MoVis - Movie Recommendation and Viasualization

Ye Chen, Yujie Yang

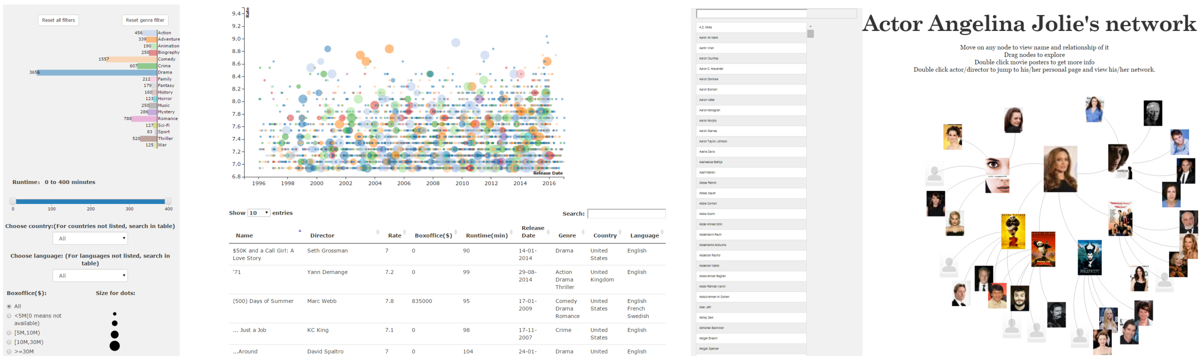

Fig. 1. Overview of two pages on MoVis website. Left: Movie Scatterplot page; right: Actor/Director Network page.

Abstract—This paper describes MoVis, an interactive visualization system for movie recommendation. The high level goal is to support 1) end users to select movies based on their preferences for movie attributes and 2) end users to explore the cooperation network of target actor/actress/director and his/her products. Based on these two tasks, two visualization pages are included in MoVis: Movie Scatterplot page and Actor/Director Network page. Two corresponding scenarios are also proposed for the two tasks. Further comments and advices are raised for future work at the end.

Index Terms—Interactive visualization, movie recommendation, scatterplot, network

1 Introduction

Nowadays, watching movies becomes a popular way for people to relax after one day's intense work, or spend spare time together with friends. A good movie recommendation system should help users find a satisfying movie without much exploration and searching.

Most existing movie recommendation websites like Criticker [2] make recommendations by generating a list of movies according to users' preferences, where users have to scroll up and down and flip over different pages to look over the movies without an overview of all the movies.

As far as we observed, no live website at present uses visualization tools to show recommended movie data or show an overview of network graph of actors/directors and their products. By implementing Movie Scatterplot and Actor/Director Network pages (Figure 1), MoVis helps users explore the movie data and actor/director network more efficiently.

Our project aims to support two groups of users: 1) end users who want to find a good movie but having no specific target on mind, and 2) end users who are interested in some specific actor/director and would like to explore his/her products or other actors/directors who have cooperated with him/her.

In the next section, we present previous work in the area including the current visualization tools for media libraries and existing visualization tools for social network. Section 3 includes the descriptions of the data and abstraction of two tasks, following by the solutions for these two tasks in Section 4. Section 5 includes the implementation of MoVis. Section 6 includes two scenarios for two end user groups. In section 7, some discussions and future works are included. In the last section, we conclude our contribution and lessons learned from this project.

