LitExplorer: A Visual Browser for Literature Collections

Project Report

Magne Tenstad, Azra Bajramovic, Una Ibrahimpasic and Eemil Hukkanen

706.057 Information Visualisation 3VU SS 2024
Graz University of Technology

1 July 2024

Abstract

This project report presents LitExplorer, a visual browser designed for efficiently navigating and exploring literature collections. LitExplorer uses visualizations to facilitate intuitive interaction with academic papers, surveys, and related literature. Drawing inspiration from the SurVis project [Beck 2016], LitExplorer enhances the user experience by providing a visually rich and organized representation of literature data. Developed using modern web technologies, Vite and Svelte, LitExplorer ensures high performance, responsiveness, and a seamless user interface. This report details the conceptual framework, design choices, and implementation strategies of LitExplorer.

Contents

List of Figures

Chapter 1

Introduction

In the expanding world of academic research, the volume of literature is constantly growing. This makes it increasingly challenging for researchers to keep up with the latest developments in their fields. Traditional methods of literature review and exploration are often done with text-based search engines and libraries. To address this issue, visual browsers for literature collections have emerged as powerful tools, offering an efficient alternative to navigate and analyze large collections of academic work.

This project is inspired by the SurVis project [Beck 2016], which demonstrated the value of categorizing academic papers in an organized and accessible manner. Building on this foundation, LitExplorer aims to enhance the user experience by incorporating modern web technologies, innovative design principles, and additional visualizations.

The primary objective of LitExplorer is to provide researchers, students, and academics with a powerful tool for literature exploration. It reads a collection of bib entires in BibLaTeX format [Lehman et al. 2024]. By leveraging a set-typed visualization for paper categories, LitExplorer aims to facilitate intuitive navigation and discovery of relevant literature, enable efficient organization and categorization of academic papers and surveys, and improve the overall user experience through a clear interface.

LitExplorer is developed using Vite [Vite 2024] and Svelte [Svelte 2024], two modern web technologies. Vite is known for its fast build times and efficient development environment that ensures high performance. Svelte is a modern front-end framework, which allows for highly responsive and dynamic user interfaces with minimal overhead. Together, these technologies provide the backbone for a developing a robust and scalable web application.

This project report begins by introducing three existing visual browsers for academic literature. Then, it explains the data preparation necessary for LitExplorer. Finally, the main chapter describes the implementation of the software and each feature in detail.

Chapter 2

Related Work

Three existing visual browsers for academic literature inspired the design of LitExplorer: Treevis [Schulz 2011], SurVis [Beck 2016], and BioVis Explorer [Kerren et al. 2017b; Kerren et al. 2017a].

2.1 Treevis

Treevis [Schulz 2011] is a visual browser for tree visualization techniques. The browser was inspired by a survey done on tree visualization techniques in 2010 [Schulz et al. 2011], in which the authors collected and categorized existing tree visualization techniques. Treevis was custom built for this survey. Schulz wanted Treevis to allow academics to effortlessly browse through the existing library of publications, to aid them in their own research.

Three different means of exploration were implemented, as can be seen in Figure 2.1. Firstly, a visual representation of the publications as a grid of image thumbnails allows authors to visually search for particular techniques. Secondly, a search box supports text search by author name, visualization technique name, publication title, year of publication, and even supports use of JavaScript Regular Expressions. Thirdly, button filters correspond to the three categories from the survey: dimensionality (2D, 3D, hybrid), edge representation (explicit, implicit, hybrid), and node alignment (axis-parallel, radial, free). Finally, once a user has found a publication of interested, they can click on the thumbnail (of the publication), and a pop-up display further details: including tile, authors, link to the publication, and the symbols which indicate to which categories the visualization belongs to (eg. 2D, implicit, axis-parallel).

2.2 SurVis

SurVis [Beck 2016] SurVis is a web-based system designed to facilitate the exploration of literature collections with a focus on usability, manual collection, and interactive selection. The system prioritizes simplicity over having many features, ensuring it is accessible to readers without extensive training. Manual collection is emphasized over automatic data extraction to maintain high data quality. Every piece of information presented in SurVis can be selected, allowing users to filter and explore related objects (publications).

The user interface of SurVis consists of several components: a list of publications, word clouds, a timeline, and versatile selectors. These can be seen in Figure 2.2. The list of publications, displayed on the right side of the screen, allows for exploration of individual records, providing essential details such as titles, authors, abstracts, and keywords. On the left side, word clouds summarize keywords and other meta-information, while a timeline above them gives an overview of the publication history and highlights the most cited works. SurVis uses standard publication records and relies on structured keywords assigned by curators to make the literature easy to navigate. These keywords are categorized to address different aspects of the publications, enhancing the organization of the surveyed field. Citation

