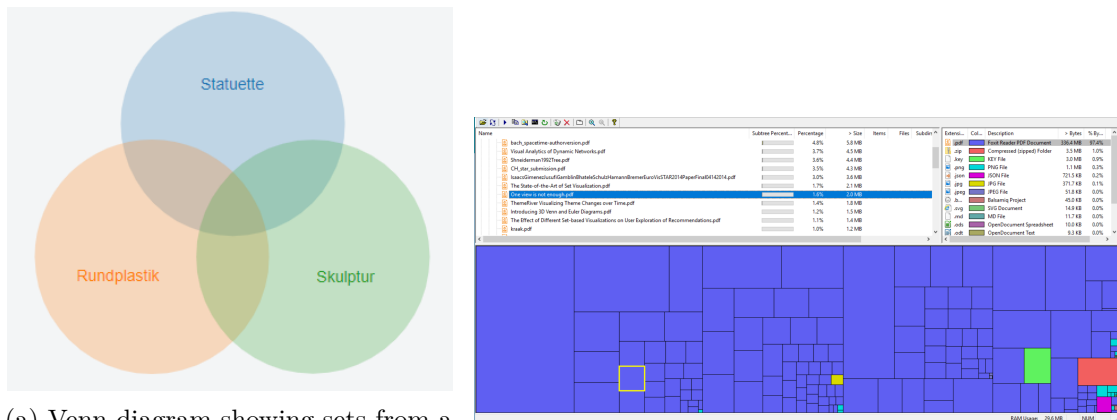
“Library Observatory” uses a Treemap to visualize the various items from the DPLA, shown in figure 2.5b. A treemap is used to visualize a hierarchical structure of some data. In this application, it is used to guide users through a selection. The selection is always in a predefined order from Provider, Collection, Format, Objects to Record. First, the users select the Provider, based on this a treemap is displayed with all collections from this provider and so on. In the later selection steps, the items are displayed as a wall of images and textual information. This method enables the users to get a fast overview of the total content and the interaction is very intuitive. The drawback is that there is no notion of the temporal aspect, only after selecting a single object the date becomes visible.

2.4 Abstract visualization of sets over time

As explained in the state of the art report [Alsallakh et al., 2014] both matrix-based and aggregation-based methods have fair to high scalability with the number of sets, which can help with displaying the amount of data within a CH collection.

The aggregation-based methods move away from the representation of single elements and aggregate multiple items into a single visual element. This provides the users with overview first and allows the exploration of these elements on demand [Shneiderman, 1996]. This approach is useful if the goal is to present users with a large number of elements.

In matrix-based methods, rows and columns can represent items or sets. The value in each cell shows the relationship between the row and column. This method is intuitive for many users who are already familiar with tables and excel sheets.



(a) Venn diagram showing sets from a CH collection and intersection points. The size of the circle is related to the number of items within the set. (b) Treemap created with WinDirStat shows a hierarchical directory structure. On top of the treemap, a regular file browser is present.

Figure 2.6: Venn & Treemap

Time is vital information for cultural heritage, so this work tries to evaluate each set visualization on the possibility of adding temporal information.

Set-based visualization is a great way to provide more insight how a complex collection is structured. Another aspect of sets is to analyse the relations and connections between those sets [Alsallakh et al., 2014].

This section takes a look at three visualization techniques followed by four applications.

Euler and Venn Diagrams

Euler and Venn diagrams are common ways to present sets and overlapping sets, see the example in figure 2.6a. Also, this method is very intuitive to casual users as explored in the work by [Verbert et al., 2014]. For displaying an overview of a collection this would be a natural way of representing objects grouped by criteria. One item is only located in only one area of this chart and items can be visible as dots, images or just the number of elements in this area. The drawback of this method is the limited scalability when multiple sets are overlapping. The view gets more crowded with items overlapping elements and more difficult to read. Usually, Venn diagrams are readable if the number of overlapping sets does not exceed four.

Venn diagrams do not communicate the information of time. One technique for adding temporal information is called small multiples by [Tufte, 1990]. Multiple small Venn diagrams can be presented in one view, each of them displaying another point in time. Changes of time can also be visualized as an animation, implemented as play button and display shifts in the collection through the point in time.

Treemap

Introduced in the work by [Shneiderman, 1992], a treemap is a method for displaying hierarchical data using nested rectangles combined with a space-filling approach. This method is often used to show how something is divided up and thereby showing “part of a whole”. One scenario where a treemap is commonly used is visualizing a directory structure on computers to find which file or directory uses the most disk space. The rectangles are sized by the amount of disk space they use, one example of this is shown in figure 2.6b.

If the collection or the selection criteria make it possible to have no intersection relations between the elements, it is possible to choose a treemap to visualize the collection.

By using a stable treemap algorithm, like the one proposed in the work by [Sondag et al., 2018], it is possible to visualize changes over time. By stacking the treemaps from different points in time, a three-dimensional visualization is created showing changes over time. With this method the rectangles of the treemap stay in their original position and make it easier for users to compare different time periods. Also with small multiples or animation, it is possible to visualize changes over time.

Heat map

A heat map is a simple matrix-based visualization which examines data through coloring of the cells. Typically, rows represent a category (e.g. city names), the columns are another category (in this case fruit) as in the figure 2.7a which shows fruit consumption per city. The cells then contain the color-coded information of the relationship by the matching row and column.

As mentioned before this matrix-based approach has the benefit of being easy to understand because many things in our daily life are structured in tables or matrices, e.g., timetables on bus stations.

Some drawback of this method is the reliance on color as the main source of information and for the color-coding a legend is required.

Replacing data from a row or column with time data is no option since we lose other information. An animation could work well in this scenario, as with small multiples if the number of row or columns is kept small. Stacking multiple heat maps on each other would work, creating a three-dimensional cube.

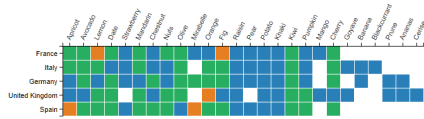
Radial Sets

<http://www.cvast.tuwien.ac.at/radialsets>

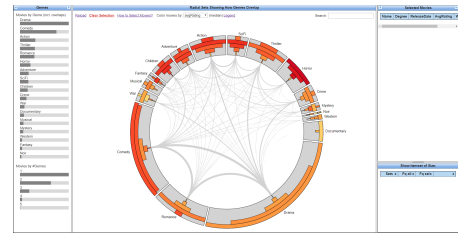
Radial Sets by [Alsallakh et al., 2013] introduces an interactive visual analysis tool for large overlapping sets. This aggregation-based tool is targeted to expert users. Radial

¹²<https://bl.ocks.org/B13f/cdb5ad854b376765fa99>, Last checked Nov 20, 2017

¹³<http://www.cvast.tuwien.ac.at/radialsets>, Last checked Feb 15, 2018



(a) Heatmap sample from the D3.js website¹². It shows the fruit consumption per city.



(b) Main View of Radial Sets depicting movies with the genres as sets¹³.

Figure 2.7: Heatmap & Radial Sets

Sets scale very well with the number of elements, but are limited to the number of sets (20-30) they can represent.

The sets are organized in a circular layout. Each set contains a histogram representing the elements in each set. The sets then are linked with arcs to other sets which represent the overlaps between those sets. Histograms within each set are grouped by degree, which is the number of how many sets the element contains. For example, a movie with only one genre has degree 1, a movie with two genres is classified as degree 2, and so on. In addition to the main view in figure 2.7b, Radial Sets contains a summary view and a detail view to show more information about a selection.

It is possible to create a selection based on various actions, by clicking on sets, bars or arcs. The elements belonging to this selection are then highlighted with red color on the visual elements. The detail view shows the selected items within a table. Upon hovering over the visual elements it shows how much elements are contained.

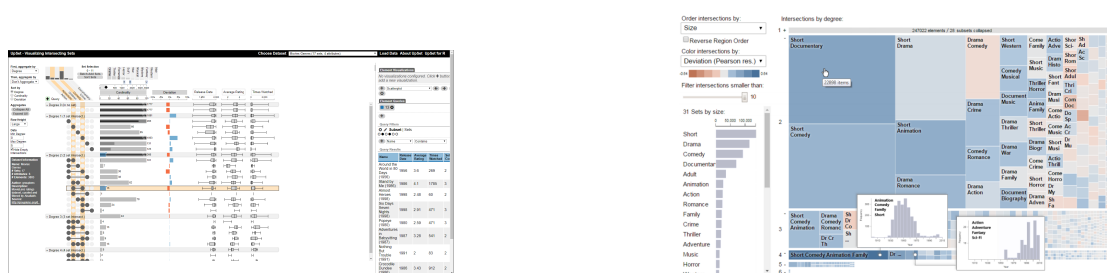
This technique contains temporal information in the data of the objects but that information is not visible in the main view. Small multiples and stacking the multiples is probably not an option because of the number of elements. An animation can still be a viable option to present changes over time.

UpSet & PowerSet

<https://vcg.github.io/upset/> UpSet

<https://www.ncbi.nlm.nih.gov/pubmed/27875152> Powerset

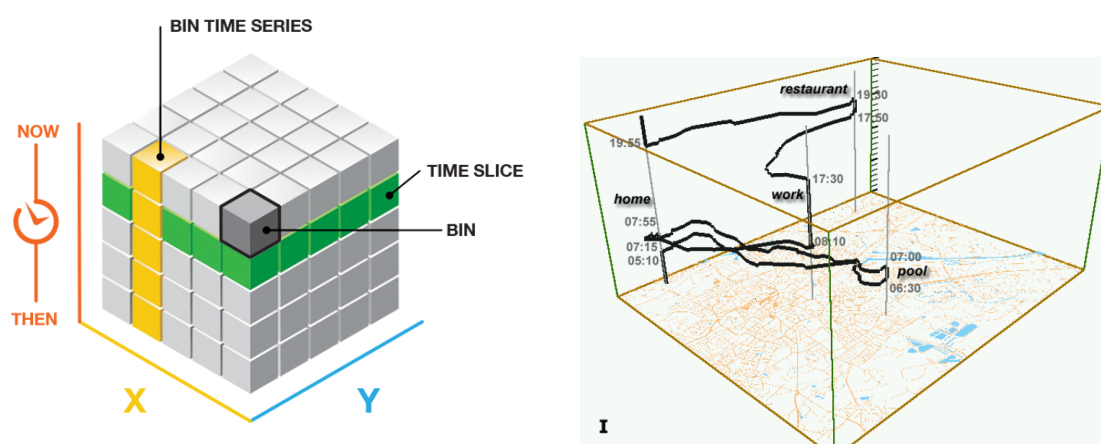
UpSet by [Lex et al., 2014] is a hybrid visualization containing visual elements from the aggregation-based method and matrix-based method to communicate overlapping sets, see figure 2.8a. Also here the target audience is aimed at experts and focuses on task-driven searches. As with Radial Sets, UpSet contains multiple visual encodings and methods to represent the data. The main view contains the sets. These are visualized in a matrix layout where sets are the columns. Rows represent a set combination. Next to the set-combination, which are also grouped by the degree of the intersection, is a bar which is used to encode the cardinality of an intersection. The second view, located on



(a) UpSet with “Movie Genres” dataset. Degree of intersections are shown as matrix.

(b) PowerSet with “Movie Genres” dataset. Degree of intersections are shown as treemap.

Figure 2.8: UpSet & PowerSet



(a) Shows a Space-Time Cube constructed by multiple smaller cubes.

(b) Example showing how simple it is to tack location data over time with a Space-Time Cube.

Figure 2.9: Space Time Cube

the right side, displays the currently selected elements by the user. A web version of UpSet is available at <https://vcg.github.io/upset/>.

PowerSet by [Alsallakh and Ren, 2017] extends on the work of UpSet. Instead of the main view containing a matrix representation of the data it now contains treemaps grouped by degree of intersections, shown in figure 2.8b. This method enables a compact overview of the data and their intersections.

Both techniques have the same drawback as the Radial Sets technique concerning temporal information.

Space-Time Cube

<http://www.aviz.fr/~bbach/spacetimecubes/index.php>

2. RELATED WORK

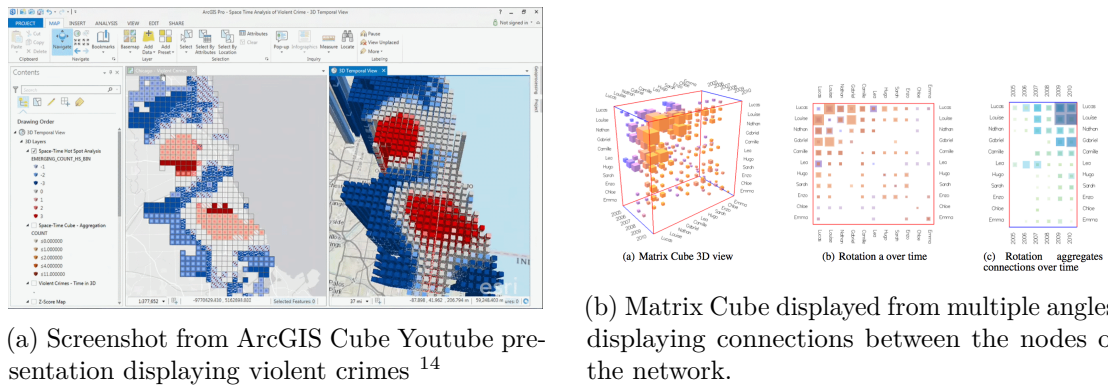


Figure 2.10: ArcGIS & Matrix Cube

Until now every visualization mentioned, based their representation on one parameter which is the set information for example movie genre. If the time should be displayed, things get difficult. In the case of a heat map, one can swap out a category with the temporal information but loses some other information. In case of the UpSet & PowerSet one could add another level of visual encoding below or above the degree of intersections, which would make the overall method more difficult to read.

A Space-time Cube by [Bach et al., 2017] could be used as a representation of any data in combination with the time aspect of the data. Figure 2.9a shows a simple Space-Time Cube constructed by smaller cubes. It is easy to see how a heatmap can fit into this visualization by stacking multiple heatmaps on top of each other to show changes over time. Most of the time a Space-Time Cube is used to show movement of an object or person on a map over a certain amount of time, as shown in figure 2.9b.

With defined operations on the Space-Time Cube other visualizations can be constructed. With these operations, it is possible to visualize many of the previously mentioned set-based visualizations or other methods are shown in the report by [Alsallakh et al., 2014]. For example, a treemap could be created by performing a “Time Cutting” and “Aggregation”; data is extracted from a certain time-range and can then be aggregated to a treemap from this time-range.

The following paragraphs will highlight two implementations of a Space-Time Cube.

In the ArcGIS¹⁵ implementation the focus is the visualization and creation of interactive maps. For example, a Space-Time Cube can visualize economic trends in geographical areas in a certain time range. Figure 2.10a shows the ArcGIS interface displaying violent crimes. More information is available in their documentation¹⁶ or via a short presentation

¹⁴<https://www.youtube.com/watch?v=rl62SnhIvJ8>, Last checked Nov 20, 2017

¹⁵<https://www.arcgis.com/features/index.html>, Last checked Nov 20, 2017

¹⁶<http://desktop.arcgis.com/en/arcmap/10.3/tools/space-time-pattern-mining-toolbox/visualizing-cube-data.htm>, Last checked Nov 20, 2017

of their system on Youtube¹⁷.

Matrix Cubes by [Bach et al., 2014] is designed to visualize dynamic networks and the changes like network topology over time. The Matrix Cube is created by adjacency matrix for each time step with the topological information about the network. By stacking those matrices in chronological order a three-dimensional cube is constructed. The cells become cubes as shown in figure 2.10b. The resulting cube behaves like a physical cube, it can be rotated and sliced either along the timeline, rows or columns each representing a different view on the data.

Finding a way to visualize set-typed data in a two-dimensional space is not an easy task, so finding a way to integrate the temporal aspect of the data is even more difficult. Imagine a simple Venn diagram with four sets and now represent the information over time.

2.5 Summary

In this chapter requirements for casual users have been presented and selected techniques for cultural heritage and set-based visualization have been described.

To visualize some information for casual users one has to look out for certain requirements for a digital collection visualization. It is also vital to keep the visualization as simple and intuitive as possible.

Looking at the visualizations in the area of cultural heritage it is clear that most of the presented techniques have a way to interact with the temporal information. Only “Wikigalaxy” and “Library Observatory” avoid the integration of time, in case of “Wikigalaxy” the focus was more on the exploration. “MKG Hamburg Collection” and “Google Arts & Culture” have a way to select time periods through options or by selecting a specific event¹⁸. In all other techniques time is an important part of exploring and finding interesting things in the collection.

The selected techniques for set-based visualization are mostly targeted towards expert users with many features and require a high cognitive load and a steeper learning curve. If a visualization wants to focus on casual users, it needs to drop some features or visual elements.

As shown in this report, many set-based visualizations exclude the information of time. The visualizations represent a snapshot in time to display the data within this time-range.

As discussed in the report, by working with the “small multiples” technique multiple versions of the same visualization can be displayed each representing another point in time. This approach can work on techniques that don’t require a large screen size to be usable.

