Information Visualization Project Report

Team 36

Xinran Yang[12547425], Chuyi Tong[12663131], Guijing Xu[12946346], Futong Han[12581135], and Kai Zhang[12712469]

University of Amsterdam

Abstract. This project aims to visualize the OmniArt dataset to provide users with an application to acquire knowledge and gain insights from the art data. Various visualization techniques and design methodologies are applied in the design and implementation process. This application includes five main functions: bar chart timeline, artwork type tree, artist ranking list, theme word could and the artwork gallery. Different visualizations are combined and connected together in this system as a whole. The application also adopts a user-friendly UI design and supports various user interactions. A RESTful API of the dataset is implemented for advanced users as well.

Keywords: Information visualization · Art application · User interaction.

1 Introduction

There is a large amount of digital artistic data scattered on the web. Art applications such as WikiArt and Google Art Project can help us explore these artworks from different perspectives. Likewise, in this study, we introduce a web application, which is called the OmniART explorer, to give users a bird's-eye view of artworks from different centuries. In our application, users can explore the artistic data and images according to their creation years, general types and artists. We choose the OmniART dataset as our main dataset since it contains massive artistic data which can be used to make an interesting visual storytelling and help users gain insights into the art history.

There are 5 main functions for the application: a bar chart timeline which shows the proportion of different artwork types, an artwork type tree which shows genres of artworks, an artist ranking list which shows the top artists in a specific period, a word cloud from artwork descriptions, and a gallery for users to explore artworks in our database. To implement the application, we combined various techniques, such as Apache ECharts and Apache Spark. We also developed a web crawler to collect the basic information of artists (e.g. birthday, nationality) from Wikipedia and rank them based on their popularity, which gives the user an overview of these masters in the artistic domain.

In this paper, our project is introduced in four parts. In section 2, the dataset of this project is introduced and the problem domain of the task is demonstrated.

In section 3, the visualization design is explained in details with five main functions, with the application implementation and structure described. In section 4, the design is evaluated with perspectives from interaction design, visual analytics design storytelling design and visual thinking design. Knowledge gained from this visualization project is also showed in this part.

2 Data and Problem Domain

2.1 Dataset background

The dataset OmniART contains more than 430000 photographic reproductions of artworks with rich annotations (Strezoski and Worrying 2017). In specific, we extract information such as artists' full names, creation years and artwork types, then we reorganize the data in a visual-appealing and user-interactive way to support our storytelling. Meanwhile, we also use knowledge from Wikipedia to enrich our data. A web crawler is used to gather the information of artists mentioned in the OmniART dataset. Information like the artist's birthday, nationality and portrait are collected. We also count the reference lists of artists' wiki pages to compute their popularity and rank them.

2.2 Problem domain

Based on the dataset we chose, this project aims to extract knowledge from the OmniArt dataset. This system is designed for both art-related professionals and amateurs. A typical usage scenario is that the user can explore artworks in the artwork gallery according to different filters, such as timeline filter, genre type filter, or artist filter.

3 Visualization Design

3.1 Design Decision

The dataset we chose is OmniArt, which contains metadata and images of more than 1 million artwork. To gain knowledge from this huge dataset, we designed a system which contains five main functions. In Fig 1, the layout of the five main functions is shown. The visualization designs of these functions and their design rationales are described and explained in this section.

Bar Chart Timeline The bar chart timeline visualizes the ratios of different artwork type along with the timeline, to explore trends in art movements. On this timeline, we can click on it to look for more details of the periods we selected, zoom in and out to see it more clearly, and select the periods to get more information in the parts of the tree, word cloud, artist list, and image gallery.

This visualization was designed according to the visualization rationales. The timeline combined with a bar chart is designed here to provide an overview of

