INTEREST-GUIDED EXPLORATION OF LARGE INFORMATION SPACE

by

Yueqi Hu

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of PhD. in
Computer Science

Charlotte

2018

Approved by:

Dr. Jing Yang

Dr. Aidong Lu

Dr. Kalpathi Subramanian

Dr. Wei Fan

ABSTRACT

YUEQI HU. Interest-guided Exploration of Large Information Space. (Under the direction of DR. JING YANG)

With the coming of the digital era, information has never been as massive and complicated as it is nowadays. New layout strategies, visual encoding methods, and exploration mantras have been developed in the visualization area to "squeeze a billion records into a million pixels". However, the result usually loses the intuitiveness and appealingness that make visualization effective in the first place. This dissertation presents an alternative approach to address the conflict between the volume of data and the scalability bottleneck of visualization. Instead of showing everything at once, it moves analysts close to their datasets to see the task-relevant part only. In this way, only a handful of information is rendered on screen at any time, and it becomes practicable to rely on visual presentations intuitive and easy to learn. The challenge is how to navigate users to find their desired information island in a massive sea, especially when the explorer may not have a crystal clear task. It is also expected that an exploration system should encourage users to discover unexpected yet useful information and build a thorough understanding of the datasets as a result. My mantra has three steps: land, modify and shift. First, a user lands in an information space via a user-given center. The system automatically extracts and visually presents information around it. Second, the user modifies the visualization to match his/her exploration intent better. Third, the user shifts the center as his/her exploration intent changes during the analysis. Flexible and intuitive visual interfaces and interactions are relied on to raise awareness and navigate exploration throughout the process. This dissertation presents the mantra and illustrates it through three distinct projects. For each project, the challenges, design concerns, implementation details, and evaluations are reported. They reveal the shared features, benefits, and

limitations of the mantra. At the end of the dissertation, I will summarize a design guideline, features, benefits, and limitations of the mantra.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 A New Visual Exploration Mantra

Visualization refers to the creation of visual representations of data in pictorial or graphical formats to help users perceive and understand information in a memorable manner. For centuries, visualization has been a powerful tool for scientists, engineers, and business people to analyze records, extract insights, and convey discoveries to the public. With the coming of the digital era, never before in history has the information at analysts' hands been generated at such high volumes. Many visualization techniques of the old days lack the scalability and flexibility desired nowadays. New visual exploration techniques are highly demanded to help uninformed dataset owners to gain insights, raise hypotheses, and formulate analysis goals.

One popular visual exploration mantra is "overview, zoom, details on demand"[1]. A visual analytics system first gives a user an overview of a dataset. Then the user browses each portion of the data interactively based on how interesting it looks from the overview. Techniques such as clustering, sampling, and dimension reduction determine the quality of the overview. However, picking suitable parameters for these algorithms is usually hard both automatically and manually. Thus, using this approach carelessly will bring in information lost and misleading insights beyond a user's awareness.

Despite these problems, the mantra is still impactful because it matches two myths of visualization design. First, a visualization system needs to load all data in the display at once, because "one picture worth a thousand words." However, overwhelming information would lessen efficiency and discourage users. In addition, from an analyst's point of view, very few tasks request all information at once. Instead, an

analyst may want to move closer to a dataset and focus on task-relevant information. Second, a successful visualization is always introduced as a perfect visual representation of a dataset, which manifests right decisions or actions. However, visualization cannot and should not replace critical thinking. On the contrary, visual thinking plays an important role in revealing unanticipated insights, triggering creative thinking, and inspiring new analysis ideas. Especially, in exploration where people start with an open mind, what people see will change what they want. In this sense, a fixed presentation of a dataset, such as an overview, prevents a user from a broader investigation.

Debunking the two myths above will extend the visualization design space and may address the scalability problem with big data. With this in mind, in this dissertation, I propose a different visual exploration mantra. I also show its benefits through three remotely different implementations.

1.1.1 The Proposed Mantra: "Land, Modify, and Shift"

My idea is simple: an exploration is a chain of clearly-defined close-up mini visual analysis task. Each mini task inspires a user about the next task by providing more information. The user extends his/her horizon bit by bit, and eventually, the user travels through the dataset.

From a user's point of view, the exploration process can be concluded with three steps: "land, modify, and shift". The exploration starts as the user "lands" on an "ego". The ego is an analysis target that either reflects the user's initial analysis interest, or lets the user start the exploration confidently with familiar knowledge. The system shows the information relevant to the ego with a visualization. Then the user can modify the visualization to support his/her analysis task about the analysis target, or the ego. The user can also continue the exploration by shifting the ego, as he/she finds a better "landing spot" for the desired information, or changes his/her analysis intent after learning new information.

The concept of ego is introduced for two purposes. First, it roughly captures the analysis intent even though the user can analyze its relevant information in different ways. With an ego, every link in the exploration has a semantically meaningful topic. Second, it sets the center of the extracted information, so that user can control information size in visualization by adjusting the boundary. It intuitively avoids information overwhelming.

1.1.2 Advantages of the "Land, Modify, and Shift" Mantra

The "land, modify, and shift" mantra has the following benefits compared to the "overview first" approach:

- It has a simple start. Picking an ego is easy for analysts who have a little experience in similar analysis tasks and a little knowledge about a dataset. Moreover, starting an exploration with a clearly-defined mini task builds confidence, while understanding an abstracted overview may discourage new users.

- It has good scalability. No matter how big a dataset is, a user only sees one small information portion defined by an ego and visualization settings. Thus, details would not be overwhelming as they were in an overview. The small volume of data in a display also makes it practical to build the visualization with basic charts, such as line charts, scatter plotsm and tables. It not only reduces the learning curve, but also enables integrating the display with other visualizations.

- It encourages serendipitous discoveries. When a user modifies the visualization, different perspectives of the extracted information are shown, which is also a learning process for the user. Thus, each modification moves the exploration to a different direction. The potential of encountering vast information in the exploration enhances user experience, encourages interactions, and leads to serendipitous discoveries.

1.1.3 Design Challenges of the "Land, Modify, and Shift" Mantra

To optimize the overall efficiency of an exploration with the "land, modify, and shift" mantra, the following challenges need to be addressed:

- Showing diverse information: Users change egos during an exploration. New egos are usually discovered from visible information portions. Therefore, a user may be trapped in an information bubble if the new egos cannot bring in diverse information portions. To address this problem, a system needs to deliberately increase the diversity of the encountered information.

- Building user confidence: Exotic design confuses users and discourages them from moving forward in a large information space. Therefore, a system should be intuitive and self-explained, so that users always know what they are doing. Conducting analysis with basic charts, such as bar charts or line charts minimizes the learning curve and ensures user confidence.

- Allowing users to move fast: Users should not spend much time on a poor analysis target. While it is relatively hard for users to predict the worthiness of an ego without knowing the task, they should be able to evaluate the worthiness quickly after seeing the visualization and then make decisions. It requires the visualization to be efficient and informative.

- Supporting information diet: Information diet describes the exploration behavior of an efficient information hunter who would collect only a little bit information from one place and move to others[2]. To support this behavior, an exploration system needs to show essential insights in plain views with details.

- Showing unvisited areas: Even though not all information is worth visiting, an exploration system should let users be aware of unexplored areas and decide

whether an exploration should visit those places. It's also a good way to break the information bubble and introduce new information.

1.2 Projects that Inspired the "Land, Modify, and Shift" Mantra

The "land, modify, and shift" mantra was inspired by three exploration projects I conducted. Although they are remotely different in motivations, use scenarios, and targeted data structures, I found myself keep revisiting similar design concerns. For example, how can a system break the information bubble and avoid redundant information in exploration? How can the system control the size of data in a display intuitively to avoid overwhelming users? How can the system navigate a user to generate exploration tasks efficiently and confidently? How can the system ensure the coverage of information at the end of an exploration process?

Apparently, the solutions are different from project to project. However, there are many shared features that can be summarized using the aforementioned "Land, Modify, and Shift" mantra. In addition, there are many design guidelines that can be extracted from those projects for future works, which will be discussed in Section 6. The three projects are briefly introduced as follows and their full details are presented in Chapters 3, 4, and 5.

1.2.1 Zooming in Animation

Area-based zooming brings many benefits to visualization: it allows users to examine the context of an interesting object by zooming in the area where the object resides, and see the labels of its neighbor objects. However, such an area-based zooming technique is not properly supported in animated visualizations, because the objects are changing their positions. A user will lose content in the zooming area during the animation play.

I designed a new zoomable user interface, called the spot-tracking lens, for animated bubble charts using the "land, modify, and shift" mantra. It enlarges the area around

a focal object and follows it during the animation. A user can see the behavior of the focal object as well as other objects that are close to it momentarily. It is also coupled with a rich set of auxiliary techniques to against change blindness[3] and support advanced analysis. Especially, our new "spotlight" highlighting technique allows a user to quickly see the top N items related to the focal object in each frame. In this way, the spot-tracking lens tells a story of the focal object, and introduces other characters on the stage when they appear in the main story. A user can change the focal object during the analysis, usually by picking a new focal object within the lens. It is an effective and complying way to collect item-based insights. It eventually builds a knowledge map of the dataset. The zooming interface is coupled with an overview that gives a big picture as well as tracks the exploration history. However, synoptic insights are overlooked, which will be the focus of the third project.

1.2.2 Interactive Movie Finder

Current item-based movie recommendation approaches suffer from the information bubble problem, namely a user is trapped in a "bubble" created by movies very similar to the example movie while he/she may prefer more diverse recommendation. Automatic approaches can improve the diversity of recommendations, but users cannot customize the results based on their drifting interest.

My interactive movie finder (IMF) allows users to explore a movie dataset and discover diverse information following the "land, modify, and shift" exploration mantra. A user starts an exploration by picking a movie as an example. Instead of giving one list, IMF provides several browsing lists associated with the focal in different perspectives. The user can pick one list to follow. Through interactions, the user can also customize the order in each list. It encourages the user to find more diverse movies. On one hand, the user can find movies through weak links, aka find movies similar to the ego in a few aspects. On the other hand, by shifting the focal movie, the system captures changes in the user's mind as he/she sees new movies.