* *Live demo of MoVis:* `http://www.cs.ubc.ca/~rain0422/MoVis/`

* *Ye Chen is a master student in Management Information Systems in Sauder Business school at The University of British Columbia. E-mail: clara.yechen@gmail.com.*
* *Yujie Yang is doing Comprehensive Course Master in Computer Science Department at The University of British Columbia. E-mail: yangyujie.hust@gmail.com.*

2 Previous Work

2.1 Visualization Tools for Media Libraries

Several visualization tools have been designed for browsing, exploration, and searching through large collections of movies, music, photos, books and personal listening histories [7, 5, 20, 13, 14, 19, 8]. Among those visualization tools, some of them implement a scatterplot-based main view [7, 5, 13, 19]. FilmFinder [7] features tightly coupled interactive filtering, where the result of moving sliders and pressing buttons is immediately reflected in the visual encoding. Musicovery [5] is an online music streaming tool that visualizes its music collection based on mood and emotional content of music tracks. The interface makes good use of space and color coding in various filter types including genre, date, artist, intended activity,and etc. PhotoFinder [13] is designed to be easy to use for searching and browsing photos by using a scatter plot thumbnail display and drag-and-drop interface. Bohemian bookshelf [19] is a visualization interface for browsing a book collection with the goal of supporting serendipity. The interface includes five interconnected views that encode physical and/or semantic properties of the books in various ways. The interface is designed to support playful interaction and provide various access points to explore the collection. LastHistory [8] presented an overview of listening histories as a personal type of data and additional contextual information in the form of photos and calendars.

However, other tools proposed some other features: treemap, disc or rectangle visualization as their main view [20, 14]. Torrens [20] proposed three visualizations for personal music collections: disc, rectangle and treemap and he suggested to use treemap visualization. Pocket PhotoMesa [14] is an image browser for the pocket pc that employs quantum strip treemaps for laying out images and zoomable user interfaces for navigation.

Apart from the usual visualization approach, some media visualization systems use machine learning algorithms such as clustering to group and present media items based on their similarity [15, 18]. Because of limit of time and goal of the MoVis, we dont focus on using machine learning algorithms on this stage.

Our Movie Scatterplot page uses interactive scatterplot for exploration and shares characteristics with some of the above examples. Those examples provide us good support of using scatterplot-based view for visualization of media libraries.

2.2 Social network Visualization

Visualization of social networks has a rich history, particularly within the social sciences, where node-link depictions of social relations have been employed as an analytical tool since at least the 1930s. Linton Freeman documents the history of social network visualization within sociological research, providing examples of the ways in which spatial position, color, size, and shape can all be used to encode information [11].

In recent years, such approaches have been effectively used in the analysis of domains such as e-mail communication [10], early online social networks [9], and co-authorship networks in scientific publications [17]. There are a number of systems for generating such social network visualizations and performing statistical analyses for the purpose of sociological research, such as JUNG [4].

In addition to sociological research, there have been numerous social visualization projects intended for end-users rather than outside researchers. Vizster[12] is a visualization tool for playful end-user exploration and navigation of large-scale online social networks. It builds upon familiar node-link network layouts to contribute customized techniques for exploring connectivity in large graph structures, supporting visual search and analysis, and automatically identifying and visualizing community structures. Co-author graph in Microsoft academic search [1] shows us the top 30 co-authors of the main researcher in the middle. It allows users to drag the nodes to explore the connection and change the main researcher in the middle. TouchGraph [6] uses a force-directed layout to present a network visualization of users of the LiveJournal online community, allowing personal networks to be expanded or contracted by user interaction.

The Actor/Director Network page shares lots of features with the above visualization tools, especially Vizster [12] and the co-author graph [1].

3 Tasks and Data Abstraction

We will describe our two tasks and do a data abstraction following idioms used in Visualization Analysis and Design [16].

3.1 Tasks

We have two tasks for two kinds of end users:

1) Finding a target movie depending on users' preference of attributes, such as rating, boxoffice, genre or release date of movies. Current movie recommendation websites, like Criticker [2], usually produce a list of movies depending on users' preference. However, users have to scroll up and down to find a target movie on the list without an overview of all the selected movies. Therefore our aim is to produce an overview of the movie data according to the users' preference, as well as looking into the details of each movie.

2) Exploring the movies casted by a specific actor/director and the other relevant actors/directors. When the user is a fan of a actor/director, he/she would be interested to see what movies were casted/directed by the actor/director and also the network of cooperation relationship. Our aim is to provide an overview of related movies and a network graph of a specific actor/director.