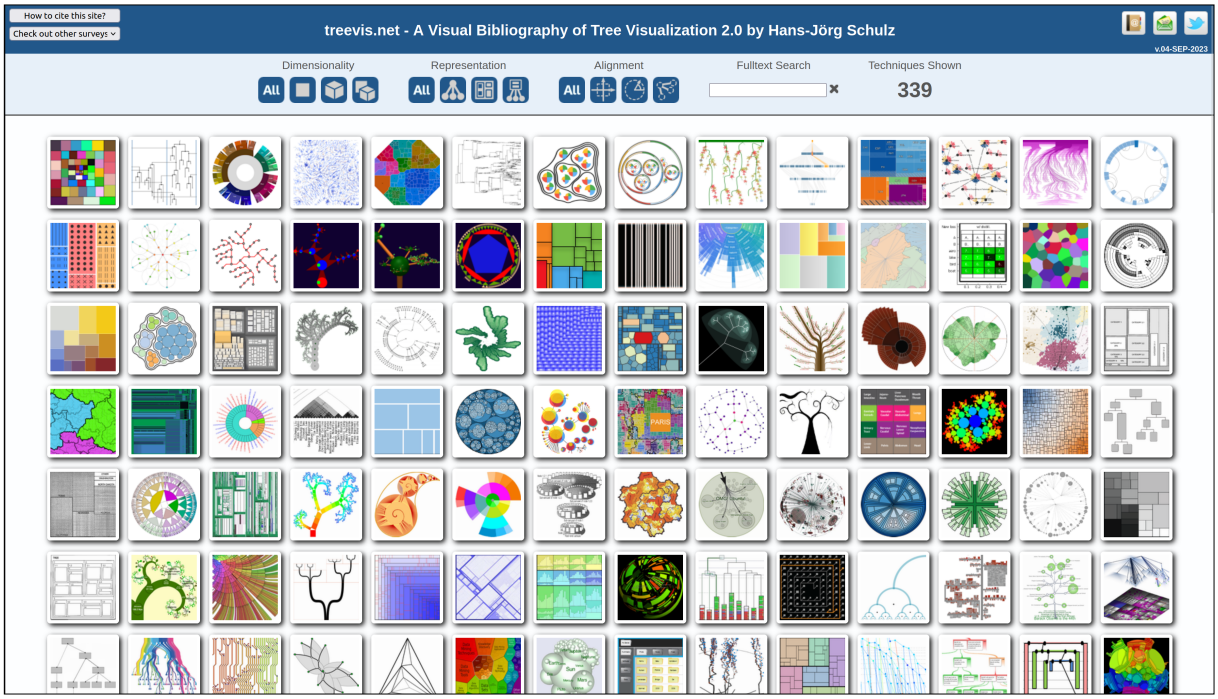


Figure 2.1: Treevis.net. [Screenshot taken by Azra Bajramovic.]

data is also integrated to reveal relationships between publications and indicate their impact within the collection. The system supports various selector types for filtering and ranking publications based on keywords, authors, series, clusters, publication years, and citation relationships. These selectors allow users to interactively connect different components of the interface, facilitating a comprehensive analysis of the literature data. Sparklines, which are word-sized visualizations, and other visual aids are used to represent selector agreements and enrich the user interface, making it easier to analyze and compare entities.

2.3 BioVis Explorer

BioVis Explorer [Kerren et al. 2017a] is an interactive web-based visualization tool that provides an electronic review of published biological visualization methods using browsing and associations with related methods. The tool is designed using a similarity map (galaxy metaphor), offering a coherent view of publications. As can be seen in Figure 2.3, its interface includes a side panel for filters, statistics, and a temporal histogram chart, while the main view displays thumbnails representing individual visualization techniques or papers. Thumbnails are positioned based on the mutual dissimilarity of entries, allowing users to quickly overview similar visualization methods. The similarity map is generated through multidimensional scaling (MDS), considering factors like publication year, assigned categories (via the Jaccard index), and sets of authors.

BioVis Explorer supports user interactions such as zooming, panning, and detailed information dialogs, where users can see publication titles, categories, and bibliographical information. Similar techniques are listed by decreasing similarity values. Filtering options in the side panel include keyword search, publication year, and a broad taxonomy divided into biological data types, data properties, and visualization tasks. Users can also adjust sliders for specific distance factors, affecting the arrangement of thumbnails, and set similarity thresholds for interactive exploration.