1.2.3 EventPool

Comparing is one of the most fundamental tasks in data analysis. With comparison, numerical data can be converted into a few manageable, meaningful categorical values. Analysis based on comparison results is more semantically meaningful than analysis based on numerical data. However, it is hard for an exploration task to take advantage of this kind of semantic-based analysis because it is hard to setup clearly-defined comparison tasks.

Again, the "land, modify, and shift" exploration mantra was applied to address this challenge for time series datasets, which resulted in a system called EventPool. In EventPool, a user "lands" on a time series dataset with an initial comparison task. The dataset is projected to an event sequence dataset, in which events show when the comparison results of an object are certain semantic values. These events are visualized with timelines. With rich interactions and coordinated views, the user can quickly understand and evaluate the comparing results. He/she can also adjust the comparison parameters and modify the visualization to support different tasks. Multiple comparison modes integrated into EventPool extend exploration possibilities.

CHAPTER 2: Related Work on Visual Exploration Mantras

2.1 Egocentric Approach

An egocentric approach is usually used to analyze the topologically related information patch space in a network related to a focal object. There are typically two categories of tasks in an egocentric analysis. At the microscopic level, analysts study how structures and attributes of ego-networks affect the focal object's behaviors. For example, romantic relationships between two people can be recognized based on how their mutual friends are connected [4]. Prell et al. suggested the number of alters, the strengths of ties connecting the ego and the alters, and the number of interconnections between alters as the three major properties typically studied in microscopic level ego-networks [5]. At the macroscopic level, analysts study the structural patterns in a group of ego-networks. For example, several works classify ego-networks into various categories, revealing different social communication patterns [6, 7]. These works consider global attributes such as network size [8] and density [9].

In theory, there is no limitation on how big a related information space should be. However, most works focus only on the nodes directly connected to the focal object [10, 11]. Navigation is needed when the egocentric network extends. Lee et al. suggest a directly manipulate interaction set described as "planting a seed and watch it growing" [12]. Based on that, Ham et al. [13] introduce the concept of doi (degree of interest) and multiple types of context information, such as essential attributes, associated content, and user interactions when measuring the relevance of each node to a focal object. Crnovrsanin et al. help users navigate in an egocentric information space by predicting a user's analysis intent and making recommendations [14]. Vizster [15] clusters nodes in the ego-network into communities based on linkages. EgoNet-

Cloud [16] introduces simplification methods to deal with large-scale ego-networks. Its approach optimizes the topological and temporal context of an ego node.

In a network dataset, the nodes directly or indirectly connected to a focal object define its information neighborhood. Such a definition can be readily extended to other data structures, as "edges" can be considered as the pair similarity between data points. Similar items and their relation to a focal object create its information neighborhood.

As far as I know, very few works have applied the egocentric approaches to exploratory analysis. There are some works exploring "user-centric" information neighborhoods in emails [17] and browsing histories [18]. However, none of them are targeted on exploring a whole dataset. The reason is that egocentric approaches may trap users in an "information bubble"[19] and prevent them from perceiving information that is less relevant to a focal point.

My mantra borrows the concept of "ego" from the egocentric approach. However, the egocentric approach is flat, namely there is only one topology among the nodes. Therefore, it is difficult for a user to change the visualization. My mantra model allows users to redefine information topology based on tasks. Thus it will provide more diverse information. Also, most existing works use node-link diagrams in visualization. My mantra can be applied to a variety of data models and thus allows more complex visualization and analysis.

2.2 Information Exploration Theories

Although we are hunger for information by nature in order to make sense and adapt our world, only a small amount of the information we encounter is actually relevant our tasks. Exploration strategies are employed to control our rate of information consumption to ensure that only the most pertinent information reaches us at any given time. This section introduces existing theories in information exploration and how the "land, modify, and shift" exploration mantra related to it.

Berrypicking: Exploration starts without clear tasks. Instead, users identify useful information encountered during the exploration, which may inspire users with more questions in return. Thus, information collection in exploration is never satisfied by a single final retrieved set, but by a series of selected fragments of information at each stage of the ever-modifying analysis. Bates [20] first noticed that new information encountered gives an explorer new ideas and directions to follow in 1989 and named it "berrypicking". Bates found that library users follow footnote chasing, citation chaining, reviewing a journal series, and search for summarizing works by an author. The search starts with a vague information need. From it, however, users iteratively seek and select relevant information related to it and end up with different paths. In the digital age, users can even create complex tactics to move forward in berrypicking[21]. My "land, modify, and shift" allow users to follow interested information in two ways: modifying the visualization to reconstruct the tasks, and shifting the ego to change the view point, each is supported by a rich set of interactions to meet different needs.

Information Diet: Information foraging theory [22] attempts to explain how users optimize their process of collecting relevant information using an information patch model and a diet model. It assumes that the desired information are distributed in a dataset in a patchy manner. In an optimal exploration, a user moves between information patches quickly and acquires knowledge rapidly when a relevant patch is found. When users moves between patches, they will try to maximize the rate of information gain.

Therefore, users need to evaluate the worthiness of a particular path[23], or "information scent". Card et al.[24] later define information scent as a user's "perception of the value, cost, or access path of information sources obtained from proximal cues". My "land, modify, and shift" is coupled with flexible interactions to ensure users build exploration tasks effectively and efficiently.

CHAPTER 3: Zoomable Animated Bubble Chart

Animation is an appealing and intuitive way to show the dynamic behavior of temporal information. However, it is not widely favored in visual analytics because human vision is quite limited at tracking and remembering multiple changes. Animated bubble chart [25] visualizes time series with two dimensions. Each frame in the animation is a scatter plot showing objects. It is efficient when there are only a few moving objects. However, it quickly loses its attractiveness as the number of animated objects increase to more than a dozen, since human recognition is quite limited on tracking moving objects simultaneously. Such an information overloading problem can be addressed if I can reduce the moving objects in a view. In this project, I applied my mantra to achieve this goal.

Zoomable user interfaces [26, 27, 28] are widely used in information visualization systems. They allow users to magnify the view and thus see more details. Zooming brings many benefits to visualization: it allows users to examine the context of an interesting object by zooming in the area where the object resides; labels overcrowded in the original view can be displayed without overlaps after zooming in; it allows users to focus on a local area and thus reduce their cognitive load.

In spite of these benefits, zooming is not as well supported in animated visualizations as it is in static visualizations due to several challenges. First, objects are typically moving rather than staying still during an animation. Therefore, zooming has to be tightly coupled with panning for users to track objects of interest. Manual panning requires a lot of user effort to intensively follow moving objects, while automatic panning may cause difficulties for users to perceive the absolute speeds of objects in the view when the camera is moving unpredictably. In addition, users may

lose their orientation with a frequently moving camera. Therefore, techniques helping users sense the speeds of objects while maintaining orientation need to be developed, which is critical to facilitate zooming coupled with automatic panning. Second, even after zooming in, multiple objects moving in varying speeds in the view can still cause huge cognitive loads on users. Users are prevented from effectively gaining insights from the data due to the change blindness problem [29]. Therefore, new interaction techniques based on the dynamic features of animation need to be developed to reduce change blindness so that users can enjoy the full benefits of zooming in an animated visualization.

Toward this goal, I present the spot-tracking lens (see Figure 4.1), a new zoomable user interface implemented on animated bubble charts. It is designed to effectively conduct the following tasks: (1) to zoom in and track a moving focal object with automatic panning. No manual panning is needed; (2) to sense the speeds and directions of the objects in the global data space from the zoomed view during the animation; (3) to capture significant dynamic events related to the moving objects in the immediate context of the focal object; and (4) to examine and follow up on the varying relationship between the focal object and its neighbors. The tasks are addressed through the following techniques:

- To support task 1, I design a moving and tracking lens providing an ego-centric view following a focal object. It couples zooming with two distinct automatic panning techniques. The stepwise panning simulates a user manually panning the view to track the focal object; and the continuous panning smoothly follows the focal object, mimicking a long shot in movies.

- To support task 2, I use a frame of reference which enhances the perception of the speeds and directions of moving objects in the zoomed view. A frame of reference generation technique has been proposed (Figure 3.1).