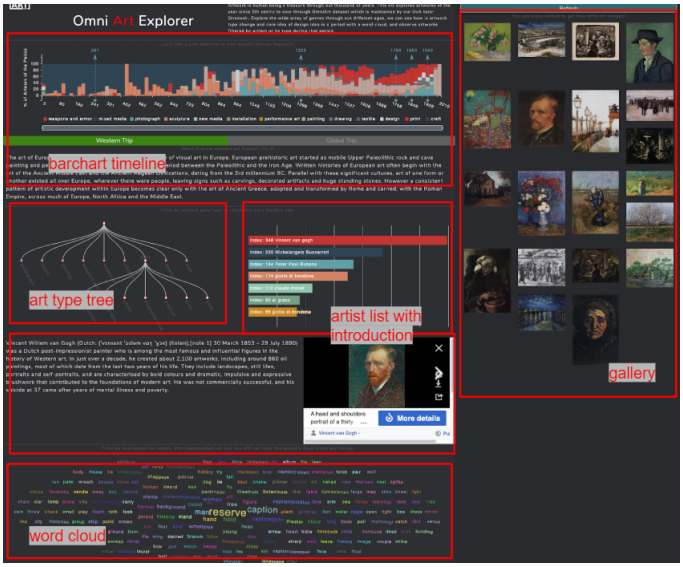

Fig. 1. Layout of five main functions in the system

the artwork types in different time periods in art history. The temporal data was visualized in linear time due to the uniqueness in art history. The interval time primitive is chosen to aggregate the data. Users could select the intervals they would like to explore. Also, the time reference points are added to the timeline. These time period marks give users a clear overview and an idea of the relative time period. Besides, the main-detail mode was designed to show the information flexibly in the form of displaying more details when hovering and clicking. Other interaction methods like filtering, reconfiguring and connecting are also designed and implemented in this part which would be discussed further in the section interaction design.

Artwork type tree The tree is designed to make our users get more information about the types and objects of the arts in the period they selected from the timeline. Users could click the nodes to unfold/enfold their child-nodes to get information about the sub-artwork-types.

The tree form is adopted in this function with the consideration of aggregating information in the visualization. Triangular vertical node-link layout clearly shows the hierarchy levels in the artwork types. The interaction approaches used here include abstracting/elaborating and connecting.

Artist List This ranking list shows the most famous artists in the user-selected period. The introduction of a specific artist is shown below by clicking on the artist name in the ranking list.

Word cloud Word cloud is a collection showing the most popular themes of artworks within users' interested areas by words depicted in different sizes, limited by the time period set in the bar chart timeline.

Since we would like to explore the theme of each artwork in history, this text visualization was designed. A form of word-cloud was chosen here to illustrate both the content and the significance of the themes at the same time. This illustration type is straightforward and easy to comprehend. We used wordle algorithm (Viegas et al. 2009) as a reference to implement our word cloud algorithm. The algorithm includes extracting keywords, filtering out stopwords and assigning weights according to word counts.

Gallery This gallery displays an image collection, comprised of many small-size images of the artworks. A large-size image would be shown once clicking on the thumbnail in the gallery. It gives users the potential details they would like to explore. The connecting interaction methodology was applied here. The gallery would be refreshed when changes are made in the other functional parts, including timeline, artwork type tree, artist list and word cloud. This enables users to focus on their specific interests.

3.2 Application implementation and structure

Application implementation The implementation processes can be divided into 4 parts: data cleaning, data scraping, data visualization, and data processing.

Data cleaning: As we are using OmniArt as our main dataset which contains massive information of artworks, the first thing to do is that extracting certain information from the dataset which directly contributes to our visual storytelling, and meanwhile, cleaning the data for further data analysis process. Since the format of metadata in the dataset is CSV, so we use Spark to read the original files, extract certain columns and process the extracted data into Apache Parquet format for the data processing in the next step. The reason why we choose Apache Parquet as our main data format is that it can significantly reduce the file size of our data.

Data scraping: The original dataset doesn't contain the detail information of artists, such as the artist's birthday, nationality, and portrait. In order to make our visual storytelling more well-rounded, we decide to use a web crawler to gather the information about artists on Wikipedia. In specific, we collect the artist's name, birthday, date of death, nationality, portrait, description, wiki URL, and reference number on the Wikipedia page, then we save this knowledge into a JSON file. Since the artists' full names in metadata are not always correct and some other reasons, we finally got 291 artists' profiles on Wikipedia.

Data visualization: To implement the frontend visualization components and user interaction, we choose Apache ECharts which is a declarative framework for web-based visualization development (Li et al. 2018). We used ECharts implementing the bar chart timeline, word cloud of topics, art type tree, and the artist list.

Data processing: To support data filter from HTTP requests, we developed a Data Processor, which can query the information from the metadata according to some specific conditions and return the information to users. Up to now, the Data Processor can support the artist filter, the artwork type filter, and the word cloud filter. Meanwhile, we also expose these filters as APIs for advanced users.