¹⁷<https://www.youtube.com/watch?v=d170TZgduy0>, Last checked Nov 20, 2017

¹⁸<https://www.google.com/culturalinstitute/beta/category/event>, Last checked Nov 20, 2017

The other option to add temporal information is the addition of a third dimension. Introducing another dimension is not an easy task and can confuse users, which makes it even more important that the visual elements need to be simple and intuitive.

Since the implementation will be included into Polycube framework, which visualizes data inside a Space-Time Cube, we have to look at the possibilities we have.

The Space-Time Cube could be a good fit for the previously listed recommendations and was not evaluated for the CH domain. Also, this visualization method is not widely used like line, bar or pie charts to present information, which in turn can trigger the exploratory behavior because its a more interesting way to show the information. The cube would support multiple data dimensions including the temporal information [Windhager et al., 2016b]. Higher dimensional integration is also supported by encoding information into the cube with color, size of elements and so on. With the enhanced navigation possibilities of the Space-Time Cube, the users can have a better understanding of the temporal encodings.

Concluded from the report, no set-based visualization provides a way to display a collection over time in one view. This work proposed a set-based visualization integrated within a Space-Time Cube to represent a digital CH collection and its changes over time. This visualization should be integratable into the Polycube project. The resulting technique targets casual users. Therefore simplicity and intuitive controls are required.

Methodology

Before exploring the data, it needs to be visible. This chapter outlines the proposed method to guide the design and development process of the visualization. In the first section, the Nested Model is explained, a multi-level approach for designing and validating of a visualization. The Design Triangle, described in the second section, is another framework for designing an interactive visualization. The last section explains the evaluation method used to validate the prototype.

3.1 Nested Model

The Nested Model, by [Munzner, 2009], splits the design process into multiple levels each of them with its own validation. Each level represents a different problem. Since the levels are nested, the output from one level becomes the input for the next one. That means design decisions from the outer levels will carry over to inner levels, also errors. In total there are four design levels as shown in figure 3.1. The first level, describes the problem and data of a specific domain, clearly understanding the target audience and the problem. Followed by the level for data and operation abstraction, transforming the domain-specific problem and data into a more abstract one. Level three deals with the encoding and interaction techniques design. The last level is the design and implementation of the algorithm to carry out the visual encoding.

Each level in this model has different threats and requires a different validation. To avoid errors cascading to inner levels, Munzner recommends two validations per level, one on entering and one leaving a level.

Based on the described method, this is the nested model for this work:

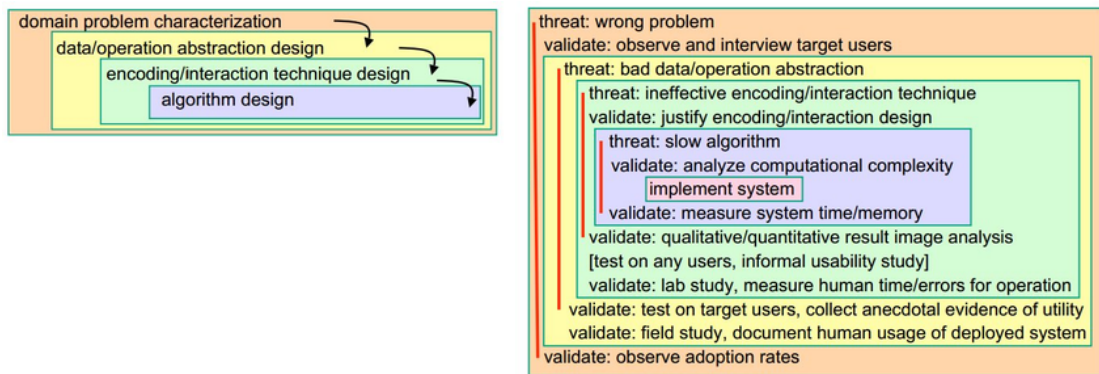
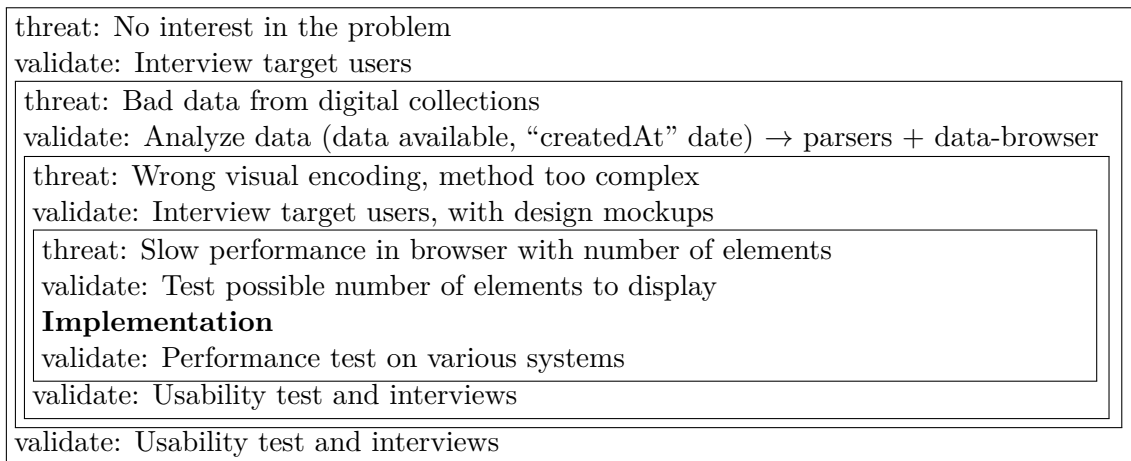


Figure 3.1: Nested Model showing the four levels.



3.2 Design Triangle

The Design Triangle, introduced by [Miksch and Aigner, 2014], provides a simple and very effective framework for the design process. The framework focuses on three crucial elements, the data, the users and the tasks. Based on those factors the designer has to answer the three main questions of the method:

- Data: What kinds of data are the users working with?
- Users: Who are the users of the Visual Analytics (VA) solution?
- Tasks: What are the tasks of the users?

The answers determine which visual representation, analytical, and interaction methods are suitable. A visual representation of the triangle is shown in figure 3.2.

Based on the questions from the design triangle, following requirements were identified:

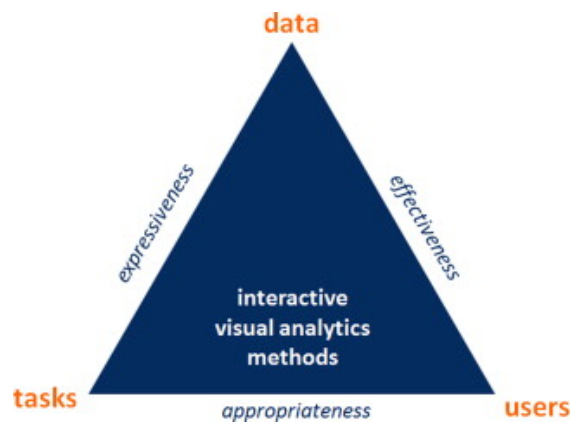


Figure 3.2: Design Triangle showing the three elements of the triangle plus who they are connected to.

- Data: Cultural Heritage data (or other domain), set-typed data, heterogeneous data (different artworks, different information).
- Users: Casual users, exploring data on their own, discovering something new, Simple visualization.
- Tasks: Exploring, no clear goal, finding new things, entertainment.

Guided by the two design methods, an interface was designed which will be explained in the next chapter.

The validation in the design and development process followed the validation rules of the nested model as described before. As the final validation method for the prototype, interviews were conducted to verify the intuitiveness of the created culture cube, more on this in chapter 6.

Design

The following chapter contains the design process of the implemented visualization, based on the design requirements. Main features of the implementation will be listed and explained in detail. The prototype will be displayed in various stages of development, including some algorithms behind the visual design and the analyses of the dataset. The first sections will cover the requirements and early analyses of the datasets. Mockups and early designs of the Space-Time Cube are covered after that. Since the cube contains five different visualizations, each of those will be highlighted and explained. The chapter concludes with the final prototype given to the participants of the user study, shown in figure 4.9b.

4.1 Design requirements

As mentioned in the previous sections there are multiple criteria a visualization should fulfill when targeted towards casual users.

How the prototype covers the various requirements is described in the following paragraph.

- **Keep the visualizations simple** — Since the resulting Space-Time Cube is a three-dimensional visualization, it is already more complex to visualize sets. To keep the visual prototype simple, set operations are limited to unions and intersections but are not the focus of this work. Further, all visual elements are represented in cubes or rectangles, to limit the number of different objects.
- **Provide orientation** — The visualization starts with the complete collection over the whole timeline. Users can freely rotate and zoom the camera. Also, the timeline can be restricted to a time-range.
- **Support exploratory search** — It is possible to select any cube or rectangle of the prototype and display the objects that correspond to this set element.

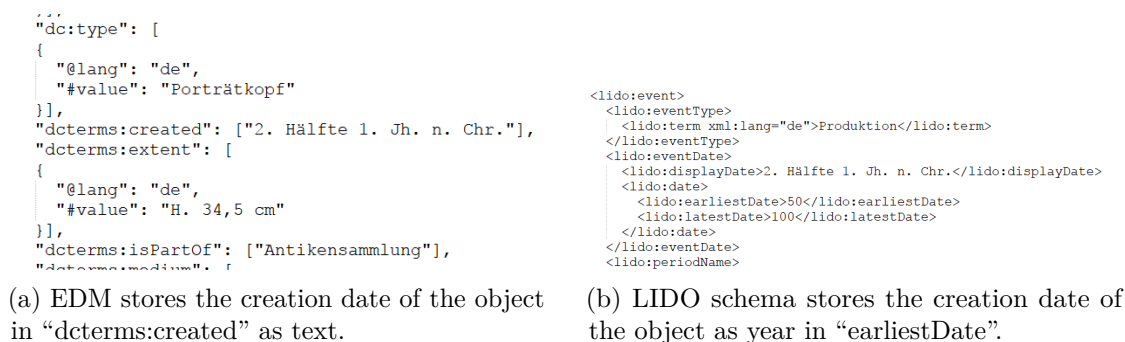


Figure 4.1: Year encodings

- **Increase user experience** — The experience the users will have depends on the data and its visualization. Therefore every interaction with the cube should display objects from the collection.
- **Keeping cognitive load low** — Controls and elements are limited in shapes and interaction methods to limit cognitive memory of the interface.
- **Intuitive interaction** — Since the elements represent common shapes and the controls are simple enough by only using mouse controls.

In addition to those requirements, the application should be accessible to a broad number of users. To tackle this requirement a web application was chosen.

4.2 Analyze the datasets

Before the data can be adequately visualized and designed, the data needed to be analyzed. For this task, parsers were created that transformed the dataset into a format that is simpler to process and modify.

As a test dataset for this work, an export of the KHM was selected. The information as provided in two different formats. The first one was extracted with the Europeana Web API and stored as a JavaScript Object Notation (JSON) file. The objects within the file were stored in the EDM [europeana, 2016]. The second format was provided directly from the KHM itself as an Extensible Markup Language (XML) file. Objects within this document were described in the Lightweight Information Describing Objects (LIDO) schema described by [Autere and Vakkari, 2011]¹.

¹<http://www.lido-schema.org/>, Last checked Nov 20, 2017

Due to the JSON file of the first format, the data was directly accessible to display in a web application and insertable into a NoSQL database, to make query and search requests easier. The dataset contains all relevant information like name, description, type of the object, material, collection the object is part of, and so on. Through analyzing the objects in the dataset, it was discovered that the temporal information did not follow a defined standard and was not easily processable. As the example in figure 4.1a, the creation date of an object was stored as “2. Hälfte 1. Jh. n. Chr.” which roughly translates to “second half of the first century A.D.”. For our prototype or design, the information of the creation year is important to order the items on a timeline. That would mean an additional parser to transform this information to number only containing the year, in this case, “50 A.D.”.

Luckily the second format did not have that drawback. The dates in the XML file were stored as numbers with a negative number representing the “B. C.” and positive number the “A. D.” years, shown in figure 4.1b. To use this information a parser was created that transformed all the XML data to JSON and stored them into the same NoSQL database as the first format.

Parsers

This section describes the parser applications, that converts the input from the dataset into a usable data structure for the prototype.

KHM-insert-into-mongodb.js reads the Europeana data from the JSON file line by line and stores it into a collection within a MongoDB database.

KHM-query-mongo.js is the script that enables querying and makes creating aggregations over the dataset possible.

For the second dataset, **LIDO-convert2json.js** is a small script that transforms the XML file with the KHM data in Lido schema into a JSON file. It also stores the resulting data directly into a collection of a MongoDB database. Searching the database is much more comfortable than in files.

Dataset explorer

Next step in the design process was to take a closer look at both datasets, for this purpose a small web application was created. This approach made the searching for appropriate information in the dataset much simpler. The dataset explorer shows all objects within the KHM dataset ordered by the creation date of the object, shown in figure 4.2a. In addition to the title, a description, the repository (KHM) and most importantly the classification of the item. This information will later be used in the culture cube to aggregate objects into sets. By clicking on the “image” or “Europeana-image”, the browser opens a new window that shows the object on the KHM website or the Europeana website. Figure 4.2b shows a dialog with all classifications with the number of objects per classification.

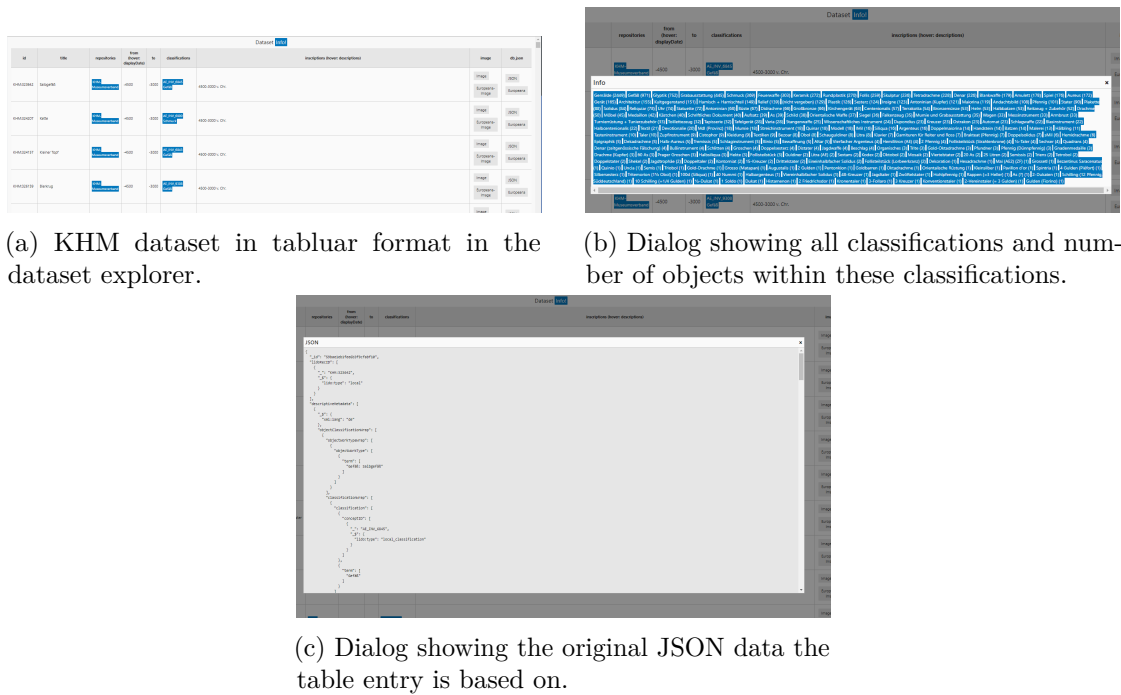


Figure 4.2: dataset explorer

The classifications are sorted by the number of objects within them. This dialog can be opened by clicking on the info button. If an item needs to be further explored, all one needs to do is click a button in the “db_json” column to open a dialog for the LIDO or Europeana JSON data, see figure 4.2c. This feature makes a closer look at the original JSON data possible, for both datasets. Technology wise the application was created with the vue.js JavaScript framework.

4.3 The First Mockups

After analyzing the dataset and deciding that the LIDO dataset is a better fit to visualize objects over time, classifications were chosen as an ideal candidate to aggregate objects in sets for this work. The first mockup was created to give an initial overview of the user-interface, shown in figure 4.3. Users were provided with a cube based view of the collection spanning over multiple years, the visible range of the years should be freely customizable by the user. The detailed view on the left side of the interface uses a tabular layout to represent the current selection. Users can interact with the cube by clicking on elements of the cube or changing the time-range. Visualizing the structure of the visible time-frame was already an idea in this mockup. Clicking on a single object should present more detailed information about the object.