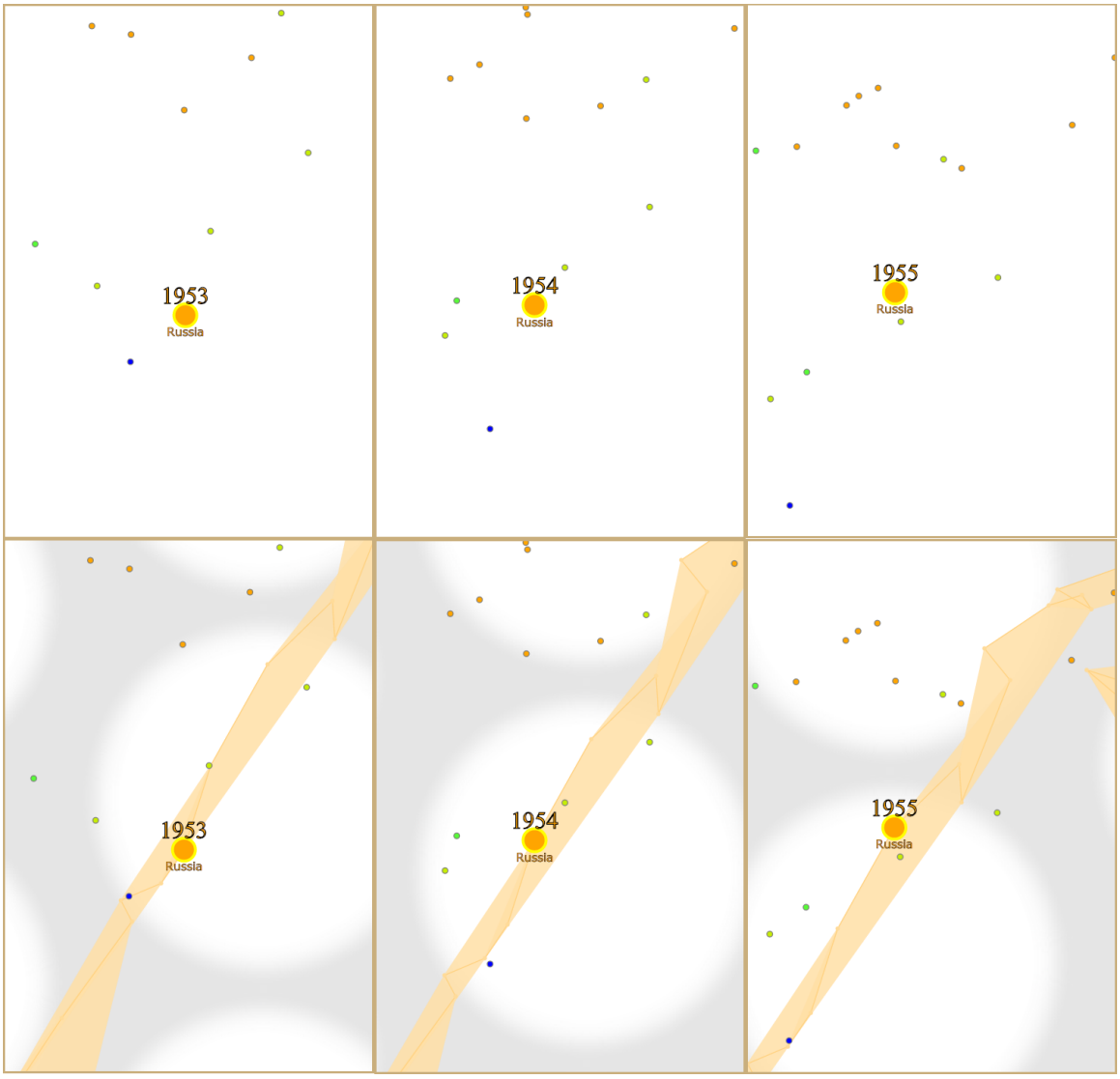


Figure 3.1: The frame of reference. Top row: screenshots of an animation showing a focal object moving from 1953 to 1955. No reference frame is used. Bottom row: screenshots of the same animation with a reference frame. It is easier to sense the movement of the focal object in the bottom view.

- To support task 3, I propose several task-oriented automatic labeling techniques. The labeled objects can direct user attention to significant events in the focal area during an animation.

- To support task 4, I add a novel spotlight to the tracking lens. It greatly augments users' capabilities to track a moving focal object and examine its immediate context. With the spotlight, users can keep their attention focused on the immediate context of the focal object, while perceptually sensing a set of objects of interest outside the immediate context (Figures 4.1 and 3.3).

- Furthermore, I propose to use scrolling to interactively control an animated bubble chart. It allows users to seamlessly switch among automatic playing, stepwise animation, slow forward/backward, and pausing, so as to improve the efficiency of analysis with animated bubble charts.

Preliminary user studies have been conducted to evaluate the effectiveness of the spot-tracking lens. The studies showed that the interface can effectively conduct the aforementioned target tasks. Moreover, I noticed an unpredicted benefit of the lens: since it allows users to pleasantly follow a focal object and observe significant events happing around it, the new tool greatly promotes ego-centric visual exploration, where users gain many interrelated, detail-rich insights around a focal object. Such long chains of insights are beneficial for sense making and reasoning. Therefore, this spot-tracking lens lends itself a potentially useful tool to promote exploratory visual analytics of dynamic data.

3.1 Related Work

Animated visualization is an important approach to analyzing temporal data since human vision is sensitive to changes and movements. Hans Rosling presented the animated bubble chart in his TED 2006 talk [25], where each bubble represented a country and its x position, y position, size, and color represented attributes of

the country. The bubbles were animated to reveal changes in attributes over time. Rosling's talk invoked a lot of interest in animated visualization. Later, InfoCanvas [30] used moving entities to encode time series data. StreamIT [31] used animation to illustrate the impact of newly arrived objects to the clustering structure of existing objects in streaming data. Visual Sedimentation [32] presented animated displays that imitated real-world sedimentation processes to represent data streams. Graph-Diaries [33] used staged animations and complementary small multiples to help users understand changes between individual time stamps in dynamic networks.

However, recent studies reported that although animated visualization is enjoyable and exciting [34, 35, 36], it can be problematic in supporting analysis tasks. For example, Robertson et al. [35] presented a user study comparing the animated bubble chart, small multiples, and traces for presenting and analyzing temporal data. The results showed that the animated bubble chart was the least effective in analysis. There are several reasons for this phenomenon. First, a large number of moving objects clutter the screen and tracking them at the same time is beyond the cognitive capabilities of ordinary users. Robertson et al. pointed out that users can be confused by having too many data objects moving in the display in their user study [35]. Farrugia and Quigley also addressed that, when conducting analysis in animation, users can easily get lost when too many data objects are moving on the screen [34]. Second, the complex manner of movement in animation may lead to change blindness [37], where users miss important temporal patterns [29, 38]. Change blindness can seriously harm the effectiveness of visual exploration in animated visualizations.

Clearly, a Zoomable User Interface (ZUI) can be a solution to the first problem by allowing users to focus on a smaller set of moving objects in a less cluttered display. There exist many examples of ZUIs. Cockburn et al. [27] presented a survey which discussed design issues of ZUIs and summarized existing zooming applications, toolkits, and models, among other categories of interfaces supporting both focused

and contextual views. Pad++ [26] was a toolkit that allows users to control where they look on a vast data surface by panning and zooming. Pocket PhotoMesa [39] used a ZUI to allow users to explore large image collections on small devices. PolyZoom [40] is a recent ZUI for multiscale and multifocus exploration in 2D visual spaces.

Zooming is often coupled with panning through automatic algorithms. For example, Jark van Wijk et al. [41] discussed how to support smooth animations from one view to another by automatically coupling panning and zooming. The speed-dependent automatic zooming [42] automatically changed the zoom level according to the panning speed. The spacetree application [43] automatically panned a view when users selected a different focal node in a tree visualization. Our approach uses automatic panning to track a moving focal object in a zoomed view, which is a scenario different from the existing examples. In existing approaches to visualizing moving objects, researchers have noticed that objects of interest may run out of the zoomed view and called it the "off-view" problem. Existing solutions, such as Canyon [44] and Halo [45], have been focused on providing location awareness of interesting objects off the view without panning the view. By panning the view following a moving focal object, our approach provides an "object-centric" solution rather than the existing "location-centric" solution, which distinguishes our approach from existing approaches.

Many suggestions have been given in the literature on how to reduce change blindness in animations. Examples include ensuring that time intervals between different scenes are long enough, limiting the number of changes through filtering and other interactions, and keeping users oriented with additional visual indicators [46]. Many of these ideas have been used in our design. In particular, our approach allows users to flexibly control the animation speed, automatically labels the most important objects while suppressing the visual representations of the other objects, provides a spotlight to guide users to follow a focal object and its immediate context, and uses a reference

frame to keep users oriented.

In most animation systems, analysts control the animation by widgets such as buttons and time sliders. Since these widgets are usually far away from the focal area, interacting with them interrupts users from watching the focal area. Archambault et al. [47] found in a user study that subjects rarely used the play controls, even though using them could reduce the analysis time. In addition, it is difficult to display a time slider that is long enough for precise speed control. Kondo and Collins [48] attached a timeline to the trajectory of a focal object, making play control more tangible. However, this technique only applies to touch screens. I propose a different approach to control animation with scrolling. Scrolling techniques have been used in story telling websites to control the slides playing speed [49]. Our approach is among the first efforts to control animated data visualizations using scrolling.

3.2 The Spot-Tracking Lens

The spot-tracking lens works as follows: (1) Activation: After users select a focal object from the overview animation (the animated bubble chart without zooming), they can use the + key to zoom in. The animation is paused and a frame of reference and a spotlight is added to the visualization. (2) Analysis: The users either set an automatic play mode or use scrolling to resume the animation. During the animation, the camera roughly follows the movement of the focal object automatically through automatic panning. The frame of reference helps the users sense the direction and speed of the focal object and other objects in the view. The spotlight highlights the focal object and its immediate context so that the users can effectively perceive interesting objects in this region and monitor their dynamically changing relationships with the focal object. If dynamic labeling is turned on, only the labels of the most salient objects, such as the ones that are the closest to the focal object, are displayed. The users can use scrolling to control the animation, such as moving slowly forward/backward to search interesting events and then pausing to examine

the detail. (3) Adjustment: At any time during the animation, users can change the zoom level using the + or -key or set another object as the focal object by clicking it or searching it by name from a search box. In the following sections, I present these techniques in more detail.