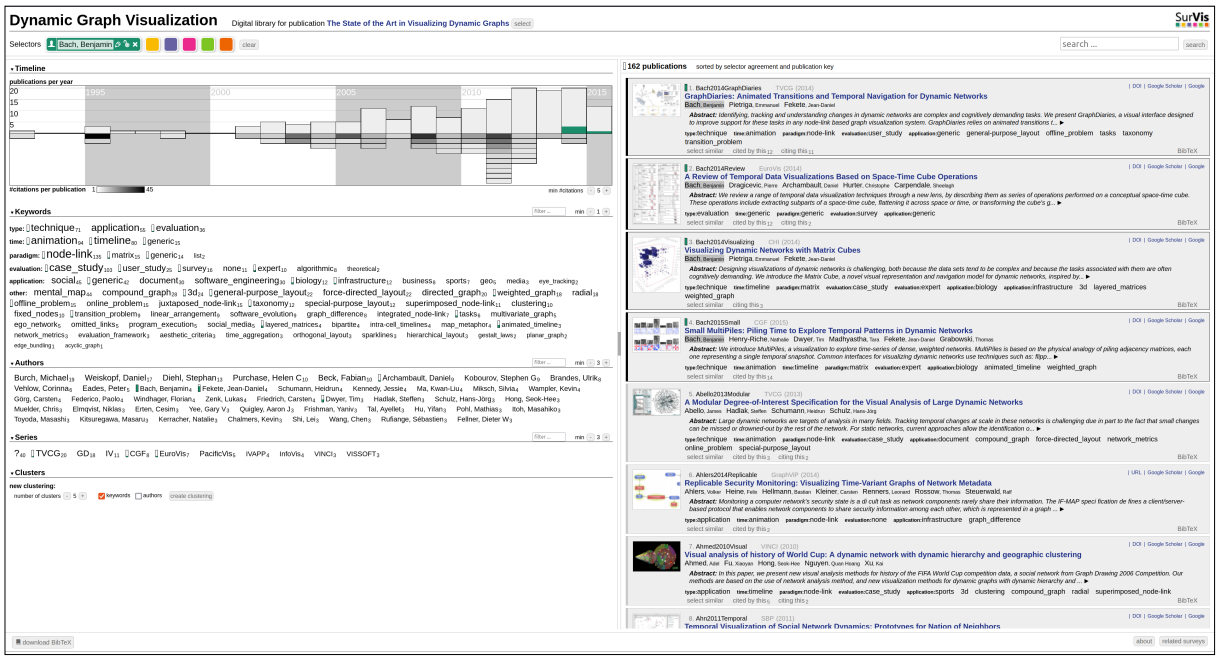


Figure 2.2: SurVis showing a survey about Dynamic Graph Visualization. [Screenshot taken by Azra Bajramovic.]

Figure 2.3: BioVis Explorer. [Screenshot taken by Azra Bajramovic.]

Chapter 3

Data Preparation

3.1 Formatting bib Entries

Bibliographic data should be prepared in the form of BibLaTeX entries [Lehman et al. 2024] in a single file called `references.bib`. When formatting bib entries, it is essential to ensure that all necessary fields are included for proper citation and reference management. For typical bibliographic entries, these include `author`, `title`, `date`, `doi`, and `abstract`. Two custom fields were added for LitExplorer: `category` (of visualization technique) and `name` (of visualization technique).

To streamline this process, automatic validation is done in Python, which checks for the presence of these fields. If any fields are missing, the required information is fetched from the CrossRef API [Polischuk 2020]. In cases where the API does not provide the necessary details, the fields have to be filled in manually to ensure completeness and accuracy.

3.2 Image Extraction

The image extraction process is done by parsing the `.bib` file to identify the relevant entries. For each publication entry, the corresponding PDF document is downloaded using the Sci-Hub API [scihub.py 2020]; if this is not possible, the PDF must be obtained manually. Once the PDF is acquired, all images within the document are extracted using PyMuPDF [PyMuPDF 2024], an efficient library for PDF processing. If PyMuPDF is unable to extract the images, manual extraction is necessary. After extracting the images, the specific image needed is identified by manually noting its x-ref, which represents the numeric reference of the figure (represented in the chosen image) in the corresponding publication PDF file. This selected image is then converted into a thumbnail format using the Pillow library [Pillow 2024], ensuring it is suitable for display purposes.

3.3 Image Permissions

Since use of images extracted from the PDFs does not necessarily fall under fair use, it is imperative to obtain proper permissions from the copyright owners. To address this, requests for bulk permissions are typically sent to publishers, such as eg. IEEE, to cover multiple images at once. Additionally, requests can be sent to individual authors for permission to use specific images.

Due to lack of image permissions, the current survey of set visualization used to illustrate LitExplorer is hosted publicly without images. However, images are included in the local version of the survey, where access is restricted.

Chapter 4

LitExplorer

LitExplorer is a browser for collections of literature. Although not a novel concept, LitExplorer has been designed with newer technologies and with the aim of improving the exisiting system which inspired its creation: Treevis.net [Schulz 2011], SurVis [Beck 2016], and BioVis [Kerren et al. 2017a] (see Chapter 2).

In order to have a demo dataset, a previously created survey of set visualization techniques was used [Ibrahimpasic et al. 2024], albeit without thumbnail images in the public version due to lack of permissions.

4.1 Implementation

There were a number of considerations which had to be kept in mind during the implentation of LitExplorer. SurVis was implemented using JavaScript with JQuery [Beck 2016]. To improve upon this, LitExplorer was completely rewritten in TypeScript using Svelte [Svelte 2024], Vite [Vite 2024], and Tailwind [Tailwind 2024]. Svelte is an open-source JavaScript framework that enables developers to create interactive web applications efficiently. It provides a component-based architecture and intuitive reactivity. Tailwind is a utility-first CSS framework, frequently used with Svelte, which provides a unique approach to styling based on CSS classes.

Secondly, due to not having obtained all of the permissions of use for the thumbnails in LitExplorer, it was needed to implement a fallback, such that the developer easily can decide whether images should be presented in the built application or not. For this, the npm script `build-no-img` was added, which complements the standard `build` script. Thirdly, LitExplorer supports both currently relevant BibLaTeX fields and older (deprecated) BibTeX fields for the input reference files.

4.2 User Interface

As previously mentioned, the design of LitExplorer was inspired by existing comparable systems. As can be seen in Figure 4.1, LitExplorer is composed of a number of panels. On the lefthand side, there are four different panels: Timeline, Categories, Authors, and Names, which allow for filtering through the literature collection. On the righthand side, there is a search box above a list of currently selected publications. The list can be switched between a Table view and a Grid view. Where available, a thumbnail image is shown for each publication. It is also possible to export the BibLaTeX entries of the currently selected publications.