3.2 Data

Our data is from IMDB PRO, where they provide table data and the attributes of each movie include name, runtime, release data, genre, boxoffice, country, language, rating, summary, list of directors and list of movie actors, as shown in figure 2. The data we have scrawled is composed of the top movies from 1995 to 2016. We have scrawled 33339 movie items from IMDB PRO. Given the purpose of MoVis is out of movie recommendation, we selected 4997 movie items whose rating are greater than or equal to 7 as our data.

Table 1. Attributes and corresponding channels encoded in scatterplot

Attribute	Channel
Release Date	Spatial region(x axis)
Rating	Spatial region(y axis)
Boxoffice class	Size
Genre	Color hue

Each movie item in the first visualization page has the following attributes:

- Release Date (Ordinal) - 1996 to 2016 year
- Rating (Quantitative) - 7 to 10
- Runtime (Quantitative) - 0-400 minutes
- Country (Categorical) - hundreds of countries
- Language (Categorial) - hundreds of languages
- Boxoffice (Quantitative) - 0-990MM USD
- Genre - 18 categories

The categories of language and country are more than one hundred. Since some of them just have few movies, we just chose several of them with higher distributions and the choices of languages and countries are reduced to 7 and 5 separately. Given the distribution of boxoffice for movies are not average, four classes of boxoffice are created to represent boxoffice intervals of [0, 5M), [5M, 10M), [10M,30M), [30M,+∞), to ensure that numbers of movies in each interval are close.

Data for second visualization page is derived from the original table data. Each actor/director data in the second visualization page has the following attributes:

- Movie he/she plays in(Categorical)
- Actor/Director(Categorical)

IMDB contains the full lists of actor information for each movie, which makes the numbers of actors for all movies very large. However, there is no need to include all those actors since usually the first several actors contribute most and attract most attention. Therefore, we only selected the first 5 actors who are the leading roles in that movie to generate our actor list. All together, the full data set in the second visualization page contains 14384 actors and 3797 directors. For each actor/director, name, link to his/her profile image and the name of movie he/she plays are used in the second visualization.

4 Solution: MoVis Description

For our two tasks, we designed two website pages respectively: Movie Scatterplot page and Actor/Director Network page. In this section, we will analyze the design choices and compare some alternatives.

4.1 Movie Scatterplot Page

To give users a general taste of all the movies, we designed Movie Scatterplot page, including a scatterplot as the main view on the right top, some filters on the left and a movie detail table below the scatterplot (see figure 1 left).

Fig. 2. When brush rectangle focusing on the three dots in the scatterplot, further movie information of those dots are exhibited in the table below

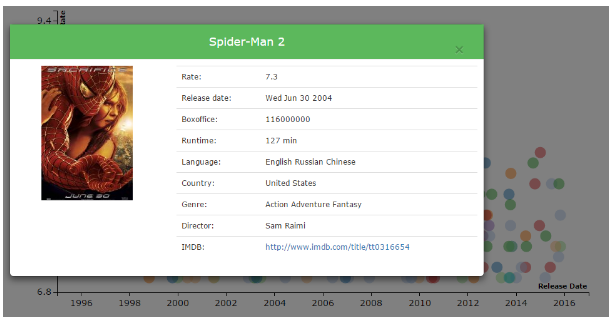

Fig. 3. When clicking on one dot in the scatterplot, a window with information of the corresponding movie pops up

4.1.1 Scatterplot

In the scatterplot, each movie item is encoded into a dot. Table 1 demonstrates attributes and the corresponding channels encoded in scatterplot.

For manipulation strategies, we considered two alternatives: zooming and constrained (brushing). At first, we implemented both strategies but we found immediately that they will affect each others. We then compared their effectiveness in our scenarios and decided to keep constrained. As for zooming, since there is always a constraint on zooming extent, it is impossible to select specific dot collections when arriving the bound. However, constrained gives us a chance to select a dot collection and meanwhile have a overview of the whole data set. Therefore, we implemented constrained strategies as brushing on scatterplot. Users are allowed to select dots in a grey rectangle and change the position and size of the rectangle. All the dots selected by brush rectangle will be highlighted and the corresponding movies will be listed in the table below (see figure 2).