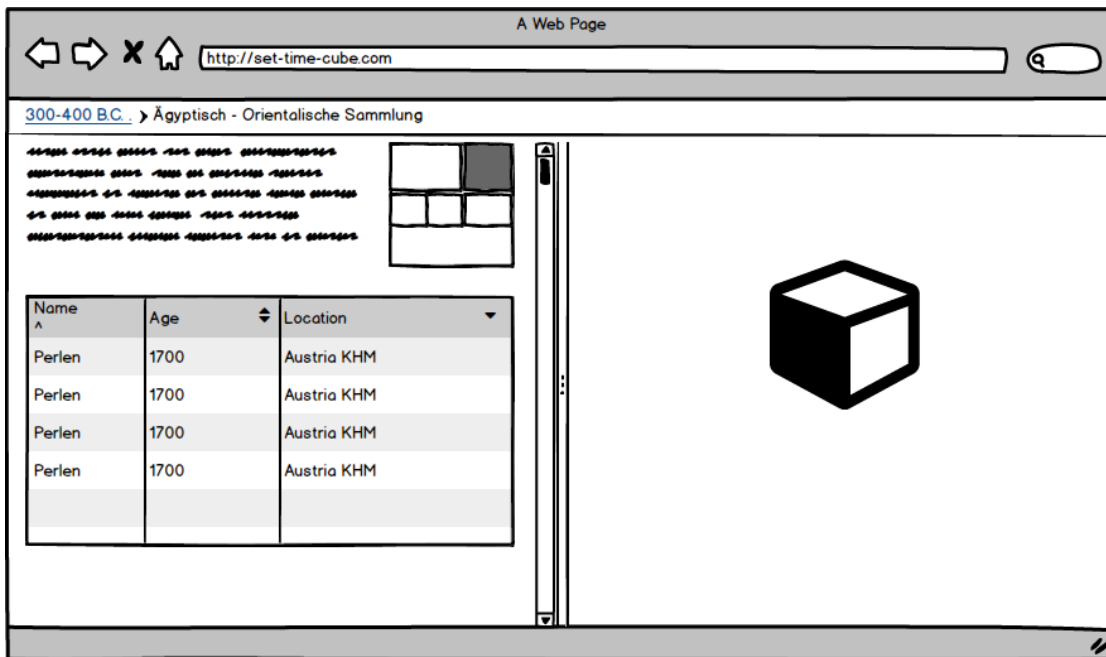
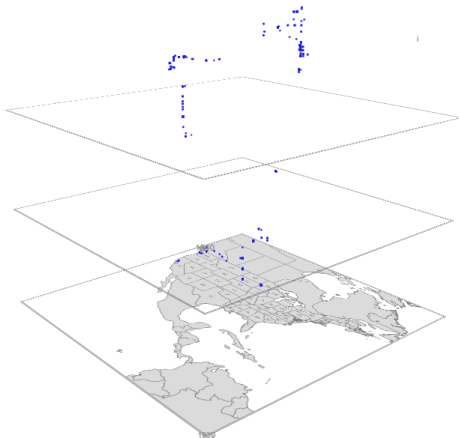
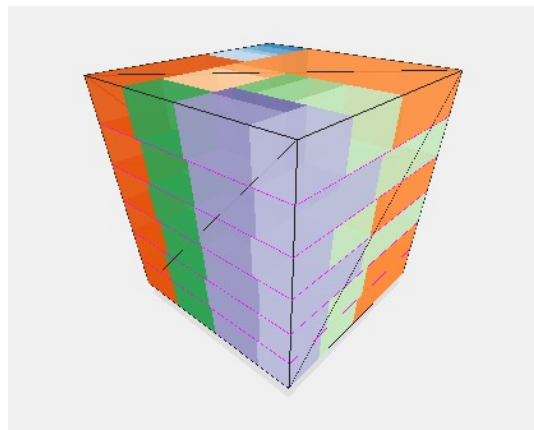


Figure 4.3: First design draft of the Culture Cube interface. Containing a detail view on the left and a Space-Time Cube view on the right side.



(a) Polycube visualizing pictures taken from a photograph on his journey through North America.



(b) Early version of the set module for Polycube. Showing a treemap visualization in six time-ranges.

Figure 4.4: Polycube

4.4 Polycube

For the design and implementation of the cube itself, Polycube project was used as a starting point². Polycube aims to support users in the exploration of CH collections. Figure 4.4a shows that visualizing objects with geographical data inside the Space-Time Cube is already possible with the Polycube framework. Many other controls and feature like, zooming, rotating, rendering, were already present in the framework. Utilizing the provided features from the framework proved very beneficial in the development process. With the integration of set visualization, many new features were added to the Polycube project. More on the implementation of the integration will follow in chapter 5.

After analyzing the framework and finding possible points to integrate the set visualization a quick proof of concept was created. The visualization in figure 4.4b contains treemaps representing the time-ranges stacked on top of each other. Each layer shows the current distribution of objects in the collection based on their classification in the current time-range.

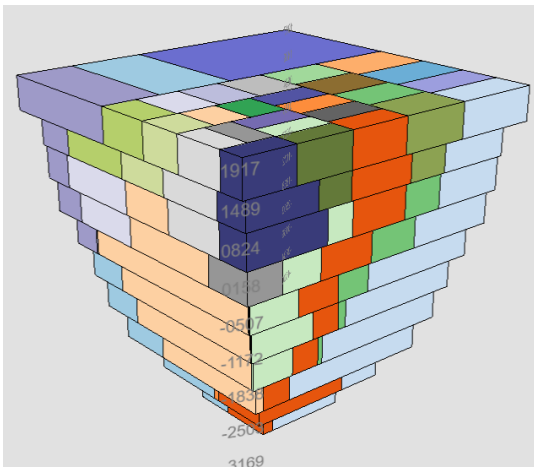
4.5 Set Visualizations

To keep the visual design simple and cognitive load low all elements are represented by cubes or rectangles. No new elements or shapes were introduced to the overall framework. On startup of the visualization, the time-range spans the whole collection to provide the user with an overview of the complete collection.

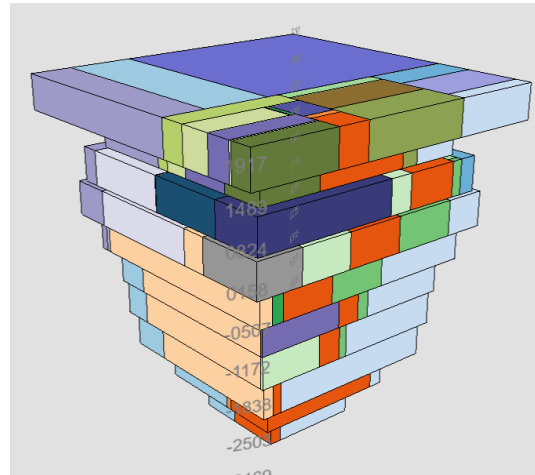
The computer mouse was chosen as interaction medium for all interactions. Following interaction are possible with the set visualization:

- **Click left mouse button** — Performs a selection of the sets on the current position of the mouse pointer.
- **Click right mouse button** — Removes the current selection of the set.
- **Holding left mouse button while moving** — Will rotate the Space-Time Cube.
- **Holding right mouse button while moving** — Will change the position of the camera orbiting the Space-Time Cube
- **Scrolling the mouse wheel** — Will zoom in and out of the Space-Time Cube, making smaller objects visible or giving more overview to the collection.

²<http://www.cvast.tuwien.ac.at/ieg-projects/polycube>³, last checked Jan 21, 2018



(a) Treemap visualizes the growth of a digital collection over time.



(b) Treemap visualizes when the objects were created.

Figure 4.5: Treemap

Combined interactions like clicking while holding the “Ctrl” button on the keyboard were not introduced to keep the interactions simple.

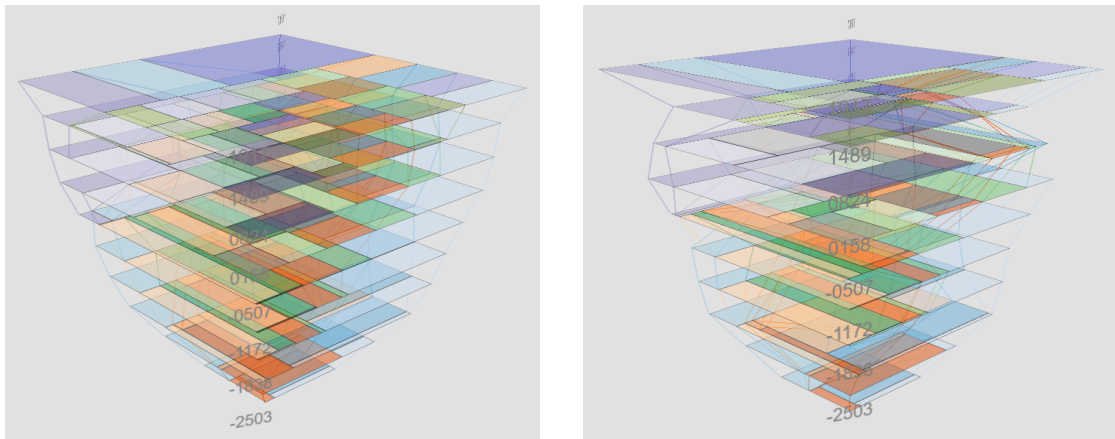
Most of the chosen initial parameters for the set visualization can be altered through configuration options. Instead of starting by viewing the whole collection, the time-range for Ancient Greece could be chosen. More on those configuration options in chapter 5.

The application enables the user to switch between two different modes to display the data: (1) show the growth of the collection over time or (2) show the items and when they were created.

Since it is not clear what the best visual representation is for sets over time, four different visualization types were designed to answer the research questions. These types can be split into two groups. Treemap types try to visualize two aspects of the data, the classification and the time. With matrix types, three aspects of the data are encoded in the visualization, the classification, the collection the objects belong to and the time. Time, in this case, is the creation year of the object.

Treemap

Treemap is the first visual style introduced. The cube is split into ten equally sized layers each representing a time-range. Each of those layers represents the overview of the collection at this moment in time and each rectangle represents a set. The overview is presented with a treemap layout since it is one of the most straightforward visualizations to gain an overview of data. The size of each layer is scaled to the total number of objects



(a) Treemap Flat visualizes the growth of a digital collection over time.

(b) Treemap Flat visualizes when the objects were created.

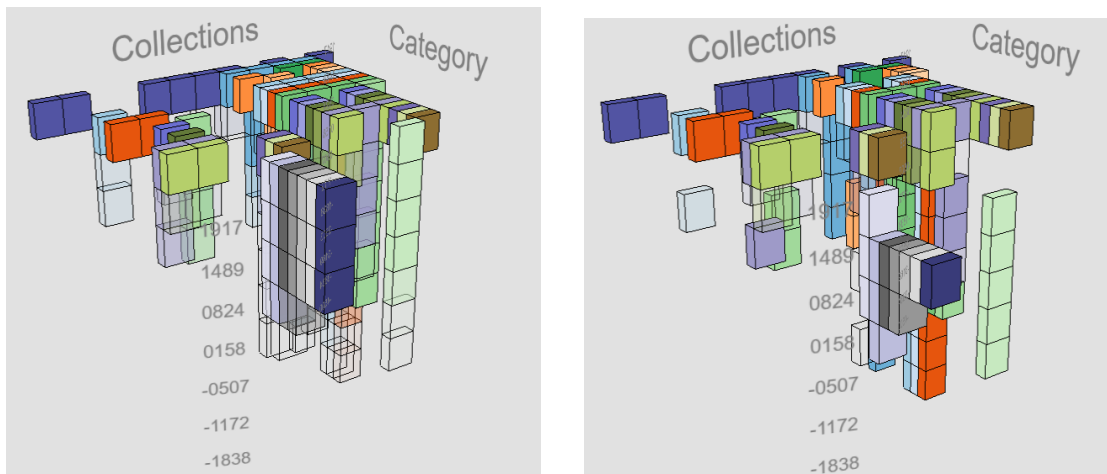
Figure 4.6: Treemap Flat

that are within the layer, which gives the image of an inverted pyramid. The *Treemap* is calculated by the classification of the objects within the time-range. For example, if an object is classified as “Gemälde” it will appear in the sets representing the category “Gemälde”. If an object has more than one category, such objects would be assigned to all its sets. So objects with “Glyptik” and “Gefäß” would show up in both sets. This decision was made since the classifications in the dataset were not hierarchical, but it is still interesting to represent more than one category per object. Figure 4.5a visualizes the growth of the digital collection based on the creation year of the objects. The layers are summed up over time. Each object stays within the collection from the moment it was created. No object is deleted from the digital collection. The collection of the present-day is shown on the topmost layer of the cube. Figure 4.5b on the other hand shows the same objects, but removes the growth visualization and thereby only shows items in the time-range they were created.

To keep the rectangles of the *Treemap* stable, an algorithm was used which will be described in chapter 5.

Treemap Flat

Since the regular *Treemap* method is filling the space between each layer, another treemap method was introduced. *Treemap Flat* uses the same calculation and positioning mechanics as the previous method, but renders the treemap not as cubical rectangles but rather as flat surfaces. To keep the visual connection between the sets over time intact, they are connected with lines in the same color. With this method a look inside the cube is possible and by adding lines, patterns forming over the years are still visible.



(a) Matrix visualizes the growth of a digital collection over time.

(b) Matrix visualizes when the objects were created.

Figure 4.7: Matrix

Also with this method both modes are available, shown in figure 4.6.

Matrix

This method is the first one to encode three aspects of the data with the cube, namely the category and collection, in combination with time. Instead of collection other properties of items could be chosen like crafting material or the country where the object was created. Visualized it has a close resemblance to a Heatmap. The cube is divided into multiple boxes based on the number of collections and categories (classifications) in the dataset. Having 12 collections and 30 categories would result in 360 possibly visible boxes per layer. For some combinations of collection and category, no boxes are visible since there is no data available in the dataset. For example, the “Gemäldegalerie” contains only objects of category “Gemälde” therefore every other box in this row is not visible. Also with this method the growth mode applies, but rather than changing the size of the boxes, the number of objects in each cube is encoded as transparency, shown in figure 4.7a. Once a box is fully colored each object with the same category is contained within the cube. Figure 4.7b displays the objects when they were created. Gaps between boxes appear since in certain years no items of this category were created.

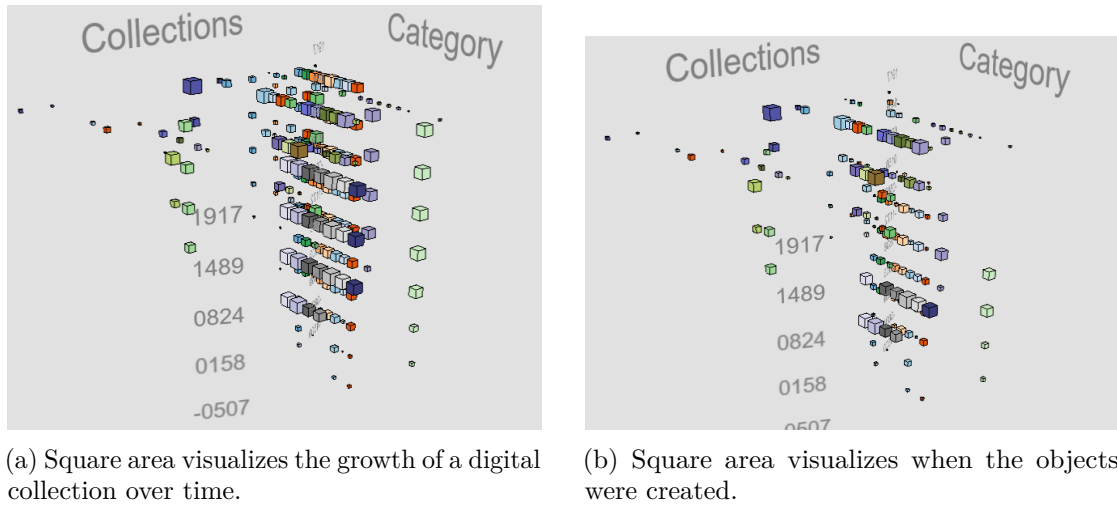


Figure 4.8: Square Area

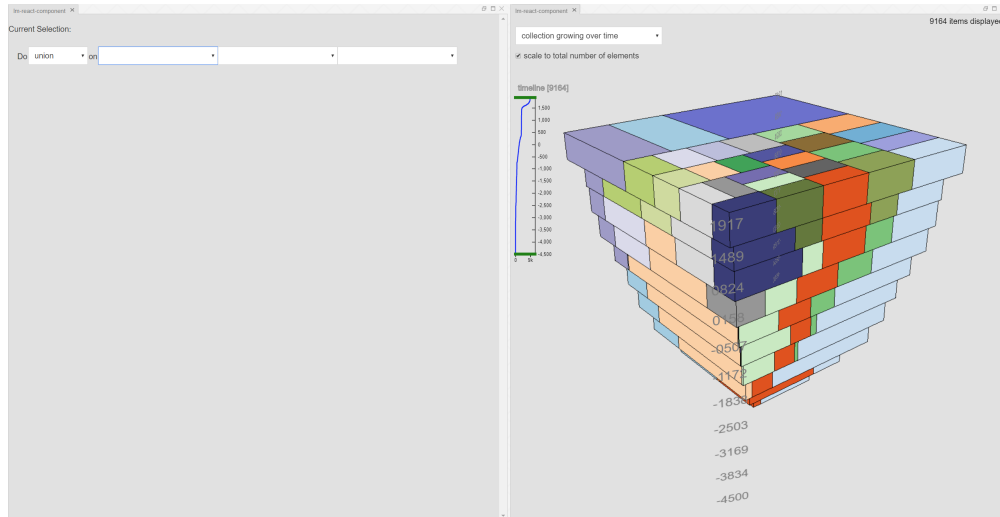
Square Area

The last method introduced is the *Square Area*. This method uses the same calculated matrix as the previous method and same positioning of the boxes. The difference lies in the size of the boxes, now representing the number of objects contained in each box. Also here it is possible to switch from growth mode in figure 4.8a to created mode in figure 4.8b.