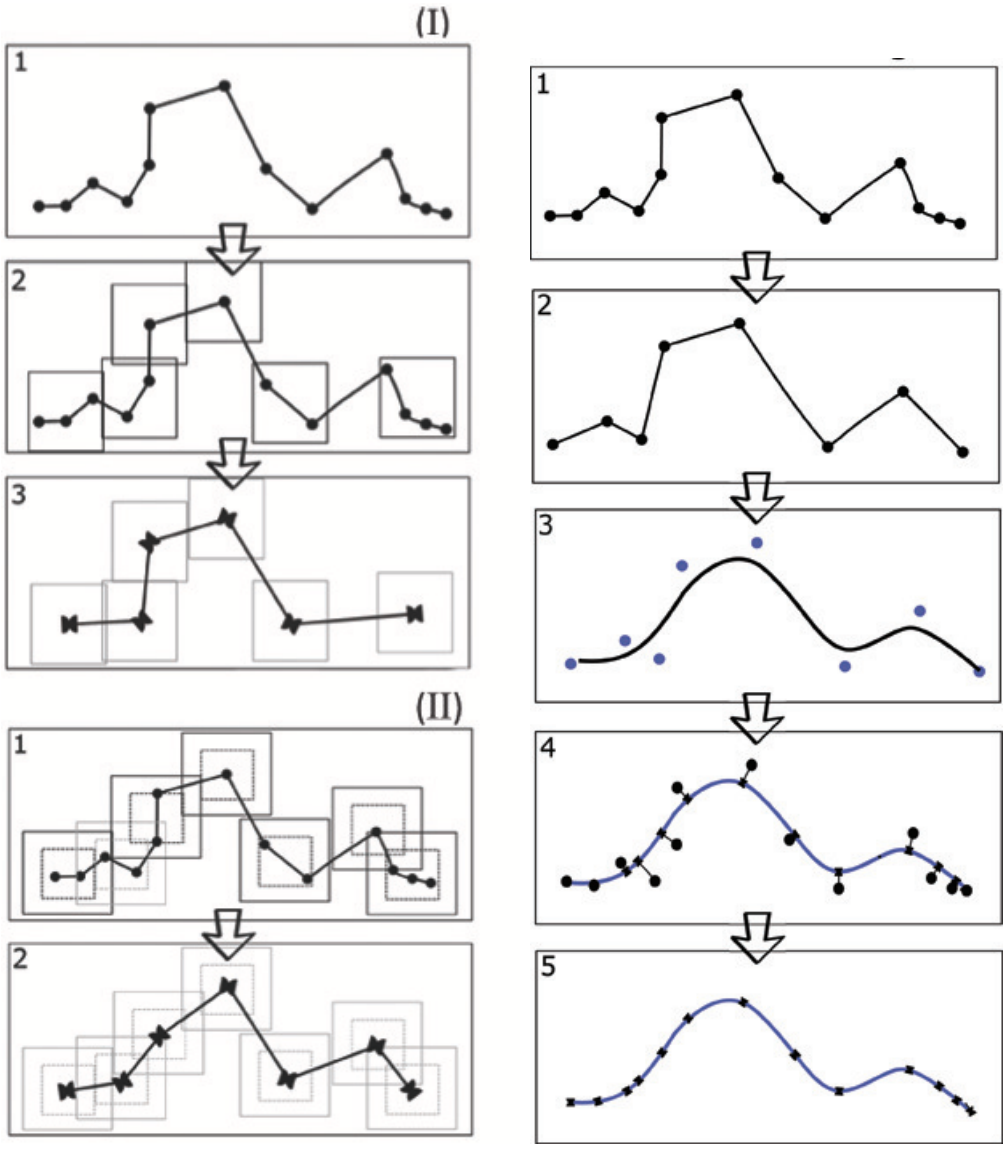
3.2.1 Automatic Panning

Three different automatic panning strategies have been explored. In our original approach, the camera is tied to the focal object and thus the focal object is always centered in the view. It is easy to observe the dynamic relationships between the moving objects and the focal object. However, this simple approach makes the camera "shake" intensively when the trajectory of the focal object is not smooth. Consequently, the unpredictable panning makes sensing the speeds and directions of the objects extremely difficult. To conquer the problem, I develop two automatic panning methods to move the camera positions smartly so that the resulting camera path is smooth and stable: Stepwise panning simulates the drag-and-drop panning effect in manual panning; and continuous panning allows the camera to smoothly follow the focal object without jitter. These two automatic panning algorithms are implemented in three steps: (1) find a sequence of view center positions according to the full trajectory of the focal object and the zoom level; (2) determine the starting time and end time of panning when the view center moves from one position to the next position; and (3) interpolate the real motion path of the camera between the two positions with smooth transitions.

Stepwise Panning: I adopt a strategy for stable camera motion which pans the camera only when the focal object is about to move out of a current camera viewport. Each camera position is computed to cover as many subsequent focal object positions along the trajectory as possible. This strategy mostly simulates the drag-and-drop panning operation when a user manually traces a moving object. When a focal object makes a sequence of small movements around the view center, camera panning is not

triggered to follow such shaking variations so no extra cognitive load is imposed on users. Figure 3.2a shows the implementation of this method: First, create a sequence of bounding boxes (I-2) along the focal object trajectory (I-1) to cover all the focal object positions. Each box has the same size as the size of the zoomed view. The box is computed to cover as many focal object positions as possible. Second, the centers of the bounding boxes are computed to decide the camera positions along the animation (I-3). This method is further optimized so that users can examine the context of the focal object better. In detail, the bounding boxes (II-1) are set to be smaller than the size of the camera view. With this approach each frame of the animation ensures a minimum distance between the focal object and the view boundary (II-2). Panning starts when the focal object leaves the last position in the current view and ends when it reaches the first position in the next view.

Continuous panning: Continuous panning follows the focal object in a smooth long shot, akin to a camera following a runner in a race in movie recording. Figure 3.2b illustrates our method for determining the view center positions. First, to reduce jitters, the focal object trajectory (Figure 3.2b-1) is simplified to a trajectory with less joints (Figure 3.2b-2) using the Douglas-Peucker algorithm [50]. To preserve more details in a larger zoom level, the tolerance parameter of this algorithm is associated with the zoom level and maps to approximately 100 screen pixels in the zoomed view. Second, joints in the simplified trajectory are used as control points (Figure 3.2b-3) to create a B-spline curve [51] . Third, for each focal object position, I calculate its closest point on the curve (Figure 3.2b-4). These points are used as the view center positions (Figure 3.2b-5). To avoid jitters, the order of the view centers on the curve is kept as the same as the order of the focal object positions. Whenever the focal object moves to a new position, the view center also moves to the corresponding nearest point on the curve. In this way, the focal object is usually close to the view center and the panning is smoothed.



(a) Stepwise panning (b) Continuous panning

Figure 3.2: Illustration of view center trajectory generation.

Discussion: Different automatic panning techniques work better on different focal object trajectory types. When the focal object is rambling within a small area, the stepwise panning works better since it leads to very few panning operations. On the other hand, continuous panning is preferred when the focal object is moving quickly along a roughly smooth path, as continuous panning inherits the nature of the object's movement and causes less jump and jitter. In our current system, users can choose one of the two techniques to fit their needs. Automatically choosing and switching panning techniques during animation remain as future work.

3.2.2 Frame of Reference

When automatic panning is conducted, users have difficulties in interpreting the movements of the objects in the zoomed view due to the combination of camera panning and object movements. For example, when the camera quickly moves upward following the focal object, the users may feel that other objects moving slower than the focal object in the same direction are moving downward. To address this issue, I introduce a frame of reference. When I sit in a train, I know the train is moving forward when I see the landscape outside the window is moving backward. The landscape plays a role as the **frame of reference**. Similarly, I provide a frame of reference which is static in the whole domain. It is displayed as the background of the zoomed view to help users sense the speed and direction of the moving objects. This background should be designed so that it is non-distracting to avoid adding visual noise to the animation. In our prototype, I design and combine two types of backgrounds as the frame of reference. One is a silhouette of the focal object trajectory and the other is a texture with symmetric patterns (see Figure 4.1).

In our initial design, I used the focal object trajectory as the background. As a reference, it is always visible in the zoomed view and provides additional information such as the trend of how the focal object moves. However, a raw trajectory itself can be cluttered lines and be visually complex and distracting. Therefore, I now use

a colored silhouette of the trajectory as a reference, which captures the outline of the trajectory. It is less cluttered and distracting while still preserving the mobility trends of the focal object. The silhouette is created using the following algorithm: Starting from N = 1, create a convex hull to cover all the positions numbered from N to N+M along the focal object trajectory (where M is a positive integer). A smaller M preserves more details of the trajectory variation, which is preferred when the zoom level is large. This process is repeated by increasing N until all the points on the trajectory are covered. Then, I combine all the convex hulls to form the trajectory silhouette. The shape of the silhouette reveals the temporal trend of the focal object. For example, the silhouette is slim and smooth when the focal object is moving consistently toward a direction. The silhouette is semi-transparent, which makes it easy to observe but not distracting.

I use the silhouette together with a texture of symmetric patterns (see the background circles in Figure 4.1) as the frame of reference. The symmetric patterns improve the situation awareness of users so that the movement of the objects is immediately perceived. Here the size of the circles in the texture indicates the zoom level.

3.2.3 The Spotlight

In stage arts, a spotlight is often shed on a focal character and moves synchronously with her. The characters in the dark do not attract attention unless they get close to the focal character. The spotlight follows the focal character to illustrate an ego-centric story around her: at times, other characters run into the spotlight to meet her or she approaches and leaves other characters. I design a spotlight in the spot-tracking lens. As shown in Figure 3.3, the spotlight is always centered on the focal object, highlighting its immediate neighborhood as a focal area. During the animation, the spotlight moves synchronously with the focal object, emphasizing its neighbor domain, attracting user attention to the object and its context, and allowing

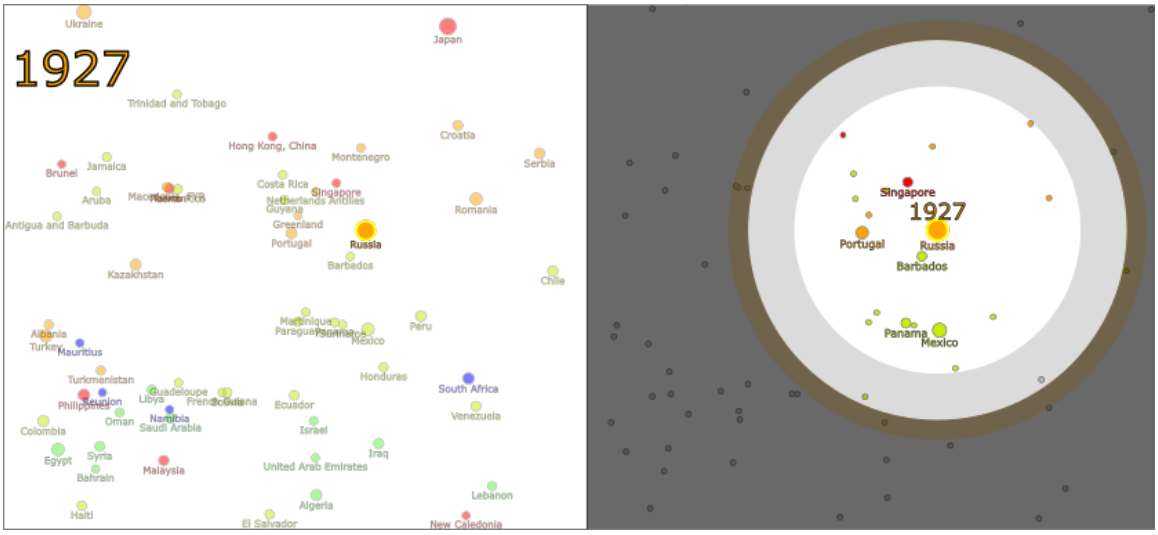


Figure 3.3: The spotlight. Left: A screenshot of an animation. Non-focal objects and their labels are semi-transparent. Right: The same animation with the spotlight turned on. Automatic labeling is turned on for highlighting fast moving objects near the focal object.

users to examine details in the focal area.