We embedded details of a movie into a pop-up window. When clicking on one dot in the scatterplot, a window with movie details will pop up (see figure 3). There is an IMDB link in the pop-up window, leading users directly to IMDB page of the selected movie. This action contributes the last step and finishes the whole movie recommendation process. Users could then explore more details of the their interested movies on IMDB website.

4.1.2 Filters

Presenting thousands of dots in a scatterplot works terribly to find interested movies. To reduce the items showing in the scatterplot, we implemented filters on the left of the page. There are five filters in

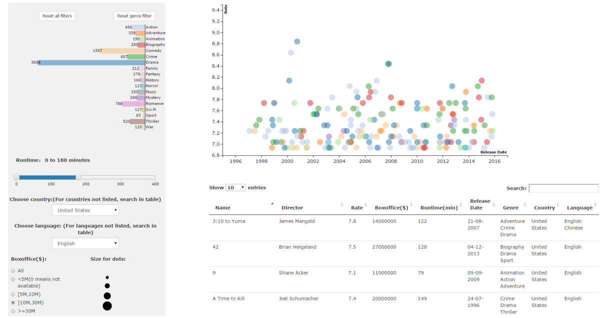

Fig. 4. Filtering results show up in the scatterplot with runtime ∈ [0, 180], country = United States, language = English and boxoffice ∈ [10M, 30M).

Fig. 5. Interactions with horizontal histogram of genre. Left: when hovering over genre drama, bar for drama in the histogram and dots for drama movies in the scatterplot are highlighted, which is demonstration of linked highlighting; right: when clicking on genre drama, dots for drama movies will be kept and other dots will be filtered out.

total: genre, runtime, country, language and boxoffice class (see figure 4). All the filtering results will immediately be reflected in the scatterplot and the detail table.

The genre filter includes a horizontal histogram, which reflects the genre distribution of all the data. The color of each bar corresponds to the color of the movie data in the scatterplot. When hovering over the bar, the bar and dots of this genre will be highlighted, which is a demonstration of linked highlighting (see figure 5). The runtime filter is a slider with range [0, 400]. Country and language are encoded in drop-down menu and boxoffice class in radiobox.

4.1.3 Detail table

To provide more support to movie details, we implemented overview-detail strategy. Below the scatterplot is a table with all the details of movies. It will list the movies filtered by filters or only the selected movies by brushing function if users are using brushing function(see figure 2). In addition, this table features several useful interactions (see figure 6). When clicking any row, the corresponding movie detail will be shown in the same pop-up window as clicking any dot in the scatterplot (see figure 3). It has searching function, where results are immediately shown when keying in something to search all the related movie datas in the table. It also allows users to rank the items by clicking the head of any columns.

Table 2 summarizes the above analysis following the idioms used in Visualization Analysis and Design [16].

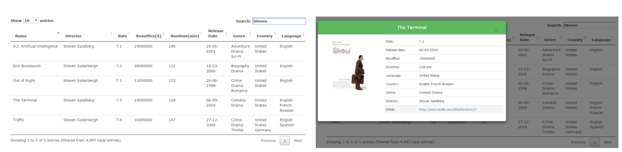

Fig. 6. Interactions with detail table. Left: when keying in "Steven" in the search bar above the table, movie items with substring "Steven" will be filtered and presented in the table; right: when clicking the fourth row, a window with information of movie "The terminal" and links to its IMDB page pops up.