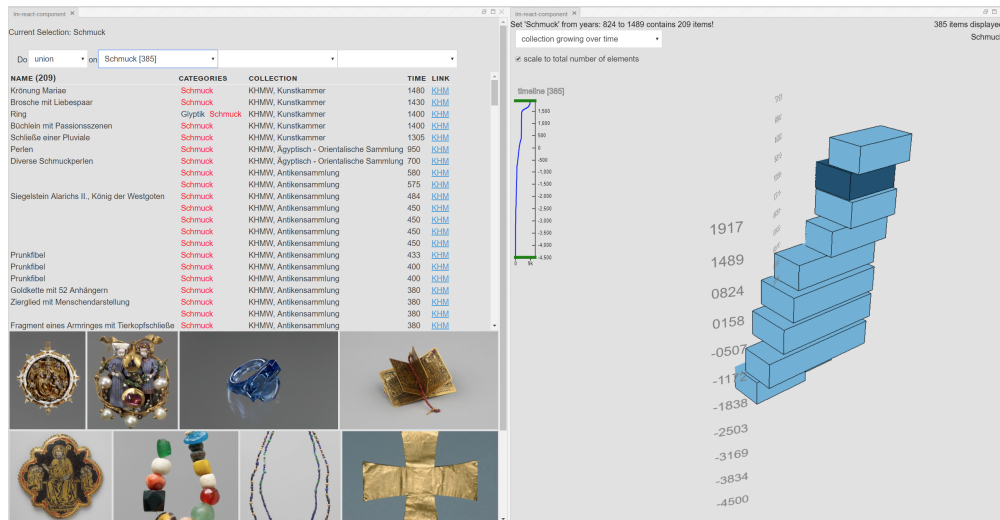
4.6 Culture Cube Interface

After designing the cube and its interactions, it is now time to introduce the finished Culture Cube Interface and all its features. The interface is split into two main views, (1) the detail view where all the selected elements are presented and (2) the cube view containing the Space-Time Cube. Figure 4.9a shows the interface when it is initially loaded. Both views are linked with each other, meaning actions performed in one view will trigger changes in the other one. As example, selecting a box on the cube side of the interface will display all objects with the relevant criteria on the detail view. As shown in figure 4.9b, after performing a selection on category “Schmuck”. The selection mechanism will be covered in more detail later on.

The different visualization types can be switched by adding the query parameter “vis_type” to the website url, possible values are *treemap*, *treemap_flat*, *matrix*, *square_area*. By adding the parameter “layers”, the user can configure the number of visible layers in the cube. Opening https://syrenio.github.io/mt-culture-cube/?vis_type=treemap&layers=50 will display the *Treemap* with 50 layers shown in figure 4.10.



(a) Culture Cube Interface after initial loading. Displays the entire data of the cube, but keeps detail view empty since no selection is available.



(b) Culture Cube Interface after performing a selection on the category “Schmuck” and hovering over the box containing objects from 824 to 1489 and displaying information on top of the cube view.

Figure 4.9: Culture Cube Interface

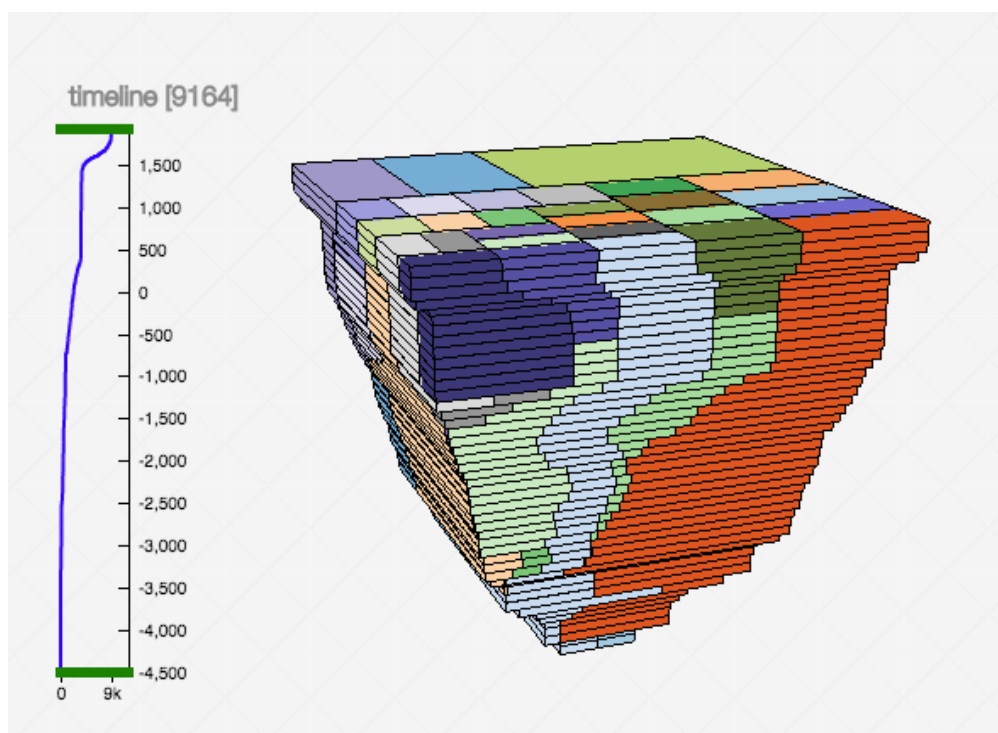


Figure 4.10: Culture Cube with 50 layers

Cube View

Figure 4.11 shows the cube view of the developed prototype. It contains the cube and various controls to support the users in their exploration of the digital collection. The cube is positioned in the middle of the view, with two axis placed at two opposite positions of the cube, avoiding loss of orientation when the user is rotating the cube. Each of the controls is highlighted with a number in figure 4.11.

At position (1), the content of the box the users is currently hovering over with the mouse cursor is displayed. It will show the name of the set, the time-range and the number of items within a box. The current hovered box will appear in a highlight color. By clicking on a box, the set representing this box will be selected and shown in the detail view.

(2) shows the drop-down to switch between the growing/cumulative representation of the collection or displaying the objects only in the time-range they were created.

With the checkbox next to (3), the users can manually disable the scaling of boxes within the cube. Manipulating the scaling can be helpful when boxes become too small to interact with them, see figure 4.8a.

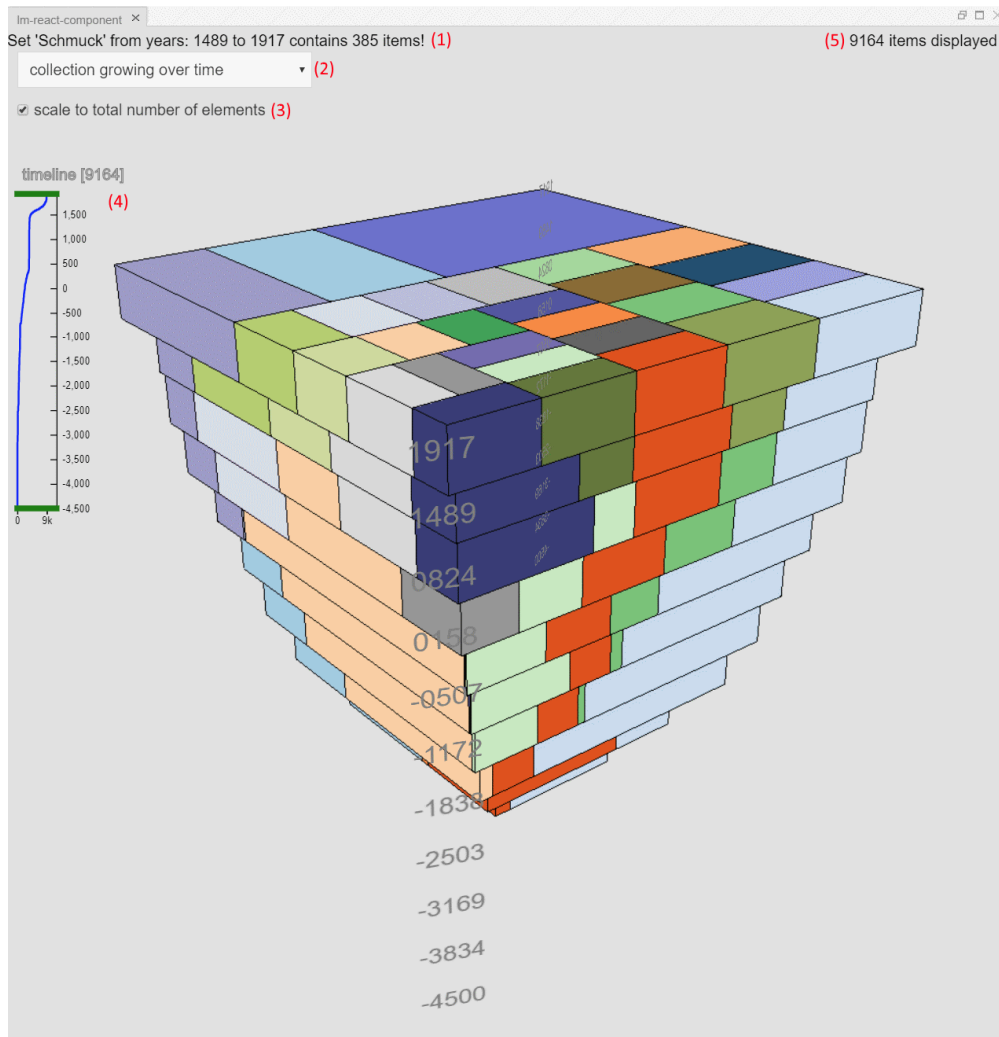


Figure 4.11: Culture Cube Interface cube view. (1) tooltip of the hovered box, (2) drop-down to switch between modes, (3) disable and enable scaling of the boxes, (4) timeline, (5) number of visualized objects.

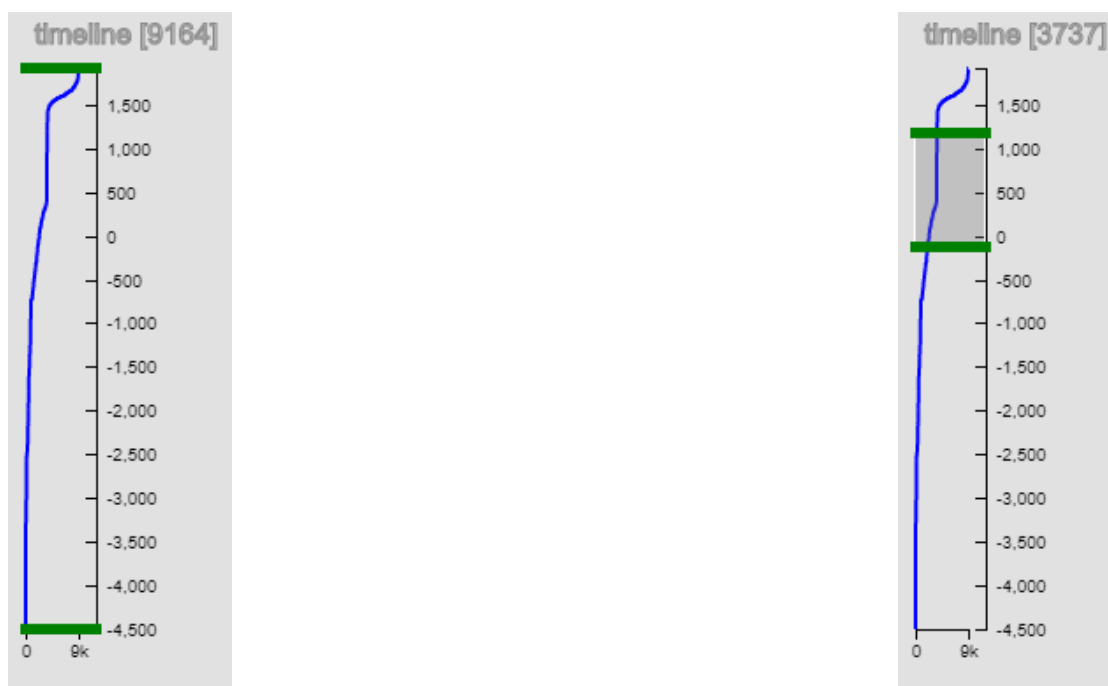


Figure 4.12: (left) timeline with no specific time-frame selected. (right) timeline with a time-frame selected between 50 B.C. and 1200 A.D.

To restrict the view to a specific time-frame, users can manipulate the timeline at (4). This can be done by moving the green bars around or by making a brushing move inside the timeline. This action will update the cube accordingly. The number next to “timeline” is the number of objects that are in the selected time-frame. Figure 4.12 shows the difference between the timeline when it shows the whole collection and when the collection is limited to show only a part of history. The blue line in the timeline is a line-chart representing the number of items within the collection to a given point in time. Finally (5) shows the current number of visualized objects in the current selection plus time-frame and additionally displays the currently active selection.

Detail View

Figure 4.13 shows the detail view of the prototype. The user can examine the content of the currently active selection in this view. In addition, the view helps users perform more complex selections. As with the previous view, the controls in the pictures are numbered

Current Selection: Schmuck (1)

Do union on Schmuck [385]

NAME (209)	CATEGORIES	COLLECTION	TIME	LINK (3)
Krönung Mariae	Schmuck	KHMW, Kunstammer	1480	KHM
Brosche mit Liebespaar	Schmuck	KHMW, Kunstammer	1430	KHM
Ring	Glyptik Schmuck	KHMW, Kunstammer	1400	KHM
Büchlein mit Passionsszenen	Schmuck	KHMW, Kunstammer	1400	KHM
Schließe einer Pluviale	Schmuck	KHMW, Kunstammer	1305	KHM
Perlen	Schmuck	KHMW, Ägyptisch - Orientalische Sammlung	950	KHM
Diverse Schmuckperlen	Schmuck	KHMW, Ägyptisch - Orientalische Sammlung	700	KHM
Siegelstein Alarichs II., König der Westgoten	Schmuck	KHMW, Antikensammlung	580	KHM
	Schmuck	KHMW, Antikensammlung	575	KHM
	Schmuck	KHMW, Antikensammlung	484	KHM
	Schmuck	KHMW, Antikensammlung	450	KHM
	Schmuck	KHMW, Antikensammlung	450	KHM
Prunkfibel	Schmuck	KHMW, Antikensammlung	450	KHM
	Schmuck	KHMW, Antikensammlung	433	KHM
	Schmuck	KHMW, Antikensammlung	400	KHM
Prunkfibel	Schmuck	KHMW, Antikensammlung	400	KHM
Prunkfibel	Schmuck	KHMW, Antikensammlung	400	KHM
Goldkette mit 52 Anhängern	Schmuck	KHMW, Antikensammlung	380	KHM
Zierglied mit Mensendarstellung	Schmuck	KHMW, Antikensammlung	380	KHM
Zierglied mit Mensendarstellung	Schmuck	KHMW, Antikensammlung	380	KHM
Fragment eines Arminges mit Tierkopfschließe	Schmuck	KHMW, Antikensammlung	380	KHM

(4)

Figure 4.13: Culture Cube Interface detail view. (1) displays current active selection, (2) drop-downs to perform set operations, (3) tabular display of selected objects, (4) area displaying an picture and description of an object.

[Kaiser Franz Joseph \(1830 - 1916\) und seine Braut bei einer Ausfahrt in der Umgebung von Ischl](#)

Das Bild, das aus dem Besitz des Oberststallmeisteramtes stammt, stellt die erste gemeinsame Ausfahrt Kaiser Franz Josephs und seiner Braut Prinzessin Elisabeth in Bayern dar, die am 21.8.1853 bei Bad Ischl stattfand. Das junge Paar sitzt in einem Landschützer, der von Graf Grüne, einem Vertrauten des Kaisers, gelenkt wird. Der Landschützer, ein ländlicher offener Wagen, ist mit sechs Pferden im "Wildgang" bespannt. Bei dieser aus Ungarn stammenden Anspannungsart gingen zwei Pferde an der Deichsel und davor vier parallel an eine Vorauswaage gespannte Pferde.

Figure 4.14: Displaying a single object in detail view with a description and picture of the object.

and will be explained in detail. The current selection will be displayed at (1), this can be a single set or an operation between multiple sets.