Application structure Our web application follows the MVC (Model-View-Controller) design pattern, which consists of three layers: the data-access layer (Model), the presentation layer (View), and the business logic layer (Controller) (Holovaty et al. 2009). We present the application architecture in Figure 2. In the data-access layer, we use Apache Spark to get the metadata from the OmniART dataset, meanwhile, using a web crawler to gather artist information from Wikipedia. Then we combine these data together and save them as our data files. The data processor is served as the main functional component in the business logic layer. In the presentation layer, we provide data visualization and user interactions through the web page and APIs.

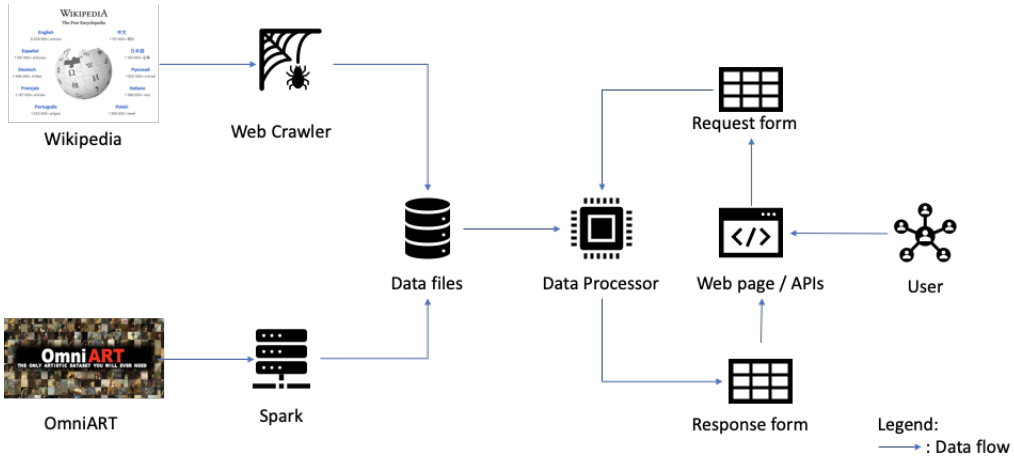

Fig. 2. Architecture of the OmniART explorer

4 Design Evaluation and Result

4.1 Interaction design

Various interaction approaches are used in this design, including exploring, reconfiguring, abstracting/elaborating, filtering and connecting.

Reconfiguring: In the bar chart timeline, users could select the time period. This operation would reconfigure the bar chart along with the timeline. Reconfiguring could be helpful for analyzing the hidden characteristics in the data. In our project, users could explore ratio patterns for different time periods.

Abstracting/elaborating: The abstracting/elaborating method is applied in the bar chart timeline. Users could also get detail information of each bar in the chart, which is a form of elaborating. This provides two levels of abstractions and a sententious visualization. Since different abstraction levels are suited for different stages ranging from overview to details, users could comprehend the information better and use them in different scenarios accordingly. In this Art Explorer application, users could have an overview of the ratios on the timeline, when it is needed they can dig into details about the exact artwork number for each artwork type.

Exploring: In the artwork type tree, the exploring approach is implemented with unfolding/enfolding the nodes. The artist list also enables users to explore the introduction of other artists than the default one. This method reveals a different subset of the data and solves the problem that users could only view a limited number of data at a time because of the cognitive and perceptual limitations, letting users of this system freely explore their interested artwork types and artists.

Filtering: In the timeline, users could set filters of each artwork type to focus on part of them to gain different ratio patterns.

Connecting: The gallery is connected with other function parts and it reacts to the changes. Different subsets of the data are visualized in different forms. This view is connected and synchronized together in the system.

4.2 Visual analytics design

The functions in our system were designed according to the basic visual analytics model (Cook et al. 2005). We combined the background model with the front-end visualizations. For example, for the artist list function, the input data of the selected period is generated from the user interaction part of the bar chart timeline. Then the model would run the algorithm to generate the seven most famous artists in this time period. Wikipedia was chosen here as a reference in this algorithm for popularity estimation. The ranking result would be displayed in the visualization. Users gain knowledge from both the model and the visualization.