Table 2. What-Why-How analysis of Movie Scatterplot page

System	Movie Scatterplot page
What: Data	Table: four ordered attributes(runtime, boxoffice, rate, release date), three categorical attributes(genre, country, language)
What: Derived	Derived table: one ordered attribute(boxoffice class), one quantitative value attribute(item count per genre)
Why: Tasks	Find items by users' preference of attributes
How: Encode	Scatterplot; horizontal histogram; table
How: Facet	Multiform; overview-detail; linked highlighting
How: Reduce	Item filtering
How: Embed	Pop-up window with details
How: Manipulate	Constrained(brush)
Scale	Items: thousands

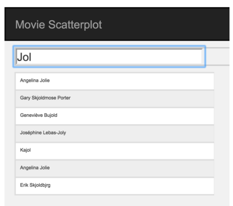

Fig. 7. Search Angelina Jolie by typing "Jol" in the search bar and the results reflect in the list immediately.

4.2 Actor/Director Network Page

Actor/Director Network Page has two parts: name list on left and network on right (see figure 1 right).

4.2.1 Name list

We implemented a name list of all actors/directors on left of the page. Searching function is supported (see figure 7). Users could either search or scan the list to find an interested actor/director. Clicking any of the name, the network of the selected person will be shown on the right.

4.2.2 Network

Relationships between actors/directors and their products can be very complex. To present the relationship intuitively, we generated a network dataset from our original table dataset and encoded the relationship of actors/directors and their products as a node-link diagram (see figure 8). Each movie item or actor/director is encoded as a node. Each node is encoded as movie poster or actor/director photo. The spatial region and size of the nodes imply their roles: the core actor/director is fixed in the middle and has a larger size than all other nodes, products are the next round of nodes with links to the core node and the outer round of nodes are the actors/directors who contribute much in the movies. Here, IMDB lists actors by significance of contribution, so we choose to show the first five actors of each movie. Actors/directors and their products are linked by curves. Table 3 summarizes the attributes and channels encoded in network.

The network features self-balanced structure and always organizes a balanced layout of all the nodes and links. Each time the network is reloaded, the core node will show up firstly in the middle, and then other relevant nodes and links are loaded and organized until reaching balance. Name of movie and actor/director are embedded as texts under each node and when hovering over any node, the relevant texts will show up and the links will be highlighted (see figure 8).

Until now we have achieved the goal of presenting the most relevant actors/directors and products of the core actor/director. However,

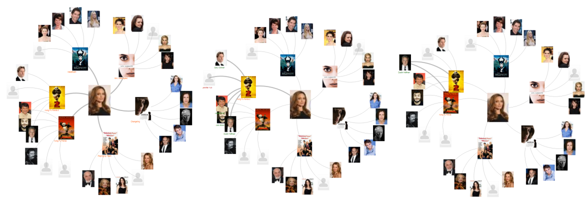

Fig. 8. Interactions with network of Angelina Jolie. Left: when hovering over Jolie's photo, links between she and her products will be bold and names of those products will be shown under the movie posters; middle: when hovering over one movie poster, links between this movie and director and actors casting in it will be bold and the names of them will show up under the photos; right: when hovering over one actor/director photo, links between he/she and the products will be bold and names of those products will be shown under the movie posters, and numbers of products imply the cooperation times of this actor/director and Jolie.

Table 3. Attributes and channels encoded in network

Attribute	Channel
Movie item	Node
Actor	Node
Director	Node
Node	Spatial region; size
Link	Curve

we need to provide further information of those actors/directors and products. Since this single network is not enough to do that, we added some interaction and manipulation. For manipulation strategies, we allowed users to drag the nodes to explore the relevant relationship. Moreover, users could double click on a movie node to get more details in a pop-up window (see figure 9) or on an actor/director node to jump to his/her network page. Those manipulation strategies are the last few steps of our recommendation process before users find their interested movies.

Table 4 summarizes the above analysis following the idioms used in Visualization Analysis and Design [16].

5 Implementation

We built the MoVis website using HTML, D3.js and CSS, and we also used python and Google Refine to extract and cleanse data from IMDB PRO.