There are two ways to get attention: let the focus be more attractive, or let everything else be less attractive. Both strategies are used in the spotlight. To make the focal object more attractive, it is highlighted by a time stamp displayed above it. To make objects far away from the focal object less attractive, the area outside of the spotlight is covered by a semi-transparent dark gray mask. The objects in this area are shown as small, dark circles, which are not attractive in the dark gray background. Their labels are also turned off. I further deliberately design the coloring of the spotlight. Inspired by five tone shading [52], a basic technique in painting for drawing how lighting effects an object, several tones are used in the spotlight to make it look realistic. A light gray middle tone area provides a transition between the spotlight area and its surrounding context. Outside the middle tone, a thin darker shadow area enhances the spotlight's boundary. It also indicates the boundary of the region where the objects are readable with their original labels, sizes, and colors. Finally, I slightly decorate the shadow using the hue of the focal object.

Users can interactively turn off/on the spotlight or adjust its size and the transparency of the dark gray mask outside it. They can also change the appearance of objects inside/outside the spotlight by toggling color (colorful or grey) and size (varying sized or small circles).

The assumption underlying the spotlight design is that objects outside of the focal area are unimportant since they are far away from the focal object. However, in the dynamic environment, an object can move close to the focal object in one moment and far away in the next. In this situation, users may keep a long term interest in it. For example, they may want to check if the object comes close to the focal object again or compare the overall movement of the object with that of the focal object in a longer time sequence. To support this need, a tracking interaction is provided where users can click an object in the view to highlight it. Highlighted objects are always displayed with labels and are never degraded and masked. Therefore, they are easily seen in the dark background when they move out of the focal area (Figure 4.1). When they move out of the zoomed view, they are displayed as half circles hanging on the boundary of the view, inspired by the off-view object display solution proposed in Halo [45] and Canyon [44].

3.2.4 Selective Labeling

Labels provide important semantic information in visualizations [53]. In animated visualizations, manual labeling approaches may require intensive human efforts since insights and relevant objects may change over time. The "label all" strategy may cause clutter even in a magnified view. I propose a new labeling strategy called selective labeling. It automatically labels or de-labels objects of interest or objects that are no longer of interest during the animation based on pre-defined user tasks. The selective labeling is implemented as: (1) at each moment, calculate an importance value for each object in the view and rank them from high to low; (2) label the top N objects (N is set by users to show more or fewer labels).

I now present how to automatically label objects that are the nearest to the focal object. Note that I can observe objects passing by the focal object during their movement. Users often consider them as close to the objects and want to identify them. It is different from static visualization where only static positions need to be considered. I propose the following algorithm to determine the importance values of objects according to such closeness:

Considering two objects p and s, whose locations at time step 1 are s_1, p_1 and whose locations at time step 2 are s_2 and p_2, I interpolate their moving path between the two time steps. The minimum distance between them can then be computed as

$$\min_{d\in[0,1]} \parallel (\overrightarrow{p_1} * d + \overrightarrow{p_2} * (1-d)) - (\overrightarrow{s_1} * d + \overrightarrow{s_2} * (1-d)) \parallel \tag{3.1}$$

where d $(0 < d < 1)$ is the time between the two steps. The average minimum distance between an object and the focal object in the time window is the importance value to be used in the selective labeling algorithm.

The importance value can also be defined in different ways to support different tasks. For example, using the moving speeds of the objects as the importance values can capture objects that are moving fast. They are changing dramatically and are thus worth the attention of users. An object can be labeled prior to a fast movement to prepare users to capture volatile patterns.

3.2.5 Scrolling

Unlike watching a movie, exploratory analyses require full control of the playing speed. The spot-tracking lens is implemented in a webpage and uses scrolling to control a virtual timeline of the animation, inspired by the parallax scrolling websites [49]. Users can scroll the mouse wheel to drive the animation and the scrolling speed determines the animation speed. For example, users can slowly scroll the wheel up or down to move the animation slowly forward or backward to capture volatile patterns.

They can also pause the scrolling to pause the animation so that they can check labels or highlight interesting objects. With scrolling, users can easily change the animation speed without the penalty of being interrupted by switching their attention to adjust a slider. Users can also drag the scroll bar to jump to a certain position of the timeline, click the left/right mouse button to move forward/backward stepwise, or start an auto-play mode by clicking a function key.

3.3 Spotlight User Study

The spotlight combined with a zoomed view tracking a focal object is an important technique in the spot-tracking lens to achieve our goal of improving animated bubble charts. A user study has been conducted to evaluate its effectiveness in helping users follow a focal object and examine its immediate context by comparing it with a benchmark technique. In Gapminder World [54], the state-of-art animated bubble chart prototype, the focal objects are displayed in dense colors and all other objects are semi-transparent. This approach was used as the benchmark technique. Our hypotheses were: (1) The spotlight allows users to examine a focal object and its immediate context more easily, especially when the object is moving in the view. (2) The spotlight allows users to examine a moving focal object and its immediate context more easily when there are distant objects moving in the view. (3) The spotlight allows users to track a small set of distant focal objects more easily while still focusing on a moving focal object.

To control distracting variables, animation videos were used in this user study. They were recorded from two systems. Both systems provided a zoomed view of an animated bubble chart with automatic panning. All objects in both systems are represented by colorful, varying sized bubbles with labels. The focal object was highlighted by a thin yellow halo and was opaque. The test system casted a spotlight on the focal object while the benchmark system displayed non-focal objects and their labels semi-transparently (alpha was set to 0.5. See Figure 3.3 on the left). The

background color was white and no frames of reference were used in both systems.

A subset of a world wealth and health dataset downloaded from Gapminder [54] was used in this study. It contains data about 200 countries since 1900. The populations of the countries were mapped to the bubble size. Life expectancy was mapped to the y axis in a linear scale and average income was mapped to the x axis in a logarithmic scale. Countries are grouped into 6 geographical regions: Asia and Pacific, Sub-Saharan Africa, Middle East and North Africa, South Asia, America, Europe and Central Asia. The regions were represented by the color of the bubbles.

Eighteen graduate students majoring in Computer Science participated in this study through an online survey and a within-subjects design was used. Five videos were embedded in the survey with a question under each of them. Each video lasted about 30 seconds, including two 8-second sections separated by a 15-second black screen. The two sections recorded an animation generated by the benchmark system and the test system, respectively. The two animations in the same video had the same focal object(s), time period, animation speed, view centers, panning path and speed, and zoom level. The impact of the order of the animations to the study results was ignored since the subjects could repeat a video as many times as possible before they answered the question and only user preference data was collected.

Two videos were used for evaluating the first hypothesis: the "fixed" video showed a focal object fixed in the center of the display during the animation; and the "moving" video showed a focal object moving in the view. Two videos were used to evaluate the second hypothesis: the "cluttered" video showed a moving focal object with a lot of distant objects moving in the view; the "sparse" videos showed a moving focal object with only a few distant objects moving in the view. When the subjects watched the four videos, they were asked to speak out the first letter of the country closest to the focal country in every year. Instead of the whole name, just saying the first letter made the task easier for non-native English speakers. The purpose of this task was to force

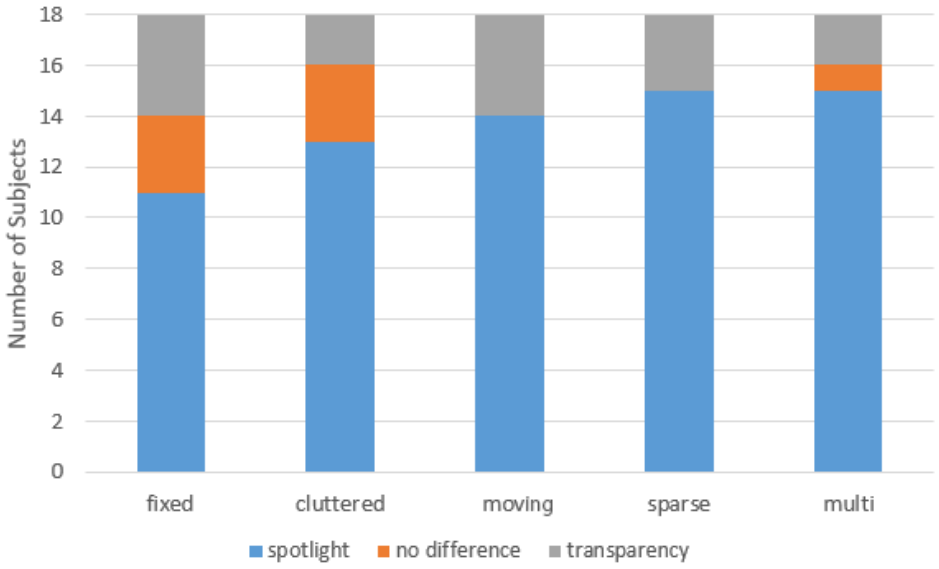


Figure 3.4: Videos preferred in the spotlight user study.

the subjects to follow the focal object and pay attention to its immediate context. In the "multi-tracking" video evaluating the third hypothesis, four secondary focal objects were highlighted by thin yellow halos and dense colors and moving around in the view. The subjects were asked to track them while paying attention to the moving focal object. For each video, the subjects were asked to select the video section that allowed them to conduct the task more easily or indicate no difference.

Study results are shown in Figure 3.4. The spotlight was preferred by more subjects than the benchmark technique in all the scenarios and all the hypotheses were supported. The difference was the biggest in the multi-tracking scenario (15 vs 2). Even when the focal area was fixed in the center of the view, the spotlight was still preferred (11 vs 4). I notice that when there were more objects moving outside the spotlight, the number of subjects preferring the spotlight dropped from 15 to 13. It suggests that it is beneficial to mute the distant objects (displaying them as small, gray circles) in the spot-tracking lens (they were not muted in this experiment).

3.4 Open-Ended User Study for Spot-Tracking Lens

In this preliminary user study, I investigated how the spot-tracking lens changed the subjects' visual exploration strategies and the types of insights they captured in analysis. I also recorded several cases generated by the subjects to illustrate the utility of this zoomable user interface. Two systems were used in this study: one was Gapminder World (Gapminder for short) [54] as the benchmark; the other was the spot-tracking lens interface presented in this project. I wondered whether our interface could provide an alternative perspective in the exploration of time series data.