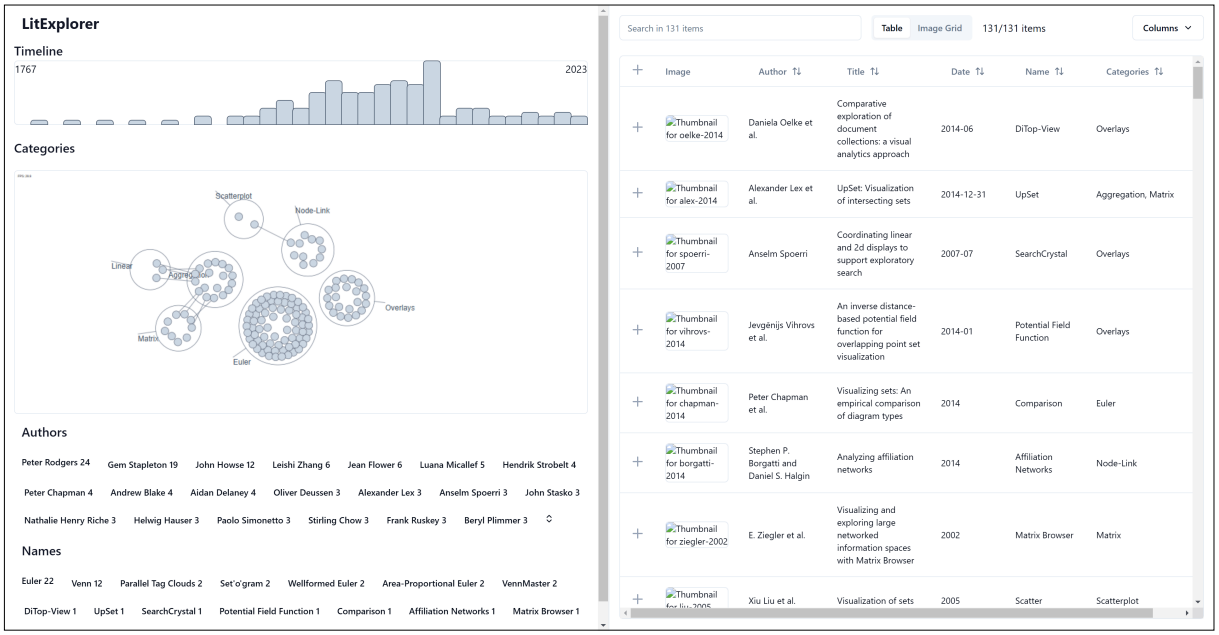


Figure 4.1: The interface of LitExplorer. [Image used with kind permission of Keith Andrews.]

4.2.1 Timeline

The Timeline panel, shown in Figure 4.2, is visualized as a bar chart, with each bar corresponding to a year within the range covered by the literature collection. When a user hovers over a year bar, it is highlighted with a red outline, while selecting a year displays it with a dark gray outline.

4.2.2 Categories

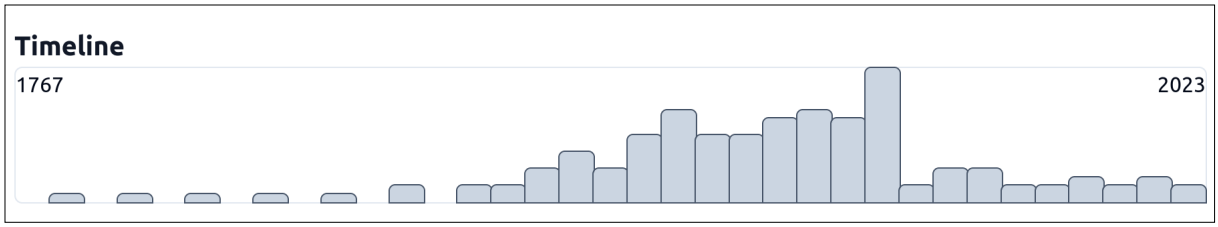
The Categories panel, shown in Figure 4.3, displays a collection of bubbles representing different categories, each visually distinct. When a category is selected, it is highlighted with a dark gray outline, accompanied by gray shading indicating corresponding years in the Timeline panel. Individual publication objects within a category bubble are shaded gray when their time range aligns with the selected period in the Timeline panel. Hovering over a publication object in the Categories panel highlights the publication in red, accompanied by a tooltip providing additional information.

4.2.3 Authors and Names

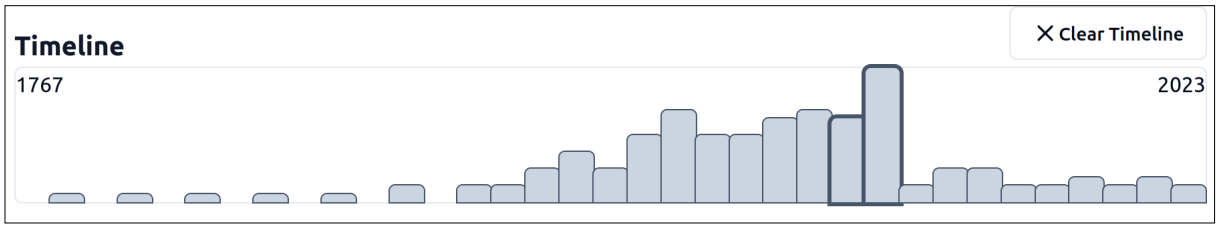
In the Authors panel and Names panel, shown in Figure 4.4, the elements are represented as individual buttons with the name of the author and the name of the visualization technique respectively, and the number of publications associated with the element. When a user hovers over an author or technique name, a distinct red outline highlights the element, providing immediate visual feedback. Upon selection, the chosen element is prominently outlined in a dark gray shade, accompanied by subtle gray shading to denote its selected status.