5.1 Data scrawling and cleansing

The data was scrawled from IMDB PRO using python, which was stored as rate7.csv. Google Refine was used to clean the data, including transformation of boxoffice from text (like MM,B) to numbers, clustering of same categorical data together, cleansing of the string data by extracting meaningless signs.

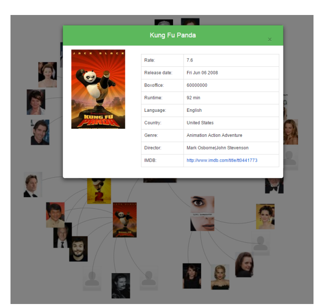

Fig. 9. When double clicking on a movie node, a window with information of the corresponding movie pops up.

Table 4. What-Why-How analysis of Actor/Director Network page

System	Actor/Director Network page
What: Data	Table: three attributes (movie item, actor, director)
What: Derived	Networks: movies/actors/directors as nodes; relationship of actors/directors and movies as links
Why: Tasks	Explore relevant nodes
How: Encode	Node-Link diagram; list
How: Reduce	Only show the first five actors of a movie
How: Embed	Pop-up window with details; highlight relevant nodes and links when hovering over any node
How: Manipulate	Drag; double click actors/directors to jump to his/her network; double click movies to show details in pop-up window
Scale	Nodes: hundreds; links: hundreds

Table 5. Split of work

Work description	Ye	Yujie
Design	50%	50%
Data scrawling and cleansing	100%	0%
HTML for both pages	100%	0%
Scatterplot and brush function	50%	50%
Pop-up window	100%	0%
Filter function and reset function on first page	0%	100%
Detail table and relavant interaction on first page	0%	100%
Search function on second page	100%	0%
Network and relavant interaction	0%	100%
Writing(slides + report)	50%	50%

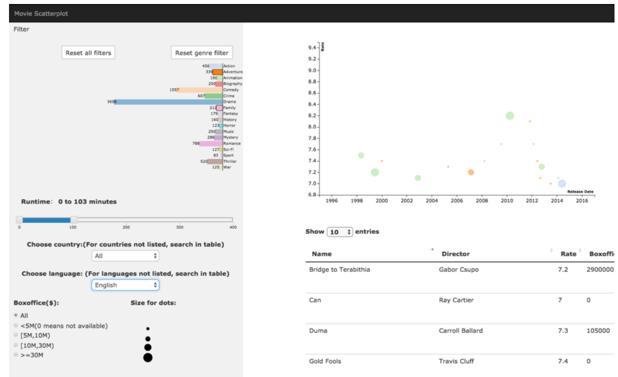

Fig. 10. Step 1 of scenario for Movie Scatterplot page: filtering results show up in the scatterplot with runtime ∈ [0,103], country = All, language = English.

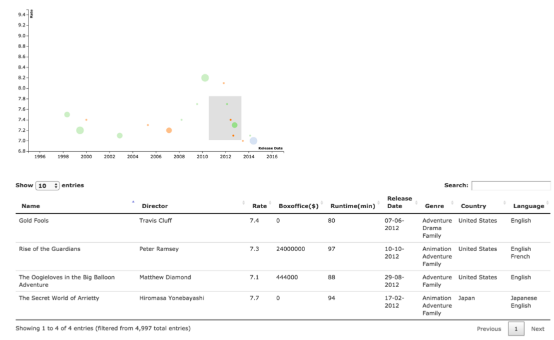

Fig. 11. Step 2 of scenario for Movie Scatterplot page: select the four movies more recently with higher boxoffice by brushing to see the details of the selected movies in the table below.

5.2 Movie Scatterplot page

The MainPage.html describes the layout of the framework. The layout of Movie Scatterplot is in Scatterplot.html, including the layout of drop-down menu and radiobox. Scatterplot.js has all the JS codes used in this page. For all the filtering and reset functions, we used javascript for algorithms. We used D3 library for the scatterplot, brush function and horizontal histogram. The detail table below the scatterplot is built based on DataTables, a table plug-in for jQuery [3]. Runtime slider is built based on a plug-in for jQuery created by Bjorn Sandvik.