Four master students with visualization background participated in this user study. They were divided into two groups and the subjects in the same group were familiar with each other. Each group was asked to have teamwork to explore world history after 1900 using the systems. They spent one hour on each system, with a break of at least one hour between the two sessions. The two sessions were conducted in the same day. One group used Gapminder first and the other used the spot-tracking lens first. The instructor trained the subjects on how to use the system before each session. She observed all the sessions and video recorded them. The subjects were told that insights about the relationships among multiple countries were more valuable than insights about single countries, since the latter can be easily captured by static displays.

First, I present two stories captured by the subjects.

The story around Germany: Subject group A investigated the dataset by setting Germany as the focal object (see Figure 3.5). They turned on the selective labeling to highlight countries close to Germany and ran the animation starting from World War II. They found that after World War II, Germany surpassed many western European countries in both x and y axes within a 20 year period. Figure 3.5 shows a snapshot of the animation after Germany surpassed France and Belgium. After

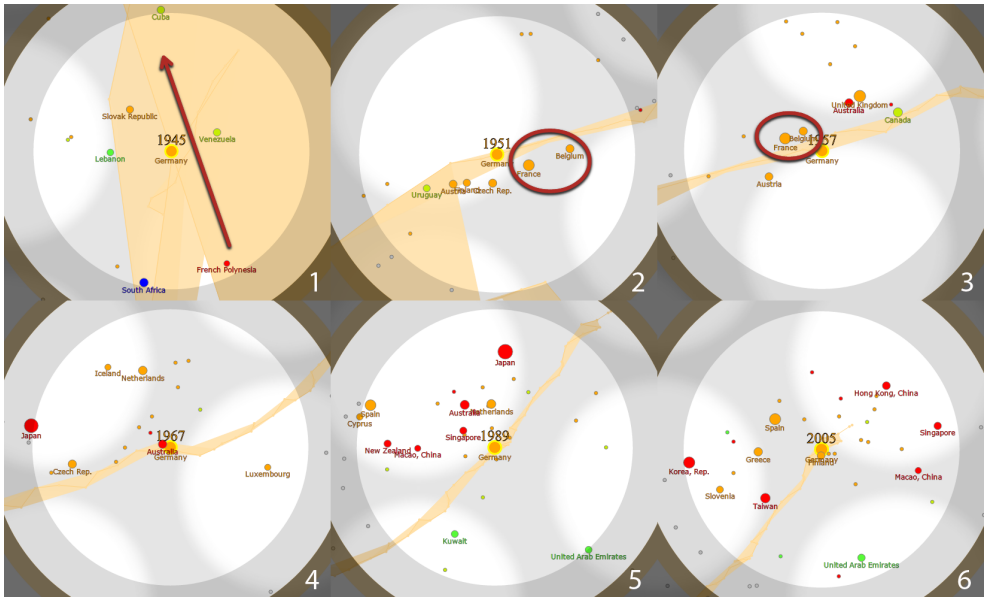


Figure 3.5: The journey of Germany after World War II. 1: Germany recovers from WWII in 1945 quickly. 2 and 3. Germany quickly surpasses other European countries such as France and Belgium. 4. In 1967, Japan moves forward to join Germany's neighborhood. 5. More red bubbles (Asian countries) join Germany's neighborhood in 1989. Meanwhile, Germany retreats a little bit as the result of reunification. 6. Red bubbles continue their progress in surpassing Germany.

this time period, Germany maintained its rapid progress and surpassed the United Kingdom. Then, the subjects noticed that Japan came into the race and joined Germany's neighborhood. They noticed that Japan had been the only Asian country (red bubble) in this neighborhood until 1980, when many more red bubbles appeared in the neighborhood (see Figure 3.5). They argued that Asia had its biggest leap at that time according to this observation.

The story around India:Subject group B selected India as the focal object (see Figure 3.6) and enabled automatic labeling for fast moving objects. In the zoomed view they were able to read the names of fast moving countries in India's neighborhood. They saw countries such as China and Vietnam during World War II. These names made sense to them since they were the World War II participants. After World War II, they were surprised to see that India experienced a slight drop in 1946 while most other countries around it were quickly recovering from the war. They

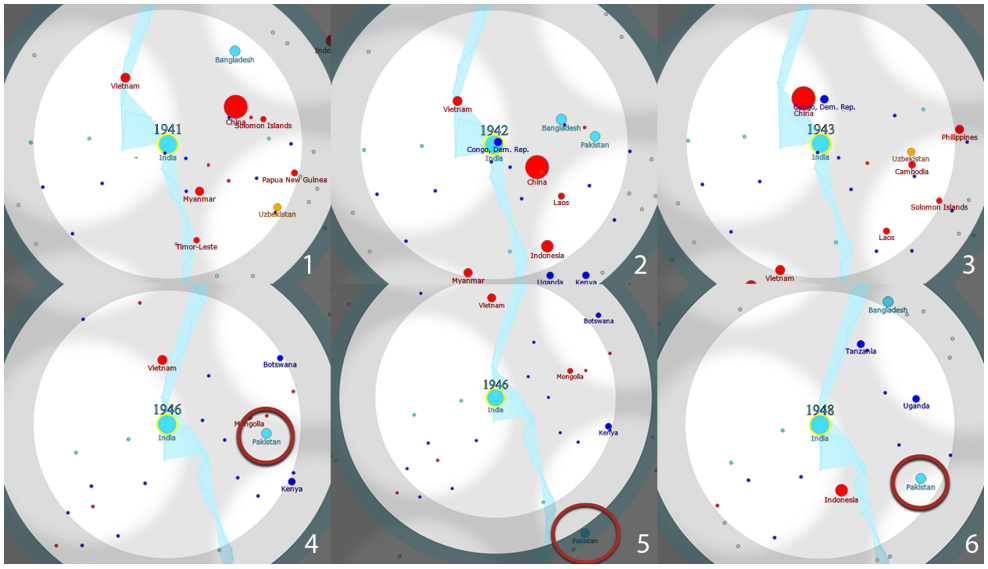


Figure 3.6: The journey of India around its independence. 1, 2, and 3. Asian countries such as China, Vietnam, the Philippines, and Indonesia are in an unstable status as they are involved in World War II. India is relatively stable. 4, 5, and 6. In 1946, countries like Vietnam are recovering from the War. India, on the contrary, drops slightly but soon recovers by 1948. Meanwhile, Pakistan also drops and recovers synchronously.

also noticed that Pakistan had a similar experience. They suspected that it might be relevant to the Partition of India and Pakistan in 1947.

I now discuss the observations from the study:

- Leaps: Gapminder users captured large leaps of big countries easily. They also captured several small countries when they were leaping to a sparse area, but they did not notice leaps occurring in cluttered areas. Spot-tracking lens users captured leaping countries when they were in the view, even in cluttered areas. They were able to capture small leaps such as the one mentioned in the India story. Such leaps are not noticeable in either Gapminder or static displays.

- Inter-country relationships: It was noticed that subjects observed many more inter-country relationship insights in the spot-tracking lens. With Gapminder, the comparison often relied on the subjects' pre-knowledge of a country's history. The subjects had to add trails for two countries to examine their relationships.

- Maintaining attention: As shown in the Germany story, spot-tracking lens users were able to keep their interest in a country longer and discover many insights about it. It was also observed that the subjects jumped to a focal object after finding interesting insights about it when exploring another focal object. The exploration patterns in Gapminder were quite different. The subjects more often followed the mode of generating a hypothesis based on pre-knowledge - verifying the hypothesis using animation - and then generating another hypothesis based on other pre-knowledge. The difference suggested that the spot-tracking lens may be a useful technique for ego-centric visual explorations and progressive visual explorations.

The leap discovery observation confirmed that the spot-tracking lens helps users examine fine details during animation as expected. I did not anticipate the advantages the spot-tracking lens provides in finding inter-country relationships and supporting ego-centric and progressive visual explorations. It hints that the spot-tracking lens may be an engaging and inspiring approach in exploratory analysis of dynamic data, which needs to be investigated in future user studies.

After the subjects finished both sections, preferences and feedback were collected from them. All subjects agreed that the spot-tracking lens and Gapminder were designed for different tasks and they should work together to get more insights. The subjects all gave positive feedback to the visualization design of the spot-tracking lens. They commented that its trajectory silhouette background looked good and was informative; they could read labels easily in the spot-tracking lens; its design was artistic; and it provided the flexibility to go back and forth and control speed through scrolling.

3.5 Project Conclusion and Future Work

In this project, I propose the spot-tracking lens, a new zoomable user interface for animated bubble charts. It couples zooming with automatic panning and a set

of novel techniques, such as frames of reference, spotlight, and selective labeling. It not only allows users to enjoy the expected benefits of zooming, such as focusing on focal objects and their context, examining details and labels in a less cluttered view, but also brings additional benefits into animated visualization. Our preliminary user study suggested that it can be an engaging user interface for ego-centric, progressive visual exploration of dynamic data. Our exploration also reveals that interactions, such as zooming, face new challenges in animated visualization that are not encountered in traditional static visualizations. It is an under-explored research area where new efforts can be challenging yet productive. I argue that new techniques developed in this area may help solve the dilemma that animated visualization is fun and exciting, but less effective than static visualization in supporting analysis tasks. After a complete set of interactions are developed in the future, animated visualization may fully reveal its effectiveness and efficiency in analysis tasks.

The spot-tracking lens still has its limitations as a zoomable user interface. First, users are limited to a small focal area in the zoomed view and thus they do not get a full picture of the dataset. This problem can be addressed by coordinating the zoomed view with an overview to form an overview + detail view interface [27]. Second, in the current implementation, users can only track one focal object at a time, which is not suitable for tasks where multiple focal objects need to be compared. This problem can be leveraged by multifocus zooming techniques such as PolyZoom [40] or distortion techniques where multiple foci can be examined. Third, the spot-tracking lens requires users to manually adjust the zoom level. Since the density of objects in the zoomed view varies over time, the screen space can be used more efficiently if the zoom level can be automatically adjusted to maintain a desired object density in the view. This problem can be addressed by coupling both automatic zooming and automatic panning. In the future, I would like to work on the aforementioned improvements as well as developing new zoomable user interfaces for other animated

visualizations such as dynamic graph visualization.