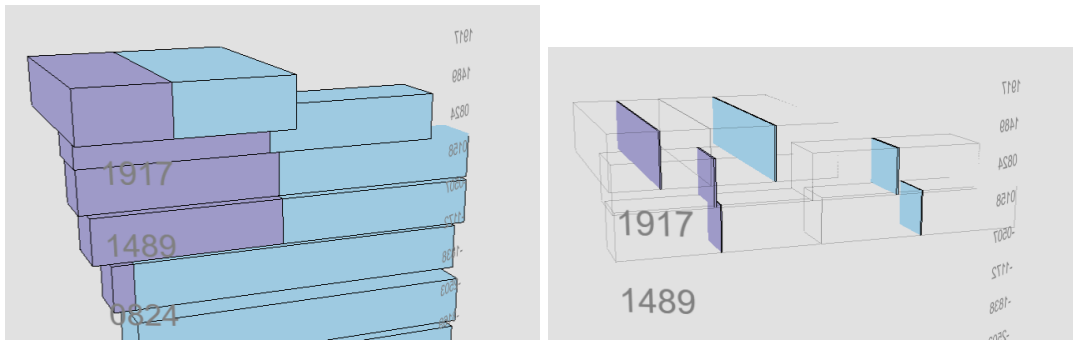
The mentioned selection and set operation can be performed at (2). For this the user selects one out of two possible operations (union and intersection), then he or she selects up to 3 different sets in the following drop-downs. Union operation between the sets “Schmuck” and “Glyptik” will show all objects that are classified either as “Schmuck” or as “Glyptik”, this will be shown as “Schmuck \cup Glyptik”. The other possible operation is a set intersection, it is shown as “Schmuck \cap Glyptik” and will only show objects that have both classifications. Since intersections between sets are not the primary focus of this work, this functionality was only design and developed in a limited scope. Possibly PowerSet by [Alsallakh and Ren, 2017] could be a candidate to visualize intersections over time. How these set operations are displayed in the cube view, will be shown in the next section.

(3) Displays a table containing all objects within the current selection. Following informations about the objects are shown in the table: name of the object, categories the object belongs to are listed in the second column, the category on which the selection is based on is highlighted in red. Column three shows collections in which the object is part of. The table is ordered by the year the object was created, displaying newer items before older ones. Additionally, each entry contains a link to the original object in the digital collection, clicking on it will open a new tab in the browser showing the object on the KHM website. The area marked with the number (4), displays a picture wall created with the first ten objects of the selection. Clicking on the picture or any entry in the table will replace the picture wall with a detailed description of this specific item, shown in figure 4.14. By clicking the name of the object will open the KHM entry of this item.

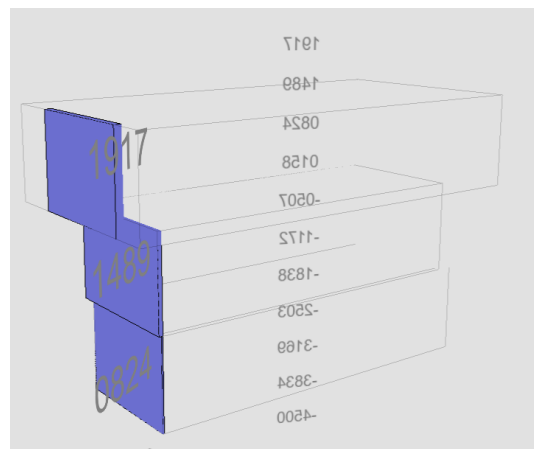
Selection

Now to one of the most important interaction possibilities of this cube: selection of a set or category. As mentioned before, a single set can be selected via the cube view or by selecting one category in one of the drop-downs on the left side. If the user chooses to perform a set-operation between multiple categories, the detail view display the information in the same way as with a single set selection. The boxes in the cube view will update according to the number of items and highlight this area. The original size of the boxes will still be visible but slightly transparent to compare it with the selected part. If the items have multiple categories, then this is shown as well and boxes for these categories will be displayed.

Since a treemap is typically used to visualize hierarchal structures, therefore one object should only be available in one box. By referencing an object in multiple boxes as done in the *Treemap* visualization type, we ignored this rule of a treemap. Later in the development process, another visualization method was added which reintroduced this rule of the treemap. The *Treemap Hierarchy* method tackles this problem by giving the assigned classifications of objects an hierarchy, by using the order of the classification



(a) Shows a union selection between “Glyptik” and “Gefäß”. (b) Shows a intersection between “Glyptik” and “Gefäß”.



(c) Shows a intersection between “Glyptik” and “Gefäß” using the Treemap Hierarchy method.

Figure 4.15: Displaying results of set-operations between “Glyptik” and “Gefäß”

entries. With this method, the object is removed from showing up in multiple boxes. An example of this is shown in figure 4.15c. Figure 4.15 shows three ways the selection of multiple sets is represented in the cube.

Juxtaposition

The last feature shown in this chapter is the possibility to split the cube, by its layers and position the layers next to each other. This feature enables users to compare data from different points in time, see figure 4.16. The transition from the cube to the Juxtaposition is done with an animation. This feature was implemented in the cube and interface but was not directly accessible to the users in the user study.

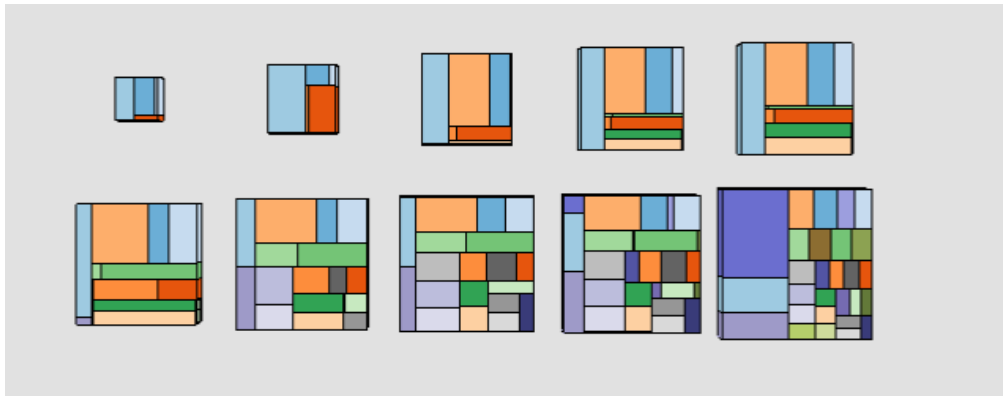


Figure 4.16: Displaying the layers in Juxtaposition.

Implementation

This chapter covers the implementation details of the prototype and the used technologies. Every technology, framework, and library will be discussed in a short overview. The second and third section will cover the architecture of the set-based cube and the final interface of the prototype.

5.1 Technology

This section contains the list of libraries used to develop the prototype. Each one of them will have a short explanation about the functionality and how they were used in the implementation. The application was chosen to be developed as a web application for easy accessibility and experience in development of these types of applications.

HTML, CSS, JavaScript

Hypertext Markup Language (HTML) and Cascading Style Sheets (CSS) are the core technologies for building web pages. HTML is used to structure the web page by using forms, paragraphs, lists, tables and so on. CSS describes the presentation of the web page, including colors, layout, and fonts of the various elements. JavaScript (JS) is the scripting language of the web and gained massive popularity in the last couple of years and is currently leading the fifteen most popular languages on GitHub¹.

D3.js

<https://d3js.org/>

Data-Driven Documents in short D3.js introduced in the work by [Bostock et al., 2011], is a JavaScript library for manipulating documents based on data. D3 transforms the

¹<https://octoverse.github.com/>, Last checked Feb 15, 2018

data to powerful visualizations with the help of HTML, Scalable Vector Graphics (SVG), and CSS. It comes with a wide variety of different visualization modules and algorithms to manipulate the data. The *d3-hierarchy* module of D3.js is used to calculate the layout of the treemap visualization which is used in the set module of Polycube.

three.js

<https://threejs.org/>

Another JavaScript library used in Polycube is *three.js*. It is a lightweight 3D library, that is easy to use. By using this library, it is possible to draw various 3D objects on a web page. The set module uses the Web Graphics Library (WebGL) renderer to draw all three-dimensional boxes and lines while the CSS3D renderer is used for the flat surfaces and rectangles.

tween.js

<https://github.com/tweenjs/tween.js/>

The next library called *tween.js* is used in the set module to create animations when interacting with the cube. This tweening engine helps to provide easy animations, by manipulating properties of a JavaScript object to target a value in a specified time.

dat.gui

<https://github.com/dataarts/dat.gui>

Dat.gui adds a small, lightweight graphical user interface to a website for changing variables in JavaScript. This interface was beneficial in the development and rapid testing in the development environment of the set module, shown in figure 5.1.

Node.js

<https://nodejs.org>

The open-source JavaScript run-time environment allows for executing JavaScript code on the server-side, which before was typically only used in the browser. One benefit which made NodeJS so popular was that the same code could run on the server and in the browser. For the development of the Culture Cube interface, it is used as the build tool and managing dependencies with the Node Package Manager (NPM).

react

<https://reactjs.org/>

To develop the Culture Cube interface *react.js*, currently the most popular front-end framework², was used. React is a JavaScript library for building user interfaces and

²<https://stateofjs.com/2017/front-end/results>, Last checked Feb 15, 2018

focusing on the view part of the application, that means the developer is free to combine other frameworks and libraries with React. All parts of an interface are organized as components and are self-contained and simple to move around.

babel & webpack

<https://babeljs.io/> & <https://webpack.js.org/>

The development environment of the Culture Cube interface also makes use of *babel.js* a JavaScript compiler and *webpack* a module bundler for JS applications. To make use of new JavaScript features that are currently not supported in browsers, babel is used for compiling new language features to the current state of JavaScript in the browser. The job of webpack is to collect all needed dependencies of our prototype and bundle it into one single JS file that can be used to deploy on a web page, additionally webpack includes a dev-server that was used to test the application.

golden-layout

<https://golden-layout.com/>

To manage the two views of the Culture Cube interface *golden-layout* was chosen. With this layout-manager users can easily resize, hide, and show the views. The library was designed to be easily integrable with React and made resizing the views of the prototype very simple.

lodash

<https://lodash.com/>

The utility library *lodash* for JavaScript was chosen to make certain tasks easier for the programmer. Some features involve interacting with arrays, manipulating objects and calculation over the dataset. This library is used in both the set module and Culture Cube interface.

5.2 Set Module for Polycube

<https://github.com/bigdata-vis/polycube>

Polycube is an in-development web-based application to explore CH objects in a Space-Time Cube based on geographical data. The Polycube framework forms the basis for the implementation of the set module. Installation of the module, configuration options, main methods, and algorithms will be highlighted.

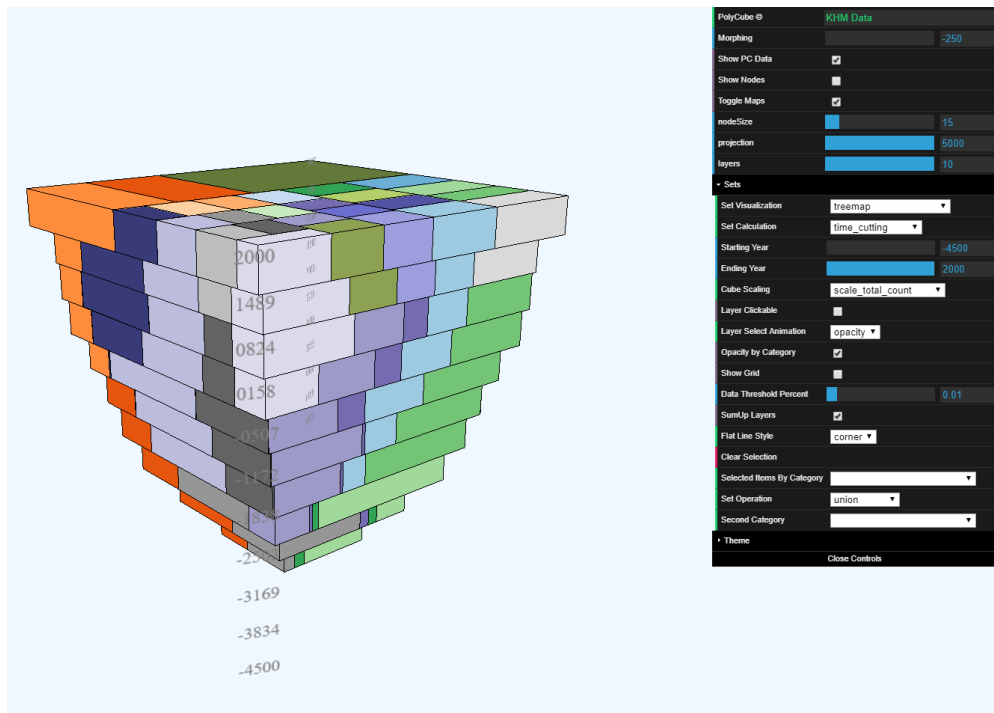


Figure 5.1: Polycube set module development environment with dat.gui configurable options.

The JavaScript module pattern is used to enhance the Polycube with features and functionality³. Each module is self-contained and private functions don't interfere with the other modules. The main module *pcmodule.js* of Polycube covers the visualization of the geographical position of items over time in a Space-Time Cube. Building on top of this framework the set module *pcmodule_sets.js* was created, with the goal of visualizing set information in a Space-Time Cube. All code for the set module is contained in this JS file.

Install & Options

Listing 5.1 shows the minimal setup for the set-based visualization with Polycube. For the current prototype, it is still necessary to provide the geo dataset for the main module. The module contains a wide variety of configuration options which are described in table 5.1. Options can be configured in a JSON object and given as argument to the *polyCube.drawSets()* function.

³<https://coryryan.com/blog/javascript-module-pattern-basics>, Last checked Feb 15, 2018

Table 5.1: List of configuration options with defaults

Name	Type	Default	Possible Values	Description
parsedData	Array			Data that will be processed and visualized as set in Polycube, more about the data format in section 5.2
vis_type	String	'treemap'	'treemap', 'treemap_flat', 'matrix', 'square_area', 'treemap_hierarchy'	Select the type of set visualization 4.5
vis_type_treemap_flat_line_style	String	'corner'	'corner', 'center'	Defines how the lines between rectangles are drawn in 'treemap_flat' mode.
vis_type_treemap_sort_by	String	'value'	'value', 'name'	Defines how the treemap algorithm orders the rectangles.
vis_type_matrix_count_opacity	Boolean	true		Enable/Disable transparency of boxes in matrix style.
vis_type_matrix_show_grid	Boolean	false		Enable/Disable a helper grid for matrix style.
vis_type_layer_clickable	Boolean	true		Enable/Disable if the layers are clickable.
vis_type_layer_click_animation	String	'move'	'move', 'opacity'	If layers are clickable define the animation that should be performed.
vis_type_select_type	String	'color'	'color', 'visibility'	Defines the visual selection mode when a selection on sets is performed.
vis_type_set_calculation_option	String	'time_cutting'	'time_cutting', 'only_time_range'	Should the calculation of sets be based on the current selected time range, still knowing about data outside this range or ignore all other data and rebuilding the cube based on this.
data_year_range	Range			Time range that should be visualized.
data_with_categories	Array	[]		Filter data based on the sets within this list (ignore all other sets).
data_scale_cube	String	'scale_total_count'	'scale_total_count', 'scale_category_count'	Switch between scaling modes of the cube.
data_layer_sumUp	Boolean	true		Enable/Disable the sum up or aggregation over time mechanism.
data_threshold	Number	0.01		Filter data that is below the data threshold (removes sets are below 1% of the data).
data_threshold_remove_categories_below	Boolean	true		Enable/Disable the removal of a set from the items if the set is below the 'data_threshold'
onLayerClick	Function	prints to console.info		Defines what happens if user clicks on an layer.
onSetClick	Function	prints to console.info		Defines what happens if user clicks on a set.
onSetHover	Function	returns information as string		Defines what happens if user hovers on a set.

parsedData

The *parsedData* property stores the list of objects visualized in the cube. Every object in the *parsedData* array must contain following fields and information.

- **lidoRecID** — Unique ID of the objects, for the prototype the ID from the LIDO dataset was used.
- **time** — The creation year of the object.
- **term** — List of sets the object is part of (in the prototype, classification/category).
- **repoName** — Name of the collection the object belongs to.

Listing 5.1: Minimal setup for polycube

```
1 <html>
2 <head>
3   <script src="//threejs.org/build/three.min.js"></script>
4     /* include JavaScript libraries like three.js, d3.js, tween.js, ...
5       */
6     <script src="js/pmodule.js"></script> // main polycube module
7     <script src="js/pmodule_sets.js"></script> // set visualization
8       module for polycube
9 </head>
10 <body>
11   /* fetching data from backend */
12   var geoDataElements = { ... };
13   var options = {
14     parsedData: listOfObjects, // contains the list of items that
15       should be used for the set visualization.
16     vis_type: 'treemap',
17     // ...
18   };
19   polyCube.drawElements(geoDataElements, numberOfLayers);
20   polyCube.drawSets(options)
21   polyCube.render();
22   polyCube.animate();
23 </body>
24 </html>
```

polycube.drawSets(options)

drawSets is the main function of the set module and takes care of drawing and redrawing the visualization within the Polycube framework. Listing 5.1 shows the pseudo code the

function goes through and the main tasks it performs.