5.3 Actor/director Network page

The layout of this page is in ActorNetwork.html, including the table of all the actors/directors names. ActorNetwork.js has all the JS codes used in this page. We used force layout in D3 library for the node-link diagram. All the manipulation and interaction functions are built in javascript. We also used existing JS Library QuickSearch to realize the live searching function with comparably quicker speed than the ajax function we wrote ourselves.

Table 5 shows the split of work. We both contributed much to this project and enjoyed our teamwork a lot!

6 Result

We provide usage scenarios for both end users.

6.1 Scenario for Movie Scatterplot page

Stephanie is a computer science student at UBC. In one Saturday night, she finished all the projects just before deadline and wanted to watch a movie at home. She opened the MoVis website and started to explore the system. She would like to see adventure movies which are also related with family, so she clicked both Adventure and family together on the histogram filter. All the movie dots with both genres Adventure and Family were kept in the scatterplot after filtering. She also limited the runtime range to 0-103 minutes since she did not want to spend too much time on the movie. She also selected English as the movie language (see figure 10). According to the scatterplot after filtering, she selected the four movies more recently with higher boxoffice by brushing to see the details of the selected movies in the table below (see figure 11). Reviewing the details of those four movies, Stephanie decided to watch Rise of Guardians for the night (see figure 12).

6.2 Scenario for Actor/Director Network page

Felix is a big fan of Angelina Jolie. One day, he wanted to watch a movie after working. He went to the Actor/Director Network page and searched for Jolie in the left search bar (see figure 7). He clicked Jolie's name and the network of Jolie will be correspondingly shown on the right side. He found movies in the graph not very interesting and decided to watch a movie played by other actors/actress in the graph. By hovering over the network graph, he found Dustin Hoffman cooperated twice with Jolie. Then he double-clicked the photo of Dustin Hoffman to open his personal network graph (see figure 13). By double-clicking the movie nodes for pop-up windows with movie details, he found an interesting movie Being John Malkovich with fancy poster. Finally, he chose this one for his night and enjoyed it a lot.

7 Discussion and Future Work

Overall, we believe MoVis is an useful first prototype of a movie recommendation system using visualization methods for the convenience

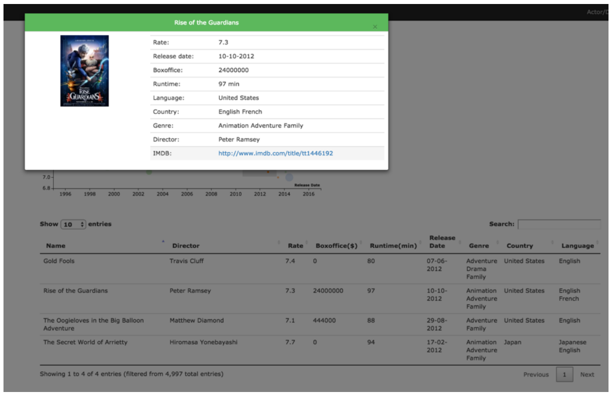

Fig. 12. Step 3 of scenario for Movie Scatterplot page: reviewing the details of those four movies, Stephanie decided to watch Rise of Guardians for the night.

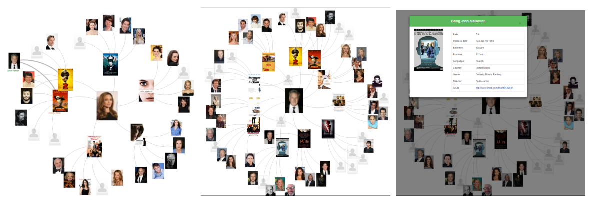

Fig. 13. Scenario for Actor/Director Network page. Left: Jolie's network; middle: double-click the photo of Dustin Hoffman and jump to his network; right: Felix decided to watch this movie due to its fancy poster.

of users. Further need for evaluations of MoVis is needed for system improvement.