4.2.4 Search Box

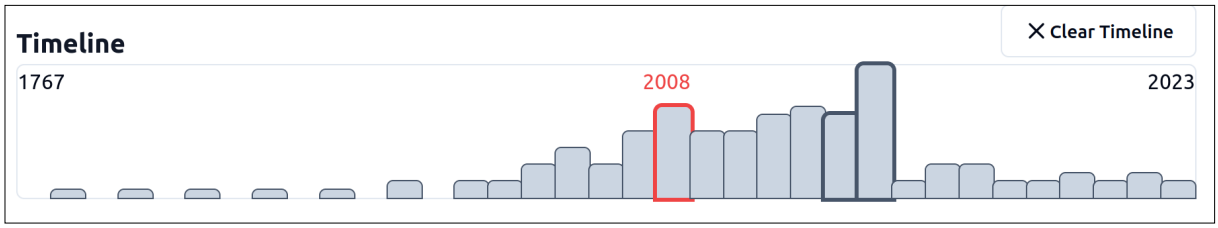
The search box feature allows for easy text search of the publication entries in the literature collection. Users can search by author name, title of publication, name of visualization technique, etc.



(a) Initial appearance.



(b) Selected range (dark gray outline).



(c) Hovered element (red outline).

Figure 4.2: The LitExplorer Timeline panel. [Images created by Azra Bajramovic.]

4.2.5 Publications List

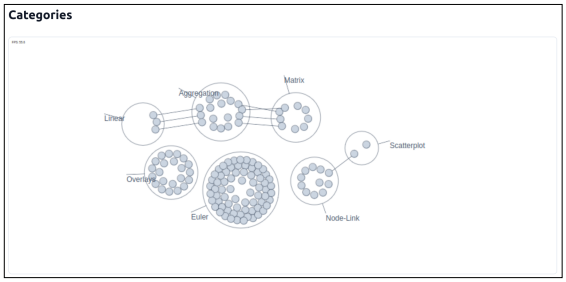
The list of publications on the righthand side of the main window can be viewed in two ways: a Table view with one row per publication, or a Grid view. In the Table view, the displayed columns can be chosen using the Columns dropdown in the right top corner of the Table view.

4.2.6 Portfolio

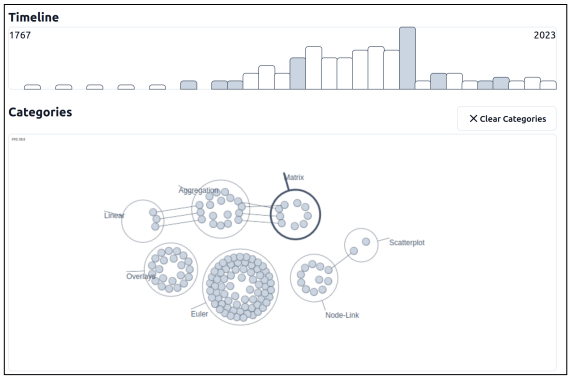
A further feature of LitExplorer is the ability for the user to collect selected publications into a portfolio, which can then be exported as a BiBLaTeX bib file. Publications can be added to the portfolio using either the plus button in the publication row in Table view or the Add to Portfolio button in the Publication Details popup window.

4.2.7 Publication Details

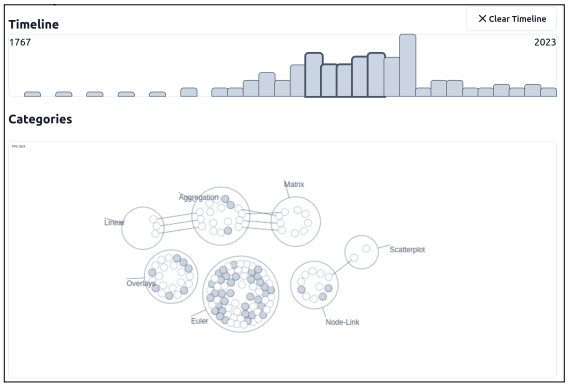
Left-clicking on a publication opens up the Publication Details popup window. It contains the following details about the publication: title, author(s), abstract, thumbnail image (where available), and BibLaTex entry. From this pop-up window the user can copy the BibLaTex entry to clipboard, add the publication to the portfolio, or visit the web page (DOI) of the publication.



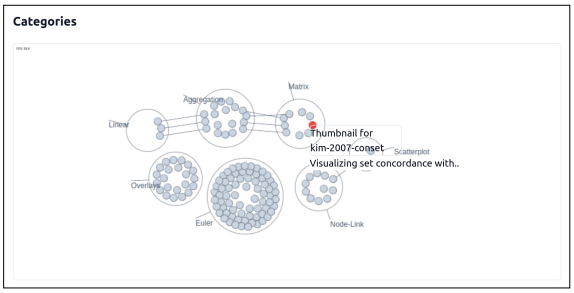
(a) Initial appearance.