Algorithm 5.1: drawSets algorithm

Data: Configuration options from section 5.2

Result: Draws a set visualization in polycube by given options

- 1 Clear previous sets and visual elements;
 - 2 Filter data based on the configuration options;
 - 3 Based on filtered data and layers: calculate ticks and time-ranges;
 - 4 Create data structure for treemap and matrix visualizations;
 - 5 **while** *not all filtered data processed* **do**
 - 6 Read current object from filtered data;
 - 7 layerNumber = calculate layer based on the creation date of the object;
 - 8 setName = read classification from object;
 - 9 Store objects in data structure, e.g. treemapStructure[layerNumber][setName]
 = object;
 - 10 **end**
 - 11 Draw Layers;
 - 12 Draw Treemap | TreemapFlat | Matrix | SquareArea | TreemapHierarchy;
 - 13 Update items based on a selection (if available);
-

In the process of drawing the treemap, an algorithm was used to keep the treemap stable over the different time ranges in the Space-Time Cube. For this, the mechanism utilizes the *d3-hierarchy* module of the *d3.js* library. Listing 5.2 shows the pseudo-code of this algorithm and the functions used from the library. The main idea is to keep the treemap stable and to avoid unnecessary movement of the different sets. The relatively stable treemap was achieved by calculating the layout once in the beginning with all entries in the dataset and stored until the cube is destroyed or rendered with new data. The stored layout will be used to recalculate the rectangles on any given layer. Features like *treemapResquarify*⁴ were very helpful. The *doTreemapLayout* function is used every time

⁴<https://github.com/d3/d3-hierarchy#treemapResquarify>, Last checked Feb 15, 2018

a layer is drawn.

Algorithm 5.2: doTreemapLayout algorithm with d3.js and d3-hierarchy

Data: Treemap dataset & layer number

Result: Rectangles of the treemap with their position and size

```
1 _tmap = d3.treemap().tile(d3.treemapResquarify);
2 if _simple_hierarchy_root was not defined then
    /* initial calculation of the layout */
3   _simple_hierarchy_root = d3.hierarchy(listOfAllSets);
4   _simple_hierarchy_root = _simple_hierarchy_root.sum(/* calculate the
    number of objects within each set over the whole dataset */).sort(/* sorting
    based on configuration option */);
5 end
6 _simple_hierarchy_root = _simple_hierarchy_root.sum(/* calculate the number
    of objects within each set in this specific layer. */).sort(/* sorting based on
    configuration option */);
7 return _tmap(_simple_hierarchy_root).leaves();
```

polycube.selectSet(setName)

The first selection method that was implemented involved highlighting all visual elements that belong to the same set. This action is possible with *selectSet(setName)*, a public method performing exactly this task. Depending on the configuration options the elements will be highlighted in a defined color, and all other elements will fade to white or other elements will disappear from the cube leaving only the selected category visible.

polycube.selectItemsBySets(setNames, overlappingOption, repoName)

For a selection of multiple sets a different function called *selectItemsBySets* was developed. This method focuses on the objects itself instead of boxes, thereby making it possible to show parts of a box, this can be seen in figure 4.15b. The function works with a list of set names (*setNames*) and a set operation (*overlappingOption*) provided by parameters. The *repoName* provides the additional possibility to restrict the visible elements by a specific collection.

Attach Module to Polycube

Attaching the set module to the Polycube was straightforward. Polycube exposes many different connection points to integrate a module, some of which will be listed in this section.

- **polyCube.root** — Exposes the HTML element where the cube attaches itself to the web page.
- **polyCube.getGLBox()** — Exposes the 3D object where all elements are added.
- **polyCube.getGLSegments()** — Exposes the objects representing the layers of the cube.
- **polyCube.getGLRenderer()** — Exposes the renderer for the cube.
- **polyCube.render_functions** — Is an array of functions, that allows a module to add functions that are called every time the cube rendered.
- **polyCube.default_functions** — Similar to *render_functions* but executes every time the cube moves back to default position. With this solution it was possible to reuse some of the animation from the main module.
- **polyCube.juxstaPose_functions** — Similar to *render_functions* but executes every time the cube animates to Juxtaposition.

5.3 Culture Cube Interface

<https://github.com/syrenio/mt-culture-cube>

This section will cover the implementation details of the Culture Cube interface that was given to the participants of the user study. The prototype was developed as React application and integrated the Polycube + set module in a React component. The application itself has no backend it only requires a location where the data is hosted and can be fetched.

The repository itself contains all the data and parsers mentioned in this and the previous chapter, the Culture Cube interface is available in the *cc-interface* directory. To start the application on your local computer only two commands in your command line interface should suffice. First make sure NodeJS is installed on your computer and move to the *cc-interface* directory.

- *npm install* — Will install all software dependencies for the application.
- *npm start* — Will start the Culture Cube interface under the address `http://localhost:3100`.

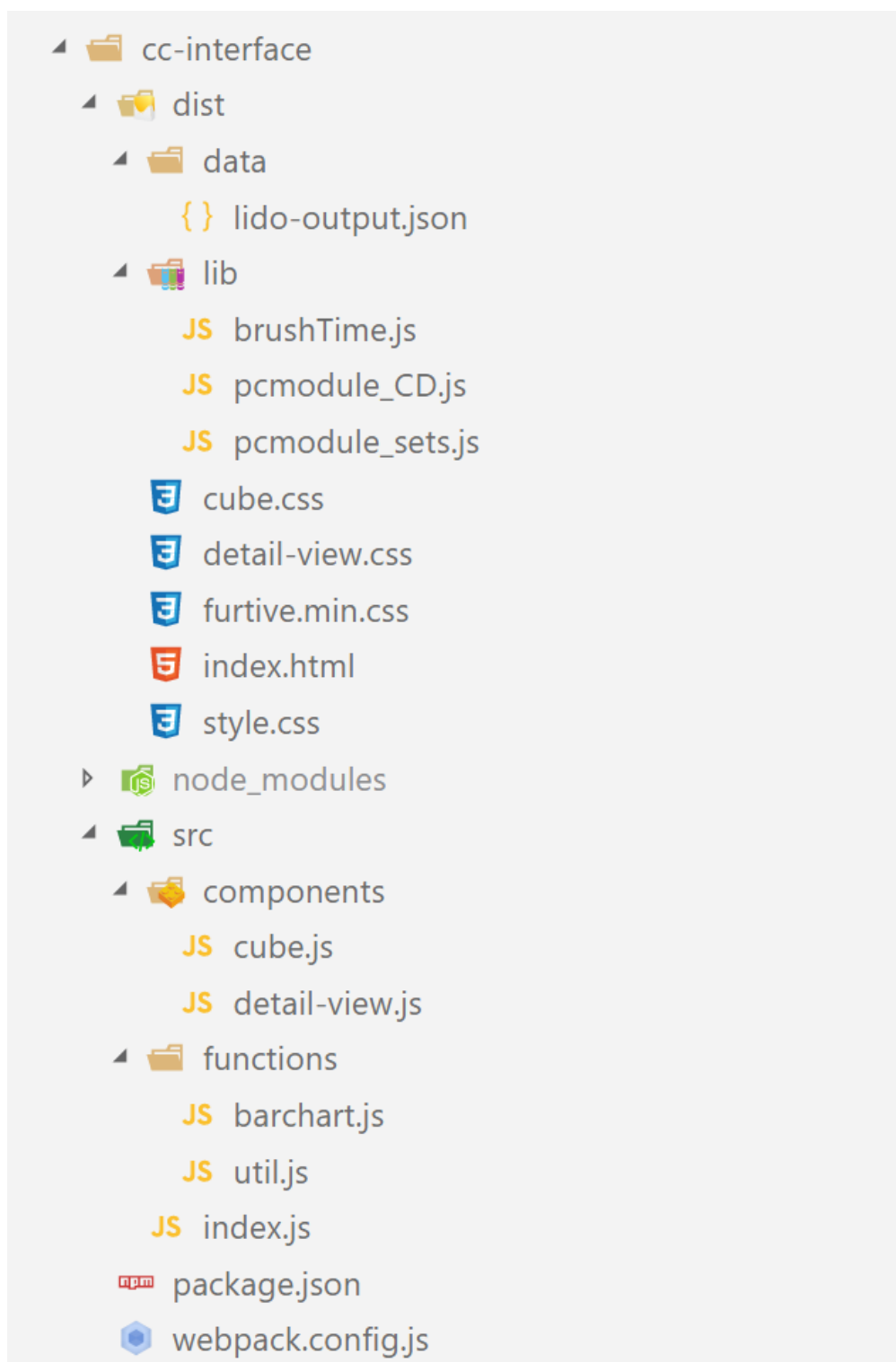


Figure 5.2: Culture Cube interface project structure

App structure

Figure 5.2 shows the project structure of the prototype. All content needed to run the prototype is included in the *cc-interface* directory. After installing all the required dependencies with *npm install*, it is possible to start the application with *npm start*. The CH collection of the KHM can now be explored on <http://localhost:3100>.

- **package.json** — Contains the project description, software dependencies and starting scripts.
- **webpack.config.js** — Is the configuration of the build-system and configuration for the development server.
- **src** — This directory contains all the relevant JavaScript source files. *Index.js* is the place where the golden-layout is configured and application is attached to the HTML.
- **src/components** — Contains the detail view component and cube view component with Polycube.
- **dist** — This directory contains files that are needed to host the application on a web server. *Index.html* is the bootstrap HTML where all files are loaded. The CSS files contribute to the design of the individual components.
- **dist/data** — The data directory holds the dataset of the KHM data in LIDO JSON format.
- **dist/lib** — Polycube, the set module and the timeline libraries are stored here.

Polycube as React component

This section will cover how the Polycube was integrated into a React component. Listing 5.2 shows important points in the integration setup. The React component is connected with the Polycube over the configuration options described in the previous sections.

Listing 5.2: React component with Polycube

```

1 import React, { Component } from 'react'
2 import { getParameterByName } from '../functions/util';
3
4 export default class CubeWrapper extends Component {
5   state = {...};
6   parsedLidoData = [];
7   layers = 10;
8   cubeOptions = {
9     vis_type: getParameterByName('vis_type') || 'treemap',
10    vis_type_set_calculation_option: polyCube.SET_CALCULATION_OPTIONS[0],

```

5. IMPLEMENTATION

```
11   data_year_range: [],
12   ...
13 };
14
15
16 constructor(props) {
17   super(props);
18
19   this.init = this.init.bind(this);
20   this.handleOnSetHover = this.handleOnSetHover.bind(this);
21   this.handleSelectSet = this.handleSelectSet.bind(this);
22   this.handleOnSelectItemsClick = this.handleOnSelectItemsClick.bind(
23     this);
24   /* connect react functions with polycube over the configuration options
25      */
26   this.cubeOptions.onSetHover = this.handleOnSetHover;
27   this.cubeOptions.onSetClick = this.handleOnSelectItemsClick;
28   this.cubeOptions.onLayerClick = (data) => {
29     this.props.glEventHub.emit('cube-selection-layer', data);
30   }
31   d3.queue()
32     .defer(d3.csv, "data/mainModuleData.csv")
33     .defer(d3.json, "data/lido-output.json")
34     .await(this.init);
35 }
36
37 render() {
38   return (
39     <div>
40       ...
41       <div id="cube"></div>
42       ...
43     </div>
44   )
45 }
46
47 init(error, mainModuleData, lido) {
48   if (error) console.log(error);
49
50   this.parsedLidoData = lido.map(d => {
51     /* extract data from lido dataset and transform to set module data
52        model*/
53     ...
54     return {
55       lidoRecID,
56       conceptID,
57       time,
58       term,
59       repoName
60     };

```

```
61     });
62
63     /* */
64     polyCube.root = document.getElementById('cube');
65     polyCube.drawElements(mainModuleData, mainModuleData, this.layers);
66     polyCube.drawSets({ ...this.cubeOptions, parsedData:
        this.parsedLidoData });
67     polyCube.render();
68     polyCube.animate();
69
70     ...
71 }
72
73 }
```

Evaluation

This chapter focuses on the evaluation process of the developed prototype described in the previous chapters and if the proposed solution can help answer the research question:

“How can an interactive visualization of cultural heritage collection, support exploratory behaviors of casual users?”

The first section will cover the feedback provided throughout the development process of the application and the resulting changes in the application. The evaluation method will be described in section 6.2 and 6.3 describes the different blocks of the tasks. The results of the user study will be examined in section 6.4. The chapter concludes with a summary of the feedback.

6.1 Feedback in the Development Process

Testing an application or idea is essential to avoid going down the wrong path and wasting time. According to [Nielsen and Landauer, 1993] most usability problems can be found with only five users. Adding more users to tests will result in listing the same issues more than once, not providing new insight. After five users discovering new insights is rare. Fixing the issue after a user test and retesting the application with users brings better results.

With this knowledge, small user interviews were conducted throughout the development process to evaluate the direction of design and application. Tests were performed by fellow students, co-workers, and advisors, to guide the application and eliminate missteps early. Most tests were performed by showing the user the current state of the application and let them interact with it and collect feedback. Multiple bugs were fixed and features

that were not intuitive were changed in later iterations of the prototype. The goal of this interviews was to keep the visual design simple and understandable.

The following list contains some feedback and the actions that were taken to address these issues.

- Too many small elements were present. → We added a filter on the data threshold to minimize the number of tiny sets that contained only one item.
- In early designs, selecting a set would color all elements of the STC white and only the selected part red. → This behavior was not well liked by users. Therefore it was changed to keeping the assigned color of the box and hiding all non-matching ones. This change also solved the problem users had with the changing of colors when selecting.
- The controls of the timeline did not look intractable. → To address handles were added to the timeline control.
- Current Selection was not visible. → Added name of the current selection on multiple places in the interface.
- Hovering over an element did not show the expected information. → Changes the hover function to show set and collection names, plus the number of items and current time-range. If a selection is active on this element, the hover function will act accordingly.
- Links to the KHM should open a tab instead of opening it in the current window. → Opened a new tab.
- User expected to rotate the cube. In early versions, a click and drag would result in a selection. → Fixed by checking if the mouse cursor moved while holding the mouse button.
- Time axis on the cube was not always facing the user and hiding behind the cube. → Fixed by adding two time axis on both sides of the cube.

6.2 Evaluation Method

For the evaluation of the prototype, it was decided to perform a small user study with fifteen participants between the age of 20 to 40. Out of these participants, only two had prior experience with InfoVis, one being an expert. Since the application is aimed at casual users, no specific user group was targeted. Users included business students, software developers, medical experts and archeology students. The tasks for the users are relatively simple in design and simple to answer once the visualization is understood.

Before performing the user study, a small test run with three users was conducted to see if the prototype needed some slight adjustments or the questions had to be more

specific. All three users were able to solve most of the tasks. Unclear questions were rewritten to be more precise. By recommendation of the users, a couple of minutes should be spent an explanation of the application and data inside. Also, small tweaks were performed in the interface. In the *Square Area* method the boxes were too small for solving tasks, this resulted in the addition of the switch to enable and disable scaling. *Matrix* visualization was dropped because users found the *Square Area* more interesting. The number of objects in the Culture Cube was not always clear. Therefore additional labels were introduced to display the current number of objects.

After the first test runs, the user study was conducted. The evaluation process included following steps:

Before the user interview started each participant was given information about the interview process, general task of the interface and data he or she will encounter. Users were encouraged to think out loud, ask questions, and explain their approach to solve the tasks.

During the observation, notes were taken about what elements were intuitive and elements of the Culture Cube were not clear enough.

- Step 1: A short overview of the application, features and the visualization.
- Step 2: Let the user explore without much interference from the interviewer.
- Step 3: After the free exploration, two sets of tasks were given to the user one for treemap style and one for matrix style visualization.
- Step 4: Answering the rating questions about intuitiveness and usability. Additionally, users were asked to answer some open questions about the visualization he or she encountered, speak about additional ideas and which method was liked the most for exploring the dataset.
- Step 5: Finally the users were handed a web address where they could explore the interface whenever they want and even give more feedback later on.

The web application was hosted on a virtual server on a hosting platform. The server was running an Ubuntu 14.04 LTS with a dual-core CPU at 2 GHz and 1-gigabyte memory. The only task of this server was to provide a place to deliver the JavaScript application and data.

The prototype itself is accessible from many different devices. Depending on the location of the interview, a MacBook Pro or a Windows 10 PC was used. The users also had the choice of performing the tasks on their device, if so this will be noted in the results.

Macbook Pro specifications:

MacBook Pro (13-inch, Early 2011)

2.3 GHz Intel Core i5 2 Core

8 GB memory
Intel HD Graphics 3000, 512 MB
Full HD Monitor

Windows 10 PC specifications:

3.3 GHz Intel Core i5 4 Core
16 GB
Nvidia GeForce GTX 970, 1 GB
Full HD Monitor

6.3 User Study

This section covers the design of the questions and tasks for the user study.

For the user study, three out of the five visualization methods were chosen. Every visualization contained the same dataset but was presented differently which makes solving a specific task easier, that is why some of the tasks will differ for each visualization. The Treemap visualization method will be covered with the *Treemap* and *Treemap Flat* method and the Matrix method will be visualized with the *Square Area* visualization. The main idea is to bring users to interact with the interface and fulfill specific tasks. Getting feedback about the usability and intuitiveness of the visual design of sets in the cube and its controls is the primary focus.