3.6 Mantra Reflection in This Project

The spot-tracking lens is a collection of new interactions working together to support the "land, modify, and shift" mantra. It is built directly upon the animated bubble chart without interfering the inherited interactions and visual design.

In the spot-tracking lens, the ego is an animated bubble whose center is tied to a focal object. By following it, the animated visualization shows an enlarged area around the bubble. It allows users to easily observe the behavior of the focal object as well as the changes within its the neighborhood over time, as if through a camera. In this way, users can collect item-based insights related to the focal object in a story-telling style. Users can also shift the focal object to another. Usually a new ego gets the users' attention when it appears in the camera with interesting behaviors.

Interactions are important to make sense and accelerate the exploration. The panning technique, along with the reference frame makes it easy to understand the trajectory of the ego. The spotlight controls how many objects should be highlighted and thus prevents overwhelming users. The filtering tool allows users to configure what information to be shown according to different tasks. The scrolling play control supports seamless transition between fast forward and frame-by-frame examination to prevent change blindness.

In addition, an overview is added as a side view for two proposes: first, it shows the position of the camera in the global coordinates. Second, it shows the coverage of the exploration by lighting up the visited bubbles and provides access points to the unvisited ones. It makes users aware of the unvisited areas.

In this project, the ego is carried by an actual object in the dataset. However, it can be easily extended to virtual objects such as the average point of an object group, which remains to be the future work.

This project was initially motivated by the information overloading problem in

animated bubble charts. However, from the user study, I observed how subjects' exploration tasks changed because of the encountered information, which resulted in exotic and unexpected discoveries. It became the first inspiration of the "land, modify, and shift" mantra.

CHAPTER 4: Interactive Movie Finder

I present Interactive Movie Finder (IMF), an interactive movie recommendation system that promotes accurate, novel, diverse, and serendipitous (ANDS) discoveries through example-based, interactive visual explorations of overlapping subspace clusters extracted from movie rating data. The subspace clusters, which group audiences with common latent interests and movies with common latent themes, are extracted from a large scale movie rating dataset using topic modeling. The subspace clusters overlap, revealing the complex characteristics of individual audiences and movies. They build meaningful, multi-way associations among the movies. Interactive movie recommendation is enabled by visually exploring the movie clusters along with movie profiles. Different from traditional overview first visual explorations for clusters, an example-based visual analytics approach customized for recommendation is developed in IMF. A set of cluster refinement and ranking algorithms as well as views and interactions are proposed to provide ANDS recommendations based on example movies provided by users. Case studies, a preliminary user study, and a set of experiments have been conducted to illustrate the effectiveness of IMF.

4.1 Introduction

In this project, I propose Interactive Movie Finder (IMF) (http://goo.gl/7U4WMx), a new visual analytics approach to movie recommendation. IMF leverages recommendation through interactive visual exploration based on subspace clustering. In particular, MovieLens [55], a dataset containing ratings from 71,567 users on 10,681 movies, is used for clustering. The motivation of the subspace clustering approach is as follows: Audiences with different tastes may like the same movie for different rea-

sons. For example, some audiences like "True Lies (1994)" for its rescue theme while other audiences may like its laugh-out-loud moments. The first group of audiences may also like rescue movies such as "Speed (1994)" and "Cliffhanger (1993)". The second group of audiences may also like comedy movies such as "Maverick (1994)" and "The Nutty Professor (1996)". Characteristics of a movie, such as the rescue theme and the comedy elements in "True Lies (1994)", are often latent and one cannot find them explicitly from the movie profile. However, if movies co-liked by audiences with similar tastes, such as "True Lies (1994)", "Speed (1994)", and "Cliffhanger (1993)", are presented side by side, the latent characteristics shared by these movies, will be observable.

IMF applies subspace clustering on MovieLens to identify groups of audiences with similar tastes and clusters of movies co-liked by them. The audience groups overlap since audiences are often interested in different types of movies; the movie clusters overlap since each movie has multiple characteristics. I argue that the following benefits may be achieved if the subspace clusters can be visually presented to users and an interactive exploration approach can be supported for movie recommendation:

Transparency: Visually presenting movies from the same cluster side by side will allow users to observe the common latent characteristics in them. With this knowledge, users can find movies and clusters matching their interests and filter out non-interesting clusters more effectively and efficiently.

Diversity: Starting from one example movie provided by users, multiple movie clusters containing this movie can be recommended to users, where each cluster carries different latent characteristics reflected in the example movie. They provide diverse choices for the users.

Scalability: With the recommendations organized into clusters and the knowledge about their common themes available, users can efficiently browse a large number of recommendations and quickly locate movies of interest. It is more scalable than

traditional approaches where users need to sequentially scan a mysterious, long list of recommendations.

Several challenges need to be addressed to achieve the benefits above. First, the traditional "overview, zoom, details on demand" visual information seeking mantra [1] used by most cluster-based visualization systems does not work well in recommendation applications. The traditional approach focuses on major clusters and their center items, while the interests of recommendation applications users may lie with small clusters or non-center items in a cluster. Thus, a new visual exploration strategy customized for recommendation applications needs to be developed. Second, due to the difficulties in parameter setting [56], subspace clustering may produce clusters with significant redundancy or coarse clusters with multiple true clusters in them. The quality of recommendation may be harmed by the redundant/coarse clusters. Third, the fact that audiences/movies may participate in multiple clusters leads to complex relationships among the clusters, which challenges the visualization design.

To overcome the aforementioned challenges, IMF employs a new subspace cluster visual exploration strategy and a rich set of visualizations and interactions. They are customized for recommendation and powered by automatic cluster recommendation, refinement, and ranking algorithms. The following interactions of IMF differentiate it from existing movie recommendation systems:

Example-based, progressive cluster exploration: In IMF, users select an example movie and multiple related movie clusters are visually presented to them. The different characteristics of the clusters are revealed, so the users can quickly spot clusters of interest. Meanwhile, if the users find a movie describing their interest better, they can set it as a new example, which then brings in new clusters and drives the exploration to a modified direction. This approach allows users to quickly access movies of interest even if they have a rough start.

Interactive cluster inspection: IMF presents multiple movie clusters without any

clutter using a grid of posters. Movies ranked higher in a cluster can be directly seen and the ones ranked lower are accessed with mouse clicks. Movies more relevant to the example movie are ranked higher, so users can quickly discover the ties between a cluster and the example movie. Users can also interactively adjust a popular-obscure scale to make either popular or less watched movies more visible. Therefore, cinema fans can easily find movies they haven't watched yet and occasional movie watchers can find popular movies of interest.

Auxiliary interactions upon profiles: Using movie profiles from The Movie Database (TMDB) [57], IMF allows users to rank clusters and filter recommendations by profile keywords, which makes the cluster browsing and movie finding even more efficient. The differences and similarities between the clusters/individual movies and the example movie in profiles are visually presented to help users decide clusters/movies of interest more efficiently.

Serendipitous map: To prevent users from being trapped in a local area of the information space, IMF provides the Serendipitous Map. It allows users to quickly access less-explored clusters and their center movies for good surprises.

History management and insight management: The history management interaction provided in IMF allows users to quickly go back to any views they explored. IMF also allows users to interactively store movies of interest for later examination.

I conducted two experiments to compare IMF with an item-based collaborative filtering recommendation approach [58]. The experiments showed that IMF allows users to access movies with less effort and provides more diverse recommendations. Our preliminary user study with cinema fans illustrated the effectiveness of IMF in promoting accurate, novel, diverse, and serendipitous recommendations.

Although IMF is a movie recommendation system, its approach is general enough to be applied to other applications where user rating data is available, such as music and product recommendation. IMF may also be useful in information retrieval

applications where overlapping subspace clusters can be constructed. For example, subspace clusters can be constructed on documents and keywords in document collections, and authors and publications in citation networks. Our approaches can be applied to those applications.

The main contributions of this project include:

- IMF, a new approach that leverages movie recommendation with subspace clusters and interactive visual explorations upon them. Experiments and a user study illustrated its efficiency and effectiveness.
- A set of subspace cluster recommendation, refinement, and ranking algorithms.
- A new example-based approach for visually exploring overlapping subspace clusters.

4.2 An Example User Scenario

I briefly introduce IMF with a user scenario. Yuki, a fan of "Spirited Away (2001)", is in a mood to watch some movies. She searched "Spirited Away (2001)" from the search box and set it as the example movie. The initial view is displayed in Figure 4.1.

Browsing clusters In the middle of the view, many movie posters were displayed in horizontal strips. Each strip contained movies from the same cluster. The most relevant cluster was placed on the top strip, which was the default focus and was highlighted. Yuki immediately saw many familiar Studio Ghibli animation movies from the top strip. "The recommendation makes sense but I have watched them already", Yuki said. She pressed the down key to move the focus to the next strip. She found that it contained many Japanese fantasy movies which have styles different from "Spirited Away (2001)". She has only watched a few of them before. As she moved down, more strips were displayed. She found a cluster of Hollywood-style fantasy movies. "Interesting. I can see why the movies are grouped into different

clusters", Yuki said. she then saw that another strip brought in the comedy element. "This system gradually introduces diverse elements into the recommendation", Yuki said.