Regards to the potential improvement of this design, parameter refinement of the model could be considered in future work. For instance, for the artist list function, users' interest of specific artists could be gathered in this system as

well by analyzing the introduction page view count of those artists. In this way, the model could be refined better.

4.3 Storytelling design

Different visualization techniques are combined and connected together in this system as a whole. Users can get an overview of artworks from various perspectives. One of our main story lines is the artwork timeline. Samples in the artwork gallery, top artist list, and word cloud of topics all change along with the timeline. Meanwhile, we also add several marks on the timeline to emphasis some special period in history, like the medieval art, the Renaissance, and the Romanticism. Users can click these marks and the timeline will be expanded for this period. Then the user can explore the artworks from this period in the artwork gallery and gain an insight into various artwork styles. On the other hand, artworks include various genres, such as painting, sculpture, and photographs. We also implement an artwork type filter so that users can use it to focus on the specific artwork type that they are interested in.

4.4 Visual thinking design

In our visual design, we applied pattern perception laws and some techniques to highlight the key information. The limitation of people's perceptual capabilities is also considered with visual queries. In the main functions, the Gestalt laws (Nesbitt et al. 2002) for pattern perception are applied. For example, the tree design is a combination of symmetry and continuity, users could spontaneously follow the link and collect information of each node on this link. In the word cloud, the law of relative size is applied to display dimensions of the significances of the themes. Some elements like color, size, and closure are adopted as well to highlight the key points for users.

Apart from the Gestalt laws, the whole application is designed in consideration of visual queries. Users could set filters of the time period, artwork type, theme in the relevant function parts. The gallery is fixed on the right side of the page to show the synchronized related images. This design allows users to trace through multi-media information, including word description in the word cloud, chart data in the timeline, image sources in the gallery, etc while focusing on their own goals.

When it comes to the potential improvement of this visual design, more clear instructions, and section arrangement could be expected. The selection bar on the top could be extended with more layers, and also be adapted to the users' behaviors, following the way of thinking. The different function sections could be rearranged to display more clearly.

4.5 Visualization knowledge

In this project, the result generated from the application were analysed to gain knowledge from the OmniArt dataset.

In the bar chart timeline, we could find that craft takes a dominant place in all the artwork types in the ancient period and medieval period. The sculpture is also another important artwork type. The ratio of those two types decreases while the ratio of painting increases. In the Renaissance period, painting is the most prevailing artwork type. Although in the late renaissance, the ratio of craft raised, about the ratio of painting. In the modern and contemporary period, more and more artwork type arose, including design, new media, installation and performance art, etc. As we can see, painting is a significant art form in the whole art history. Craft and sculpture are getting relatively less and less nowadays, while more and more new artwork types fertilize contemporary art.

The artwork type demonstrates a clear overview of different artwork types in the dataset to provide users with the background knowledge of artwork categorization.

In the generated word cloud, the result shows that some themes are more frequent than others, e.g., "reserve", "woman", "tree", etc, implying that portrait and landscape are some of the common themes. Due to the lack of data in the OmniArt dataset, the result still needs to be improved in further work.

The artist ranking list indicates that the top three famous artists are Thomas Rowlandson, Vincent Willem van Gogh, and Michelangelo di Lodovico Buonarroti Simoni.

References

1. Cook, K. A., & Thomas, J. J. (2005). Illuminating the path: The research and development agenda for visual analytics (No. PNNL-SA-45230). Pacific Northwest National Lab.(PNNL), Richland, WA (United States).
2. Holovaty, A., & Kaplan-Moss, J. (2009). The definitive guide to Django: Web development done right. Apress.
3. Li, D., Mei, H., Shen, Y., Su, S., Zhang, W., Wang, J., ... & Chen, W. (2018). ECharts: A declarative framework for rapid construction of web-based visualization. Visual Informatics, 2(2), 136-146.
4. Nesbitt, K. V., & Friedrich, C. (2002, July). Applying gestalt principles to animated visualizations of network data. In Proceedings Sixth International Conference on Information Visualisation (pp. 737-743). IEEE.
5. Strezoski, G., & Worring, M. (2017). Omniart: multi-task deep learning for artistic data analysis. arXiv preprint arXiv:1708.00684.
6. Viegas, F. B., Wattenberg, M., & Feinberg, J. (2009). Participatory visualization with wordle. IEEE transactions on visualization and computer graphics, 15(6), 1137-1144.