To avoid user fatigue only two visualizations will be presented to each participant per session, one of the two treemaps and the matrix style. Method type and order of visualization will differ between participants.

For the evaluation, a list of various tasks and questions were created. All the questions can be found in the appendix of this work. Tasks for *Treemap* and *Treemap Flat* methods are listed in “Treemap Style” section of the appendix. Section “Matrix Style” contains the tasks for the *Square Area* method of the evaluation. Every task has a correct or list of correct values the users should find out. “Rating Questions” contain a “Likert Scale” asking the users to rate features of the Culture Cube from “strongly disagree” to “strongly agree” based on the usability and intuitiveness. Scales were randomly flipped to avoid survey bias. The user study concludes with a section of “Open Questions” asking the user about further ideas and feedback.

6.4 Results

This section contains the results and general feedback from the participants of the user study. The fully answered tasks of the users can be found in the “Answered Tasks” section of the appendix. To easily identify questions each has a unique ID assigned.

Table 6.1: User Study Treemap answers

Task	Correct Answers	Incorrect Answers
T1	15	0
T2	8	7
T3	15	0
T4	15	0
T5	14	1
T6	9	6
T7	15	0
T8	15	0
T9	14	1
T10	11	4
Total	131	19
Percent	87.33%	12.67%

Some tasks have the users write numbers as answers, users were informed that exact numbers are not needed. Already values near the correct answer were counted as a success. For example, guessing the time-range in which the most items were created is hard and depends on the zoom factor. For this, a wider time-range was selected as the correct answer, and every answer that was in this range counted as correct. The interaction with the interface and feedback is more important than solving all tasks correctly.

Treemap & Treemap Flat

In table 6.1 is the summarized result of the tasks based on the treemap visualizations and overall success of the users with this visualization. For this work, it was interesting to find out which of the treemap visualization worked better for users. Eight participants used the *Treemap* and seven used the *Treemap Flat* visual method to perform the tasks. The separated results are available in tables 6.2 and 6.3. Between both visualization method, there are not many differences. In task T2, “Which category has the highest number of total objects, before the year 1000?” both groups had trouble identifying the largest set in the treemap since many sets were similar in size. In task T6, comparing two categories and their growth over time seemed more intuitive in the *Treemap Flat* visualization where the planes are connected with lines. About 87 percent of the tasks were answered correctly.

Matrix

Table 6.4 shows the results of the tasks based on matrix visualization method. The users had trouble answering questions that involved counting the boxes inside the Space-Time Cube (M2, M3). These tasks could be solved by using the scaling function of the interface, but this feature was used less. Comparing questions that are present in both treemap and

Table 6.2: Answers for Treemap method Table 6.3: Answers for Treemap Flat method

Task	Correct	Incorrect
T1	8	0
T2	4	3
T3	8	0
T4	8	0
T5	8	0
T6	3	5
T7	8	0
T8	8	0
T9	7	1
T10	6	2

Task	Correct	Incorrect
T1	7	0
T2	4	3
T3	7	0
T4	7	0
T5	6	1
T6	6	1
T7	7	0
T8	7	0
T9	7	0
T10	5	2

Table 6.4: Answers for Square Area method

Task	Correct Answer	Incorrect Answer
M1	15	0
M2	10	5
M3	11	4
M4	12	3
M5	15	0
M6	15	0
M7	15	0
M8	14	1
M9	15	0
Total	122	13
Percent	90.37%	9.63%

Table 6.5: Total Answers

Vis Type	Correct Answer	Incorrect Answer
Treemap & Treemap Flat	131	19
Square Area	122	13
Total	253	32
Percent	88.8%	11.2%

matrix style visualization no significant difference was found. About 90 percent of the tasks were answered correctly. Since both visualization methods are different regarding the task a user can perform, a more detailed comparison would not make sense.

Table 6.6: Rating Questions answered by users

Question	--	-	-/+	+	++
R1	0	2	0	5	8
R2	0	2	2	8	3
R3	8	6	1	0	0
R4	0	3	4	3	5
R5	0	3	0	5	7
R6	2	7	2	2	2
R7	0	0	2	2	11
R8	1	1	2	3	8

Rating Questions

This section contained the collected overview of the rating questions done in a Likert scale. Table 6.6 presents the results in a tabular form and the same results are also presented in figure 6.1. In general, most users had fun exploring the data in the Space-Time Cube and found that the controls were intuitive but could be improved. They way set operations between sets are performed was easy to understand, but the visualization of the intersection could need more explanation. Half of the participants expected fewer elements to be visible, after a selection, but were confused because of layers. The addition of the two time axes helped the users navigate even when rotating the cube. Selection of the time-range with the brushing timeline was easy to understand. Some users had problems interacting with the controls of the cube itself since not all actions performed with the mouse are intuitive enough, like repositioning the camera position by holding and moving the right mouse button. The scaling options help users solve a task, but most of them needed a reminder of this function.

Feedback & Open Questions

The last part of the evaluation will cover the feedback and observations made throughout the interviews. All open questions were collected and summarized, plus all the observations and spoken feedback were added.

The overall feedback from users was very positive, most of them liked the approach of presenting the collection in a new way. After some initial learning curve the interaction with the cube was easy to understand. By filtering the sets by clicking or drop-down, users had the option to look at the data in details. All users enjoyed the possibility to manipulate the cube to show objects in a specific time-range. Collecting all the tasks answered by users, 88.8 percent of all tasks were answered correctly, shown in table 6.5.

The more interest the user had in the data presented in the cube, the more they explored the data. This interest immediately triggered an exploratory behavior with these users

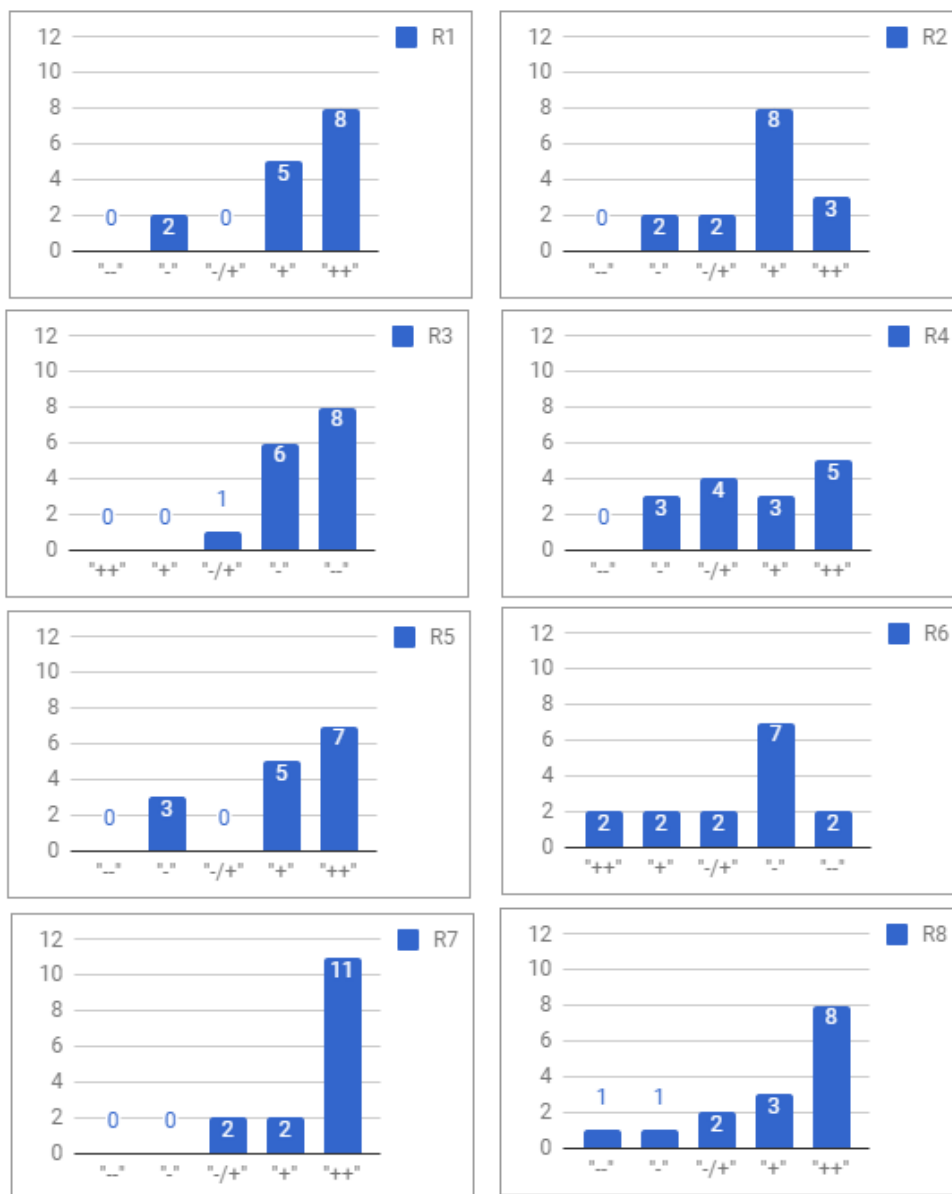


Figure 6.1: Resulting charts of the rating questions with Likert Scale.

and resulted in them enjoying the exploration for more than 20 minutes, before finally starting with the tasks. A better-known dataset like movies, books or music could trigger more exploration from users.

Based on the collected answers in table 6.7 the preferred visualization method was the *Square Area* method. The least favorite method was the *Treemap* visualization, mainly because it hid many elements behind the outer elements of the cube. In addition to that, some users mentioned the jumps between each layer with the *Treemap* method are too big. Changing the number of layers from 10 to 20 would visualize the growth in smaller steps and make it easier to follow the flow of data. One user raised the idea of combining the *Treemap* with *Treemap Flat* methods.

Most users understood the usage of the partially filled boxes for the representation of set intersection. Even with explanation and demonstration at the beginning of the user study, the users rarely used the scaling options of the interface.

After performing the user study, it was clear the Treemap style was more appealing when looking at the changes over time. The *Square Area*, on the other hand, is more interesting at when looking when elements were created.

Half of the users discovered interesting changes between epochs, like the appearance of paintings and immediate decline of burial equipment. It was interesting to see appearing technologies like firearms and the possibility to watch coins more closely than in the museum itself.

With the Treemap styles a problem was discovered, “The whole surface is not representative of the total number of items”, when items are represented in more categories. This problem was discovered while performing an intersection. To avoid this issue the *Treemap Hierarchy* method can be used.

Some users remarked that the information could be easier to access by using multiple two-dimensional visualizations than including everything into a three-dimensional visualization.

Users were surprised about the fluidness of the 3D visualization in the web browser.

The most raised feedback and improvement ideas are listed below.

- **Switching Modes** — Most users found that the feature for switching between the modes was hard to understand. Moving from a *Treemap* representing the mode “growth of the different categories” to displaying the data only in the time-range the item was created, was initially confusing for users. One way of handling this would be to limit a visualization method to one mode. Treemap would be ideal for showing the growth and matrix would be suited to display the created mode. Another way of handling it would be to let the users decide at the beginning of their exploration what they want to explore and explain what each mode means. With the possibility to later exit and reenter with the other mode.

Table 6.7: Preferred visualization method

No Answer	Treemap	Treemap Flat	Square Area
1	2	5	7

- Convenience features were missing** — Most users mentioned the lack of certain convenience features while solving the tasks. A reset button could reposition the cube and all parameters to the initial position of the cube. For an exact input of a time-frame, the timeline could be enhanced with text boxes. Based on the angle of the camera it is hard to connect the axis to the elements in the cube. In the drop-downs instead of the empty entry for selecting all elements, an entry should be created which displays “all elements” to make it more clear what will happen. Displaying the currently active time-range near the timeline would be a great help. A search box in the detail view can help search or reopen already known objects. De-selection or Undo the last action would help users navigate more easily. Drop-downs in detail view were not updated based on the selection performed.
- Animation to communicate changes** — In some cases, it is hard to pinpoint the elements that have changed through the action performed, more animations could be included to highlight the updated parts. With short color flashes in the cube, timeline or axis the changes can be spotted more easily. For example, by selecting a time-range with the timeline, the time axis next to the cube could flash to signal an update. Additionally, a zoom-in and zoom-out animation could trigger to visualize the change.
- Style changes** — The split of the layers or time-ranges felt unnatural for users, better split based on blocks of 50, 100, 200 years or events in time or by epochs. The visualization can be further improved by having different colors on the time-axis for B.C. and A.D. years. More distinct colors could be added to improve the visibility of certain sets. Some description and tooltips need changes to communicate the visualization more clearly. For example, “Set ‘..’ form year: 1489 - 1917 contains 121 items!” could be restructured to “Set ‘..’ collected 121 items until 1917!”.
- Explore a single layer** — Users had the wish to explore a single time-range without manipulating the timeline, but rather by pulling the layer out of the cube and examining this layer separately.

6.5 Summary

This chapter described the evaluation process and detailed steps of the performed user study, as well as the user feedbacks and results, were shown. Discovered issues with the prototype during the evaluation were highlighted and possible improvements discussed. The evaluation shows that most users felt supported by exploring the collection by using the interactive visualization, especially by manipulating the timeline. Controls were intuitive enough to explore the data, although with an initial learning curve. Based on the evaluation, a set-based visualization can be integrated into a Space-Time Cube framework to support the exploration of a cultural heritage collection.

Conclusion

7.1 Conclusion

In this work, a prototype was introduced to answer the previously asked research questions. A user study and interviews, showed a set-based visualization can support the exploration of CH collections. By constructing the visualization within the framework of a Space-Time Cube.

How can an interactive visualization of cultural heritage collection support exploratory behaviors of casual users?

A prototype was implemented that engages the users in exploring the data within the construct of an interactive Space-Time Cube. Through the evaluation of the interface, users were able to explore the digital collection. The feedback provided by the users was then integrated into a next version of the application to support users even further in their exploration of any digital collection.

All of the following sub-questions of the main research question will be answered through the same user study, which evaluates whether the created interface was usable and helpful for users.

How can a set-based visualization be enhanced by the temporal aspect of data?

Two set-based visualizations were chosen to provide an overview of a digital collection, namely treemap and matrix. Normally both methods do not include the temporal information into the visualization itself, except if it displays time. To achieve the inclusion of the temporal aspect of the data and visualize the changes over time one could use animation or small multiples to communicate this. In this prototype multiple layers of these visualizations were stacked onto each other to communicate these changes over time, each layer representing a time-range of the overall dataset.

How can a set-based visualization be represented in a Space-Time Cube?

Based on the Space-Time Cube from the Polycube project a set-based visualization was

integrated, by stacking different 2D visualizations. All cultural objects from a specific time period and based on certain other criteria were aggregated and visualized as boxes, representing the sets.

How can a set-based visualization in combination with temporal aspect be applied in cultural heritage collection?

This question was answered by feeding the application with data from the KHM and letting users explore it. The interface only requires the data to provide a temporal information to position items along the timeline and a criterion to aggregate sets.

As of writing this thesis, to the best of our knowledge, there does not exist another application that uses a three-dimensional Space-Time Cube to represent data from a digital cultural heritage collection. This approach is a first step towards representing set-based information with a Space-Time Cube.

Despite the positive feedback some findings need a detailed discussion.

Keeping the visualization simple is recommended as is leaving some space between the elements so that no item is completely hidden from the user. This issue was mostly responsible for the dislike of the *Treemap* method. Another recommendation is to give users points to orientate, for example by connecting elements to axis.

The Number of participants was relatively small and did not cover older generations. Not all users were initially interested in cultural data, for those users a movie dataset would have triggered more exploration.

The current prototype has no specified limit on the number of items, it only increases the calculation time and depends on the specifications of the computer and browser. All sets that pass a certain configurable threshold of elements in the set are visualized. Setting this threshold option to 0 would display all sets, making the smaller sets difficult to read.

The use of this visualization with other datasets was not performed, but is technically possible, depending on the data. The prototype can be applied to many different datasets and domains, if the data has temporal information and set information.

Some lessons learned during the development of the prototype:

- Explaining the Space-Time Cube without a visual example was difficult. An early implementation helped communicate the idea.
- Smart grouping of visual elements made toggle and animations simpler to implement.
- Using grids to help with calculation and positioning in 3D space is recommended.
- Finding a usable stable treemap algorithm was difficult, but after some research in the D3.js library a possible solution was found. D3.js has many specific modules for different tasks working with data, not only visualization.

7.2 Future Work

Throughout the design, development and evaluation phase various improvements and additional features were discussed, but are out of scope for this thesis. The following list contains ideas and features that can be implemented in future work:

- Zoom into the cube until images of the objects within the sets appear. If a specific item is selected, highlight it with a marker.
- Provide an integrated interactive tutorial with the interface.
- Let users construct the cubes by filtering what categories should appear within the cube and change these filters on the fly.
- Move some of the filtering and calculation from the browser to a server to gain more performance in the initial loading.
- Enhance the options to configure the prototype by letting the user choose the information that is used to build sets.
- Introduce a visualization representing the sets as pillars, which could be based on circle packing.
- Give users the option to define multiple time-ranges and organize them. Alternatively, build the layers based on epochs or time periods.
- Visualize other collections like Movies, Music or Games since users already have a general overview of these topics than cultural heritage and museum artifacts.
- Visualize a Space-Time Cube with Augmented Reality (AR) and let users interact with it.
- Further user study with more participants and possible cooperation with a museum.