(b) One category (Matrix) is selected. Corresponding years are shaded grey in the Timeline panel.

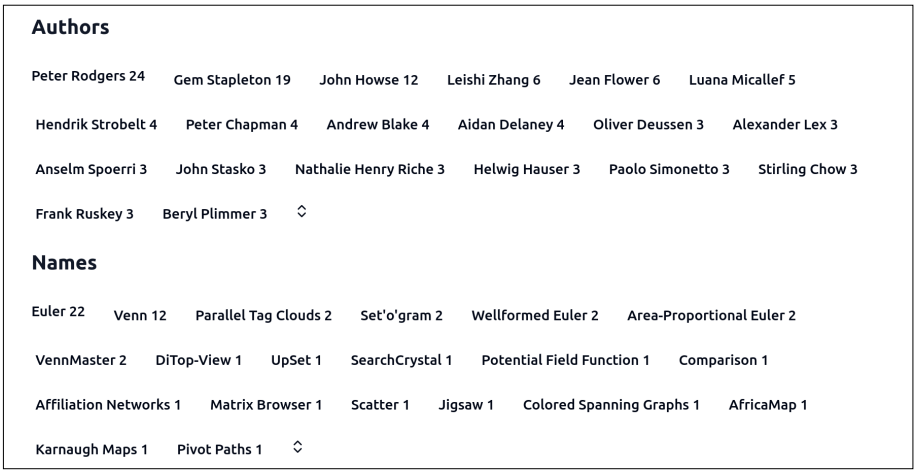


(c) Selecting a range in the Timeline panel. Publications outside the time range are now hollow.

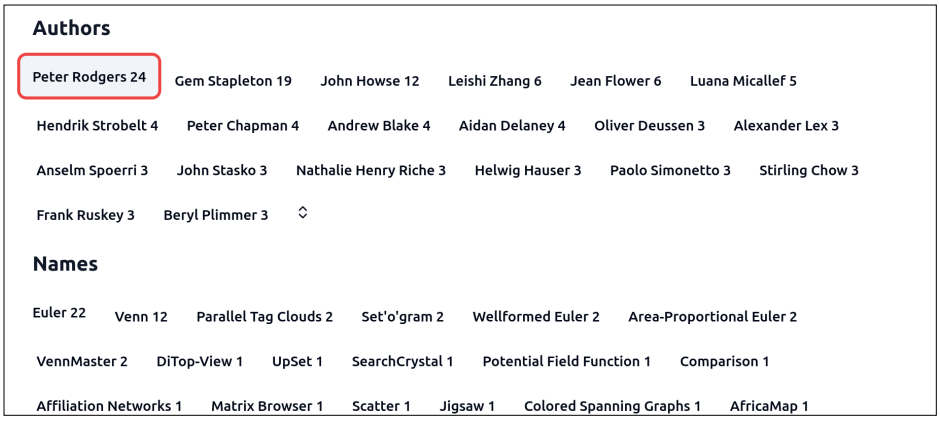


(d) Hovering over one publication in the Categories panel highlights it in red and displays a tooltip.

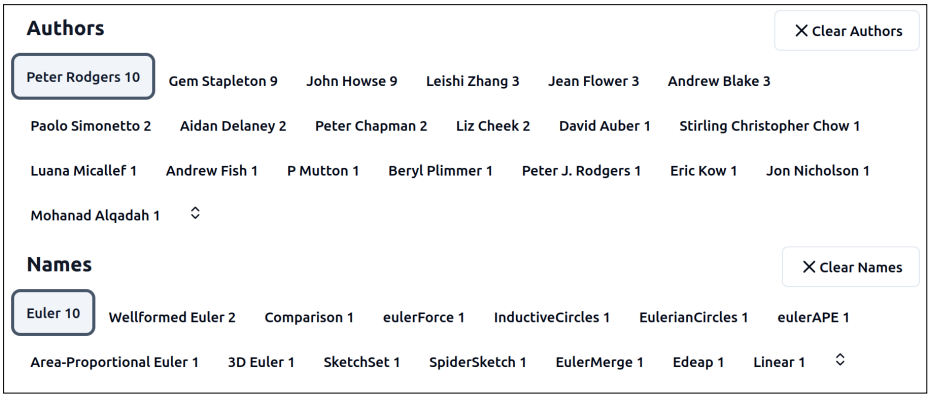
Figure 4.3: The LitExplorer Categories panel, in combination with the Timeline. [Images created by Azra Bajramovic.]



(a) Initial appearance of Authors and Names panels, nothing is selected.

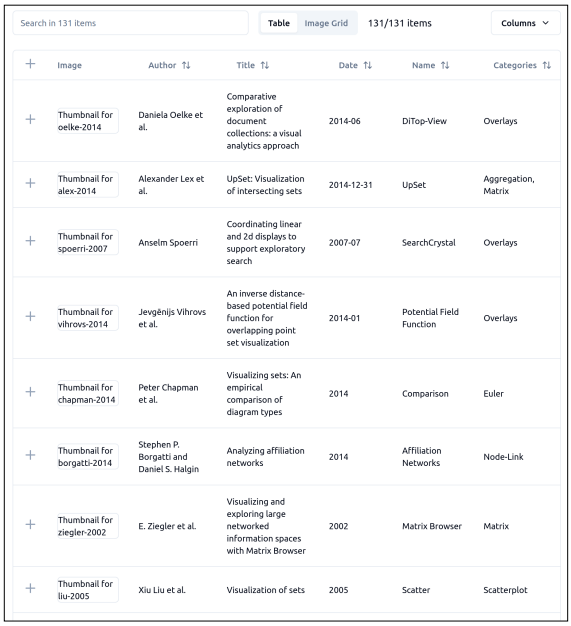


(b) Hovering over an author or technique name highlights it in red.