During the implementation process for the first page, we tried to implement zooming function. But it arouses the conflict with brushing function. For future reference, zooming function as data manipulation could be further explored. And how to deal with genre data should be further considered, that is which genre tag should be used to represent one movie with multiple genre tags. Further, the loading speed for filtering movie data is slow for larger data set, therefore how to design the algorithm to improve the loading speed should be further investigated. For the second page in MoVis, two problems should be further studied. First, the searching function efficiency should be improved. Even though we used the existing JS library QuickSearch, it still takes some time to load given the data size. Secondly, some actor/director nodes has no image link. So we need more images in network graph to refine our website.

8 Conclusion

This paper introduces MoVis, including two interactive visualization system for movie selection. We used 1) Scatterplot-based multiform views with filtering functions to allow users pick a satisfying movie, 2) Node-link diagrams to allow users explore the cooperation network of actors/directors and their products.

In the designing process of MoVis, we were exposed to several design choices for movie data representation. We realized a good designer should consider multiple factors when considering a visualization method. In the implementation process, we learned basic web implementation skills from the beginning, especially come to understand the power of D3 and other visualization tools.

Acknowledgments

We wish to thank Tamara Munzner and Zipeng Liu. From Tamara's course, we obtain insight of information visualization and practice visualization tools in this project. Zipeng kindly helped us to fix several bugs and scrawl some image links from IMDB.

References

[1] Co-author graph. `http://academic.research.microsoft.com/VisualExplorer`.
[2] Criticker. `http://www.criticker.com`.
[3] Datatable. `https://www.datatables.net/`.
[4] JUNG: Java universal network/graph framework. `http://jung.sf.net/`.
[5] Musicovery. `http://musicovery.com`.
[6] Touchgraph. `http://touchgraph.com`.
[7] C. Ahlberg and B. Shneiderman. Visual information seeking using the filmfinder. In *Conference companion on Human factors in computing systems*, pages 433–434. ACM, 1994.
[8] D. Baur, F. Seiffert, M. Sedlmair, and S. Boring. The streams of our lives: Visualizing listening histories in context. *Visualization and Computer Graphics, IEEE Transactions on*, 16(6):1119–1128, 2010.
[9] O. Buyukkokten, E. Adar, and L. Adamic. A social network caught in the web. *First Monday*, 2005.
[10] D. Fisher and P. Dourish. Social and temporal structures in everyday collaboration. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 551–558. ACM, 2004.
[11] L. C. Freeman. Visualizing social networks. *Journal of social structure*, 1(1):4, 2000.
[12] J. Heer and D. Boyd. Vizster: Visualizing online social networks. In *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pages 32–39. IEEE, 2005.
[13] H. Kang and B. Shneiderman. Visualization methods for personal photo collections: Browsing and searching in the photofinder. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 3, pages 1539–1542. IEEE, 2000.
[14] A. Khella and B. B. Bederson. Pocket photomesa: a zoomable image browser for pdas. In *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*, pages 19–24. ACM, 2004.
[15] B. Moghaddam, Q. Tian, N. Lesh, C. Shen, and T. S. Huang. Visualization and user-modeling for browsing personal photo libraries. *International Journal of Computer Vision*, 56(1-2):109–130, 2004.
[16] T. Munzner. *Visualization Analysis and Design*. CRC Press, 2014.
[17] M. E. Newman. Coauthorship networks and patterns of scientific collaboration. *Proceedings of the national academy of sciences*, 101(suppl 1):5200–5205, 2004.
[18] G. P. Nguyen and M. Worring. Interactive access to large image collections using similarity-based visualization. *Journal of Visual Languages & Computing*, 19(2):203–224, 2008.
[19] A. Thudt, U. Hinrichs, and S. Carpendale. The bohemian bookshelf: supporting serendipitous book discoveries through information visualization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1461–1470. ACM, 2012.
[20] M. Torrens, P. Hertzog, and J. L. Arcos. Visualizing and exploring personal music libraries. In *ISMIR*, 2004.