List of Figures

2.1	MKG Collection & Google Arts & Culture	11
2.2	Wikigalaxy & Museum of the World	12
2.3	Histography by Matan Stauber, displays events as little dots on a timeline, which can be manipulated. Certain categories of events can be selected on the left side of the interface.	13
2.4	Deutsche Digitale Bibliothek visualized, shows the content of a digital collection in multiple visualizations on one page. Word-clouds, line and bar charts are all visible.	14
2.5	Past Visions & Library Observatory	15
2.6	Venn & Treemap	16
2.7	Heatmap & Radial Sets	18
2.8	UpSet & PowerSet	19
2.9	Space Time Cube	19
2.10	ArcGIS & Matrix Cube	20
3.1	Nested Model showing the four levels.	24
3.2	Design Triangle showing the three elements of the triangle plus who they are connected to.	25
4.1	Year encodings	28
4.2	dataset explorer	30
4.3	First design draft of the Culture Cube interface. Containing a detail view on the left and a Space-Time Cube view on the right side.	31
4.4	Polycube	31
4.5	Treemap	33
4.6	Treemap Flat	34
4.7	Matrix	35
4.8	Square Area	36
4.9	Culture Cube Interface	37
4.10	Culture Cube with 50 layers	38
4.11	Culture Cube Interface cube view. (1) tooltip of the hovered box, (2) drop-down to switch between modes, (3) disable and enable scaling of the boxes, (4) timeline, (5) number of visualized objects.	39

4.12	(left) timeline with no specific time-frame selected. (right) timeline with a time-frame selected between 50 B.C. and 1200 A.D.	40
4.13	Culture Cube Interface detail view. (1) displays current active selection, (2) drop-downs to perform set operations, (3) tabular display of selected objects, (4) area displaying an picture and description of an object.	41
4.14	Displaying a single object in detail view with a description and picture of the object.	41
4.15	Displaying results of set-operations between “Glyptik” and “Gefäß”	43
4.16	Displaying the layers in Juxtaposition.	44
5.1	Polycube set module development environment with dat.gui configurable options.	48
5.2	Culture Cube interface project structure	54
6.1	Resulting charts of the rating questions with Likert Scale.	66

List of Tables

5.1	List of configuration options with defaults	49
6.1	User Study Treemap answers	63
6.2	Answers for Treemap method	64
6.3	Answers for Treemap Flat method	64
6.4	Answers for Square Area method	64
6.5	Total Answers	64
6.6	Rating Questions answered by users	65
6.7	Preferred visualization method	68
1	Answers to the treemap tasks	86
2	Answers to the matrix tasks	87

List of Algorithms

5.1	drawSets algorithm	51
5.2	doTreemapLayout algorithm with d3.js and d3-hierarchy	52

Acronyms

- ACM** Association for Computing Machinery. 7
- API** Application Programming Interface. 2
- AR** Augmented Reality. 73
- CH** Cultural Heritage. 1–3, 9, 55, 71
- CSS** Cascading Style Sheets. 45
- DPLA** Digital Public Library of America. xi, xiii, 1, 15
- EDM** Europeana Data Model. 2, 28
- HTML** Hypertext Markup Language. 45
- HTML5** Hypertext Markup Language. 5
- IEEE** Institute of Electrical and Electronics Engineers. 7
- InfoVis** Information Visualization. 2, 60
- JS** JavaScript. 45
- JSON** JavaScript Object Notation. 5, 28, 48
- KHM** Kunsthistorisches Museum Wien. 9, 28, 29
- LIDO** Lightweight Information Describing Objects. 28, 30
- NPM** Node Package Manager. 46
- STC** Space-Time Cube. 3
- SVG** Scalable Vector Graphics. 46

UI User Interface. 11

VA Visual Analytics. 24

WebGL Web Graphics Library. 5, 46

XML Extensible Markup Language. 28

Appendix: User Tasks and Questions

Your answers don't have to be exact. The focus lies in the exploration of the data within the cube. Before starting the questions, please take a quick look at the application and possible interactions. In the cube-view (right) you perform selection action with the left mouse button and de-select with the right mouse button. In the detail-view (left) you can perform union and intersection operations on the categories. Also, the result of a selection will be displayed here. Please think out loud and if you have any questions, please ask.

Treemap Style

In this view, the data is split into various time-ranges divided into layers. Each layer is visualized as a treemap (based on categories) containing the data collected until this point in time. Each layer is also scaled to the amount of data it contains to the total number of objects contained in the whole cube. All layers are chronologically stacked to form a cube.

	Question	Answer
T1	Which category has the highest number of total objects?	
T2	Which category has the highest number of total objects, before the year 1000?	
T3	How many objects are classified as "Schmuck"?	
T4	How many objects are classified as "Schmuck" and "Glyptik"?	
T5	Name one category that has grown over the whole timeline	
T6	Compare 2 categories, "Gefäß" and "Glyptik", which grow faster (more items in shorter time) over the whole timeline?	
T7	How many objects are in the cube?	
T8	How many objects are in the collection from year -1000 to 1000?	
T9	How many objects were created from year -1000 to 1000?	
T10	Identify the time-range in which the most objects were created (before the year 1000)	

Matrix Style

In this view, the data is split into various time-ranges divided into layers. Each layer is visualized as a matrix (based on categories and collections) containing the data collected until this point in time. Each cube within each layer is scaled to the amount of data it contains to the total number of objects contained in the whole cube. All layers are chronologically stacked to form a cube.

	Question	Answer
M1	How many objects are in the "Kunstkammer" collection?	
M2	Which collection contains the widest variety of categories?	
M3	Which category is present in the most collections?	
M4	In which collection are the most objects with the category "(nicht vergeben)"?	
M5	How many objects are classified as "Schmuck"?	
M6	How many objects are classified as "Schmuck" and "Glyptik"?	
M7	How many objects are in the cube?	
M8	How many objects are in the collection from year -1000 to 1000?	
M9	How many objects were created from year -1000 to 1000?	

Rating Questions

These questions help evaluate the effectiveness and how intuitive the cube works in communicating a collection. Please answer truthfully and avoid the neutral answer as much as possible, only if you can not decide.

--	strongly disagree
-	disagree
-/+	neutral
+	agree
++	strongly agree

	Question	--	-	-/+	+	++
R1	Visualization supports the exploration of the collection	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R2	Visualization and controls are intuitive	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R3	Union and intersection operations are difficult to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R4	The visual representation of the element when performing an intersection operation is intuitive	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R5	The axis helped me navigate in the space-time-cube	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R6	I had problems interacting with the space-time-cubes controls	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R7	The selection of a time-frame was easy to understand	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
R8	The scaling option helped me solving tasks	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Open Questions

If you have any further thoughts, feedback or preferences, please write them down here.

	Question	Answer
O1	What would you change?	
O2	Did you find anything interesting in the collection?	
O3	Is there a dataset or collection you would like to explore?	
O4	What controls or elements were not helpful for you or not intuitive (e.g. clicking, timeline, ...)?	
O5	Which of the visualization would you prefer?	<input type="checkbox"/> treemap <input type="checkbox"/> treemap-flat <input type="checkbox"/> square-area

Appendix: Answered Tasks

This appendix contains the answers to the tasks from the user study. Each table shows the answers of one visualization method we previously presented. Table 1 contains the additional information which exact treemap style was used on this user. User 10 used a tablet to interact with the Culture Cube interface.

Table 1: Answers to the treemap tasks

User	Style	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10
1	treemap	Gemälde	Grabaustattung	385	12	Gefäß	Glyptik	9164	3645	2789	-242 to 24
2	treemap_flat	Gemälde	3714	385	12	Gerät	Glyptik	9164	3707	2849	-100 to 450
3	treemap	Gemälde	Grabaustattung	385	12	Gefäß	Gefäß	9164	3723	2965	-100 to 450
4	treemap_flat	Gemälde	Grabaustattung	385	12	Gefäß	Glyptik	9164	3721	2863	-265 to 340
5	treemap	Gemälde	Grabaustattung	385	12	Rundplastik, Schmuck, Gefäß, Gerät	Beide	9164	3723	3059	-265 to 340
6	treemap	Gemälde	Gefäss	385	12	Schmuck	Gefäß	9164	3714	2856	-100 to 400
7	treemap	Gemälde	Gefäss	385	12	Rundplastik, Schmuck, Gefäß	Glyptik	9164	3711	2853	-100 to 450
8	treemap_flat	Gemälde	Grabaustattung	385	12	Gefäß	Glyptik	9164	3711	2853	0 to 500
9	treemap	Gemälde	Gefäss	385	12	Gefäß	Glyptik	9164	3711	2853	-100 to 450
10	treemap_flat	Gemälde	Grabaustattung	385	12	Schmuck	Gefäß	9164	3711	2845	-265 to -340
11	treemap_flat	Gemälde	Gefäss	385	12	Schmuck	Glyptik	9164	3707	2849	-100 to 450
12	treemap	Gemälde	Grabaustattung	385	12	Gerät, Gefäß, Schmuck	Gemälde	9164	3774	2856	-100 to 450
13	treemap_flat	Gemälde	Grabaustattung	385	12	Gerät	Glyptik	9164	3700	2800	-100 to 450
14	treemap_flat	Gemälde	Relief	385	12	Amulett	Glyptik	9164	3721	2863	-40 to 40
15	treemap	Gemälde	Gefäss	385	12	Gerät	Gemälde	9164	3707	2849	-800 to -500

Table 2: Answers to the matrix tasks

User	M1	M2	M3	M4	M5	M6	M7	M8	M9
1	2129	Sammlungen Schloss Ambras	Gemälde	Münzkabinett	385	12	9164	3645	2789
2	2129	Sammlungen Schloss Ambras	Gemälde	Münzkabinett	385	12	9164	3711	2856
3	2129	Sammlungen Schloss Ambras	Gemälde	Ägyptisch - Oriental Sammlung	385	12	9164	3714	2656
4	2129	Sammlungen Schloss Ambras	Gemälde	Münzkabinett	385	12	9164	3714	2857
5	2129	Kunstkammer	Gemälde	Münzkabinett	385	12	9164	3707	2849
6	2129	Sammlungen Schloss Ambras	Gemälde	Münzkabinett	385	12	9164	3711	2854
7	2129	Kunstkammer	Gefäß	Münzkabinett	385	12	9164	3711	2853
8	2129	Kunstkammer	Gemälde	Münzkabinett	385	12	9164	3711	2853
9	2129	Sammlungen Schloss Ambras	Gemälde	Münzkabinett	385	12	9164	3701	2834
10	2129	Ägyptische	Schmuck	Münzkabinett	385	12	9164	3712	2854
11	2129	Sammlungen Schloss Ambras	Gemälde	Ägyptisch - Oriental Sammlung	385	12	9164	3711	2879
12	2129	Kunstkammer	Schmuck	Münzkabinett	385	12	9164	3174	2858
13	2129	Sammlungen Schloss Ambras	Gemälde	Münzkabinett	385	12	9164	3700	2845
14	2129	Sammlungen Schloss Ambras	Gemälde	Münzkabinett	385	12	9164	3715	2847
15	2129	Sammlungen Schloss Ambras	—	—	385	12	9164	3707	2849

Bibliography

- [Aigner et al., 2011] Aigner, W., Miksch, S., Schumann, H., and Tominski, C. (2011). *Visualization of Time-Oriented Data*. Springer Publishing Company, Incorporated, 1st edition.
- [Alsallakh et al., 2013] Alsallakh, B., Aigner, W., Miksch, S., and Hauser, H. (2013). Radial sets: Interactive visual analysis of large overlapping sets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2496–2505.
- [Alsallakh et al., 2014] Alsallakh, B., Micallef, L., Aigner, W., Hauser, H., Miksch, S., and Rodgers, P. (2014). Visualizing sets and set-typed data: State-of-the-art and future challenges. In Borgo, R., Maciejewski, R., and Viola, I., editors, *Eurographics conference on Visualization (EuroVis)– State of The Art Reports*, pages 1–21. Eurographics, Eurographics.
- [Alsallakh and Ren, 2017] Alsallakh, B. and Ren, L. (2017). Powerset: A comprehensive visualization of set intersections. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):361–370.
- [Autere and Vakkari, 2011] Autere, R. and Vakkari, M. (2011). Towards cross-organizational interoperability: The lido xml schema as a national level integration tool for the national digital library of finland. In *International Conference on Theory and Practice of Digital Libraries*, pages 62–68. Springer.
- [Bach et al., 2017] Bach, B., Dragicevic, P., Archambault, D., Hurter, C., and Carpendale, S. (2017). A descriptive framework for temporal data visualizations based on generalized space-time cubes. 36(6):36–61.
- [Bach et al., 2014] Bach, B., Pietriga, E., and Fekete, J.-D. (2014). Visualizing Dynamic Networks with Matrix Cubes. In *Proceedings of the 2014 Annual Conference on Human Factors in Computing Systems (CHI 2014)*, pages 877–886, Toronto, Canada. ACM.
- [Bostock et al., 2011] Bostock, M., Ogievetsky, V., and Heer, J. (2011). D³ data-driven documents. *IEEE transactions on visualization and computer graphics*, 17(12):2301–2309.

- [Dörk et al., 2011] Dörk, M., Carpendale, S., and Williamson, C. (2011). The information flaneur: A fresh look at information seeking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 1215–1224, New York, NY, USA. ACM.
- [Dörk et al., 2017] Dörk, M., Pietsch, C., and Credico, G. (2017). One view is not enough. *Information Design Journal*, 23(1):39–47.
- [europeana, 2016] europeana (2016). Definition of the europeana data model v5.2.7.
- [Glinka et al., 2017] Glinka, K., Pietsch, C., and Dörk, M. (2017). Past visions and reconciling views: Visualizing time, texture and themes in cultural collections. 11.
- [Lex et al., 2014] Lex, A., Gehlenborg, N., Strobel, H., Vuillemot, R., and Pfister, H. (2014). Upset: Visualization of intersecting sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):1983–1992.
- [Mayr et al., 2016] Mayr, E., Federico, P., Miksch, S., Schreder, G., Smuc, M., and Windhager, F. (2016). Visualization of cultural heritage data for casual users. In *IEEE VIS Workshop on Visualization for the Digital Humanities*.
- [Miksch and Aigner, 2014] Miksch, S. and Aigner, W. (2014). A matter of time: Applying a data-users-tasks design triangle to visual analytics of time-oriented data. *Computers & Graphics, Special Section on Visual Analytics*, 38:286–290.
- [Munzner, 2009] Munzner, T. (2009). A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–928.
- [Nielsen and Landauer, 1993] Nielsen, J. and Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, pages 206–213, New York, NY, USA. ACM.
- [Purday, 2009] Purday, J. (2009). Think culture: Europeana.eu from concept to construction. *The Electronic Library*, 27(6):919–937.
- [Shneiderman, 1992] Shneiderman, B. (1992). Tree visualization with tree-maps: 2-d space-filling approach. *ACM Trans. Graph.*, 11(1):92–99.
- [Shneiderman, 1996] Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343.
- [Sondag et al., 2018] Sondag, M., Speckmann, B., and Verbeek, K. (2018). Stable treemaps via local moves. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):729–738.

- [Tufte, 1990] Tufte, E. R. (1990). *Envisioning information*. Graphics press.
- [Verbert et al., 2014] Verbert, K., Parra, D., and Brusilovsky, P. (2014). The effect of different set-based visualizations on user exploration of recommendations. In *CEUR Workshop Proceedings*, volume 1253, pages 37–44. University of Pittsburgh.
- [Walsh and Hall, 2015] Walsh, D. and Hall, M. M. (2015). Just looking around: Supporting casual users initial encounters with digital cultural heritage. In *Proceedings of the First International Workshop on Supporting Complex Search Tasks co-located with the 37th European Conference on Information Retrieval (ECIR 2015), Vienna, Austria, March 29, 2015*.
- [Whitelaw, 2015] Whitelaw, M. (2015). Generous interfaces for digital cultural collections. *Digital Humanities Quarterly*, 9(1):38.
- [Windhager et al., 2016a] Windhager, F., Federico, P., Mayr, E., Schreder, G., and Smuc, M. (2016a). A review of information visualization approaches and interfaces to digital cultural heritage collections. In *Proceedings of the 9th Forum Media Technology (FMT2016)*, page 74–81. St. Pölten University of Applied Sciences, Institute of Creative Media Technologies.
- [Windhager et al., 2016b] Windhager, F., Mayr, E., Schreder, G., Smuc, M., Federico, P., and Miksch, S. (2016b). Reframing cultural heritage collections in a visualization framework of space-time cubes. In *Proceedings of the 3rd International Workshop on Computational History (HistoInformatics2016)*, volume 1632 of *CEUR Workshop Proceedings*, pages 20–24. CEUR-WS, CEUR-WS.