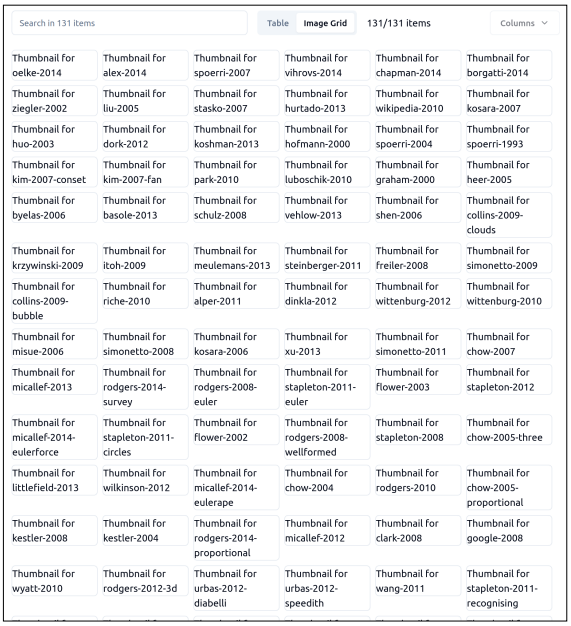


(c) Selecting an author or techique name highlights in grey with a dark grey border.

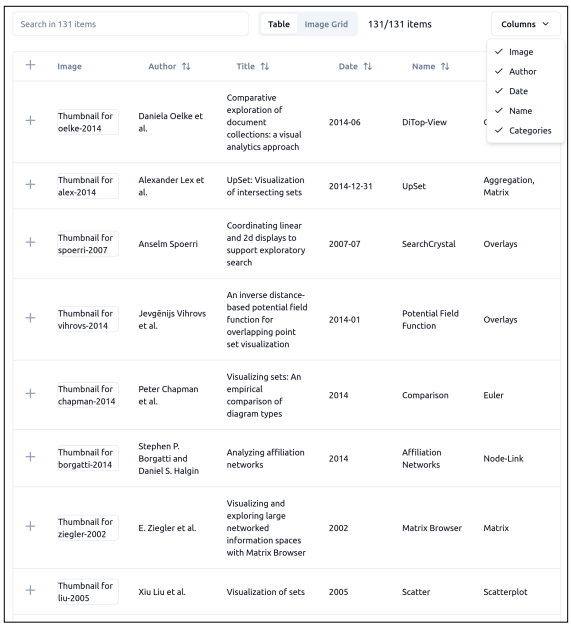
Figure 4.4: The LitExplorer Authors and Names panels. [Images created by Azra Bajramovic.]



(a) Table view.



(b) Grid view.



(c) Columns dropdown to choose columns.

(d) Some columns removed.

Figure 4.5: The Table view and Grid view of publications. Where permissions are available for images, thumbnails would be shown. [Images created by Azra Bajramovic]

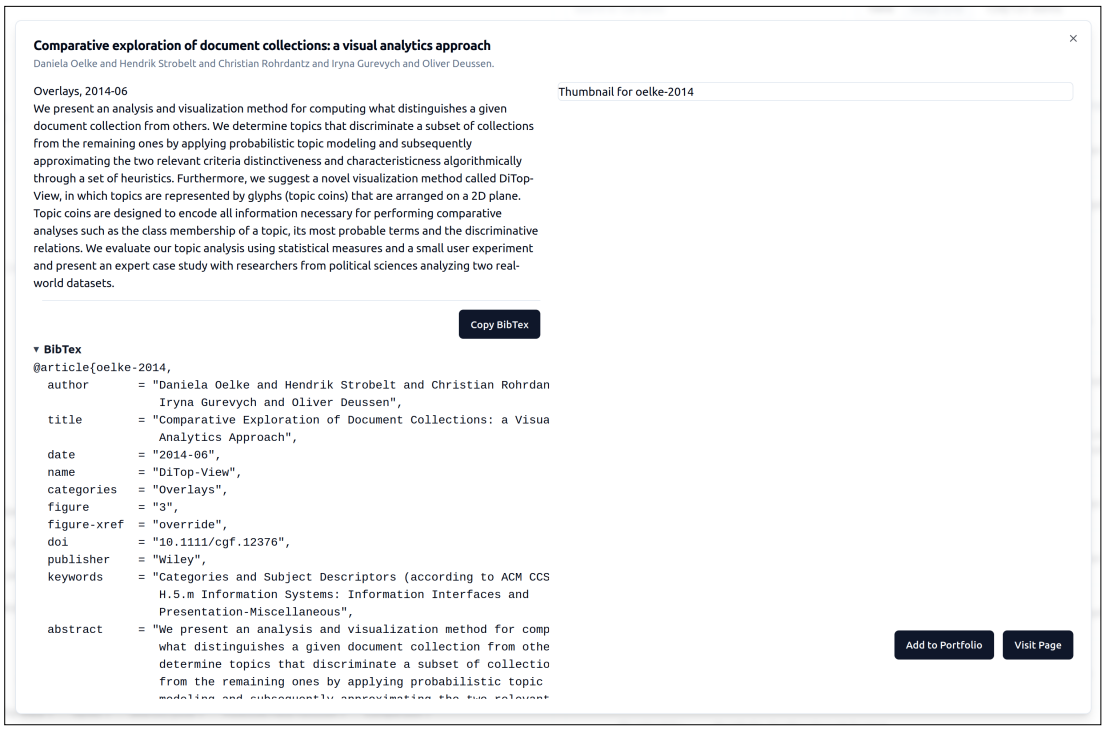


Figure 4.6: The Publication Details popup window for a publication in LitExplorer contains title, author(s), abstract, thumbnail image, and BibTex entry. [Image created by Azra Bajramovic.]

Chapter 5

Concluding Remarks

This project report describes LitExplorer, a comprehensive system for managing, visualizing, and interacting with a collection of literature references and associated images. The development of LitExplorer and the careful process of data preparation produce easy-to-use visual browser for a literature collection, which can significantly enhance the way researchers access and manage bibliographic data.

The goal of LitExplorer was to build a visual browser for literature collections with modern web technology, combining some of the features of existing systems, like filtering by timeline, categories, authors and names. Other notable features include text search, publication overview lists in tabular and grid form, and the possibility to collect publications into a portfolio for export.

In the future, LitExplorer could be extended in various ways, including the possibilty to dynamically add publications to the database (without rebuilding the application), and options to specifically include keywords, co-citation analysis, and perhaps generate a similarity map where similar papers are grouped together.

Bibliography

Beck, Fabian [2016]. *Visual Analysis and Dissemination of Scientific Literature Collections with SurVis*. IEEE Transactions on Visualization and Computer Graphics 22.1 (2016), pages 180–189. doi:10.1109/TVCG.2015.2467757 (cited on pages I, 1, 3, 9).

Ibrahimpasic, Una, Eemil Hukkanen, Magne Tenstad, and Azra Bajramovic [2024]. *Set Visualization: A Survey of Set Visualization Techniques*. Survey Paper. Graz University of Technology, 13 May 2024. https://courses.isds.tugraz.at/ivis/surveys/ss2024/ivis-ss2024-g2-survey-setvis.pdf (cited on page 9).

Kerren, Andreas, Kostiantyn Kucher, Yuan-Fang Li, and Falk Schreiber [2017a]. *BioVis Explorer: A Visual Guide for Biological Data Visualization Techniques*. PLoS One 12.11 (01 Nov 2017). doi:10.1371/journal.pone.0187341 (cited on pages 3–4, 9).

Kerren, Andreas, Kostiantyn Kucher, Yuan-Fang Li, and Falk Schreiber [2017b]. *MDS-based Visual Survey of Biological Data Visualization Techniques*. EuroVis 2017 Posters (Barcelona, Spain). 12 Jun 2017. doi:10.2312/eurp.20171175 (cited on page 3).

Lehman, Philipp, Philip Kime, Audrey Boruvka, Joseph Wright, and Moritz Wemheuer [2024]. *BibLaTeX: Sophisticated Bibliographies in LaTeX*. 22 Mar 2024. https://ctan.org/pkg/biblatex (cited on pages 1, 7).

Pillow [2024]. *Python Imaging Library*. 17 Jul 2024. https://python-pillow.org/ (cited on page 7).

Polischuk, Patrick [2020]. *Crossref REST API*. 08 Apr 2020. https://crossref.org/documentation/retrieve-metadata/rest-api/ (cited on page 7).

PyMuPDF [2024]. *PyMuPDF*. 17 Jul 2024. https://github.com/pymupdf/PyMuPDF (cited on page 7).

Schulz, Hans-Jörg [2011]. *Treevis.net: A Tree Visualization Refrence*. IEEE Computer Graphics and Applications 31.6 (2011), pages 11–15. doi:10.1109/MCG.2011.103 (cited on pages 3, 9).

Schulz, Hans-Jörg, Steffen Hadlak, and Heidrun Schumann [2011]. *The Design Space of Implicit Hierarchy Visualization: A Survey*. IEEE Computer Graphics and Applications 17.4 (2011), pages 393–411. doi:10.1109/TVCG.2010.79 (cited on page 3).

scihub.py [2020]. *scihub.py: Python API for Sci-Hub*. 20 Jun 2020. https://github.com/zaytoun/scihub.py (cited on page 7).

Svelte [2024]. *Svelte: Cybernetically Enhanced Web Apps*. 17 Jul 2024. https://svelte.dev/ (cited on pages 1, 9).

Tailwind [2024]. *Tailwind CSS: A Utility-First CSS Framework for Rapidly Building Custom User Interfaces*. Tailwind Labs, 16 Jul 2024. https://tailwindcss.com/ (cited on page 9).

Vite [2024]. *Vite: Next Generation Frontend Tooling*. Evan You & Vite Contributors, 16 Jul 2024. https://vitejs.dev/ (cited on pages 1, 9).