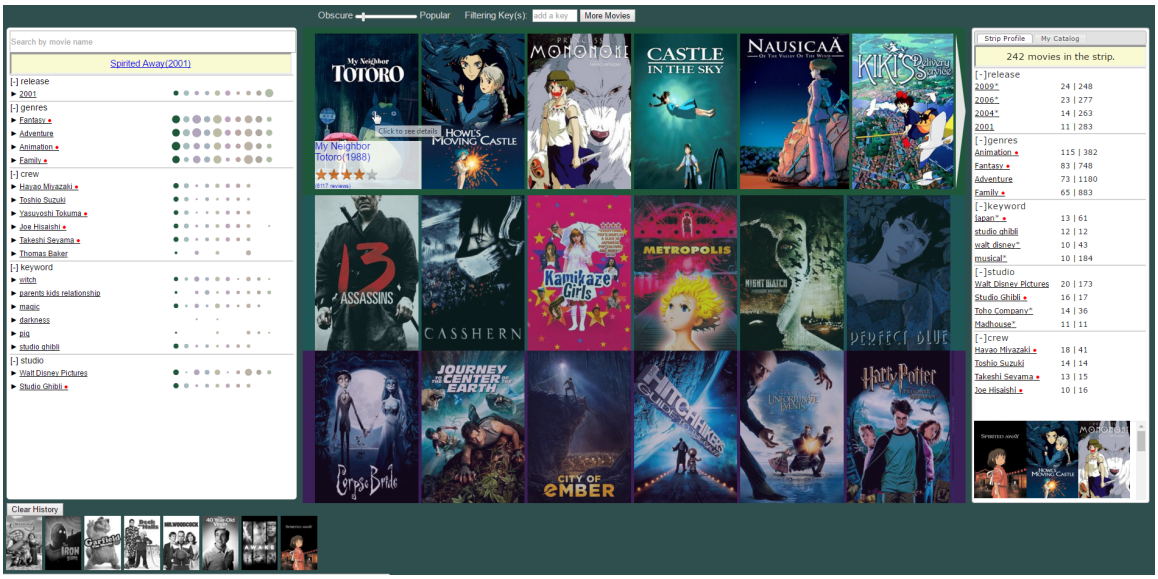


Figure 4.1: Interactive Movie Finder: the view presented to a user when she looks for movies similar to "Spirited Away (2001)". Left: Keywords of "Spirited Away (2001)"; Middle: Posters of the recommendations. Each strip shows top ranked movies in a cluster. The top strip has the focus; Right: Featured keywords of the focus strip. Bottom: The History Panel. The mouse is hovering over the poster of "My Neighbor Totoro (1988)". Its keywords are highlighted by red dots in the left and right panels.

Using the obscure-popular scale Yuki scrolled back to the top strip. She wanted to find a "Spirited Away (2001)" style movie which she hadn't watched. The right panel, which provides basic information about this cluster, showed that there were 242 movies in this cluster. The six leftmost ones were in the view port and scrolling the strip could bring out more movies. Yuki used a more efficient interaction: she adjusted the obscure-popular scale above the strips to the obscure side. Movies on the strips were immediately re-ordered and the less-watched movies were placed leftmost and became visible (see Figure 4.2). Yuki was happy to see several movies she never heard of.

Examining a movie of interest Yuki hovered the mouse over a poster and its title and rating were displayed. In the left panel, the keywords shared by the movie

and "Spirited Away (2001)" were highlighted by red dots. She found that this movie, "Tales from Earthsea (2006)", was also produced by Studio Ghibli and many crew members of "Spirited Away (2001)" also participated in its production. "It might be a movie I'd like to watch", she thought. She thus clicked the poster to pop out its detail window. The plot, trailer, and IMDb link to this movie were provided. After watching the trailer, Yuki clicked the button "add to my catalog" so that she could watch it later.

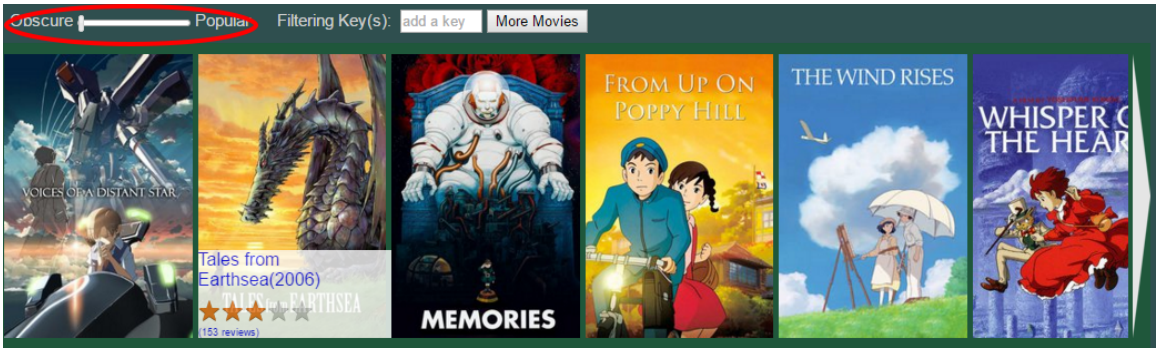


Figure 4.2: The same movie cluster as shown in the top strip of 4.1 is re-ordered to promote novelty.

Filtering and Reordering Yuki examined keywords of "Spirited Away (2001)" in the left panel. She appreciated its fantasy element, so she clicked "Fantasy" to use it as a filter. Only fantasy movies were left in the middle view. By clicking a dot in the left panel representing the keyword-cluster relationships, she directly moved the focus of the middle view to the cluster with the largest number of fantasy movies. From this strip, she collected several movies of interest. Yuki then focused on fantasy animation movies by adding "Animation" into the filter. By clicking a button beside "Animation", She sorted the clusters based on the number of animation movies contained in the clusters.

Progressive exploration Yuki browsed the strips, and soon noticed "Shrek 2 (2004)", a movie she liked. She wanted to find more movies like it, so she clicked the "Find more movies like this" button from its detail window. It became the example

movie and a new set of movie clusters were recommended.

Figure 4.3: The middle view of IMF with example movies of different styles.

Using the Serendipitous Map Yuki wanted to check if there were any fantasy animation movies out of her search range, so she clicked the "More Movies" button, which brought up the Serendipitous Map, where many significant movies not yet visited are displayed as posters (see Figure 4.4). "The Iron Giant (1999)" got her attention. She dragged it into the map and the map showed that she could access many movies not yet visited from it. She set it as an example movie and found many good surprises from the recommendations.

4.3 Interactive Movie Finder

IMF is a fully working, web-based prototype hosted at http://goo.gl/7U4WMx, with Python's Django as the server. I first describe its subspace clustering and example-based subspace recommendation, refinement, and ranking algorithms, and then present the visualizations and interactions.

4.3.1 Subspace Clustering

The data used in IMF for subspace clustering are 10 million movie ratings from 71,567 audiences (about 10,681 movies) on MovieLens [55]. Ratings earlier than the year 2006 are removed so that the rating standard is more consistent. IMF employs a topic model [59] to generate overlapping subspace clusters, while other subspace clustering algorithms that allow subspaces and clusters to overlap can also be used. Subspace clusters are extracted from the rating data with the following steps:

Step 1: A document is generated for each audience with movies they like. I assume an audience likes a movie if they gave it a rating higher than 3 out of 5. The IDs of such movies are placed into the audience's document. Each ID is considered a term in the topic model. I discarded documents (audiences) with less than 20 terms or more than 2,000 terms, since, according to Tang et al. [56], documents should not be too long or too short to improve the quality of topic models. The resulting corpora contains 71,078 documents (audiences) and 8,690 terms (movie IDs).

Step 2: The documents are sent to Mallet [59], a machine learning for language toolkit, for topic modeling. Topic modeling [60, 61] utilizes the fact that documents exhibits multiple topics (e.g. audiences have multiple tastes) and a document blends multiple topics with different proportions (e.g. an audience has a blend of tastes). A topic can be considered as a subspace cluster that ties together a subset of documents exhibiting the topic (e.g. a subset of audiences with the same tastes) and a subset of terms reflecting the topic (e.g. a subset of movies co-liked by the audiences).

Topic modeling requires the number of topics as the input. I heuristically set this number to be 100, so 100 topics are generated. Setting a good parameter is a challenge in topic modeling - if the number is larger than the actual topic number in the corpora, redundant topics are generated; while if the number is smaller than the actual number, there can be coarse topics that mix multiple actual topics. Since there is a lack of guidelines on how to set the number of topics in a topic model [56], I

propose an example-based approach to refine clusters and remove redundant clusters (see Sections 4.3.4 and 4.3.3). Thus, the heuristic topic number works well in IMF.

Step 3: A subspace cluster is constructed from each topic. Topic modeling generates overlapping subspace clusters since a term (movie) or a document (audience) may be associated with multiple topics with non-zero probabilities. The audiences associated with a topic constitute an audience community with similar tastes in movies, called a **fan group** in this project. The movies associated with the topic are the movies co-liked by the fan group, called a **movie cluster**. A person can belong to multiple fan groups since she/he likes multiple types of movies. A movie can be preferred by multiple fan groups since it has multiple latent characteristics.

The average rating for each movie is calculated. Attributes of the movies, such as year of release, crews, genres, keywords, studio, actors, posters, and trailers, are collected from The Movie Database (TMDB) [57]. In particular, the top five actors in the payroll of each movie are collected. All crew members, such as director, producer, writer, and music conductor, are collected. Each distinct term, such as an actor name or a genre name, is referred to as a keyword in this project. Keywords appearing less than 10 times in all movies are dropped from the dataset.

4.3.2 Calculating Similarities and Finding Similar Movies

The same pair of movies can have different similarities as viewed by different fan groups. For example, if two movies both have strong fantasy elements while one has a much stronger comedy element than the other, their similarity as viewed by fantasy movie fans should be much larger than it is for comedy movie fans. I capture this fact by calculating similarities among movies within the individual fan groups. The similarities are used to identify inner-cluster neighbors of movies.

The weights of different audiences vary in the same fan group (e.g. big fans vs. causal viewers); the same audience has varying weights in different fan groups (e.g. an audience watched a lot of fantasy movies and occasionally watched comedy movies).

The weights are considered in similarity calculations and the following example illustrates the reason: to evaluate movie similarities in a Sci-Fi movie cluster, the opinions of big Sci-Fi movie fans are more important than the opinions of casual viewers of Sci-Fi movies. I set the weight of an audience in a fan group as the probability of the audience documents associated with the topic assigned by the topic model. In particular, I define movie m's feature vector inside a cluster k as:

$$P_m^k = (\varphi_{1,m}^k, \varphi_{2,m}^k, ..., \varphi_{i,m}^k, ..., \varphi_{N,m}^k)^T, \tag{4.1}$$

in which $\varphi_{i,m}^k$ is audience i's weight in topic k if the audience has ever given a rating higher than 8 out of 10 to movie m, otherwise it is 0. N is the total number of audience members in this fan group. Following existing item-based collaborative filtering approaches, the similarity between movies j and h within topic k is calculated using improvised Jaccard similarity:

$$Sim_k(j,h) = \frac{\sum_m min(P_{j,m}^k, P_{h,m}^k)}{\sum_m max(P_{j,m}^k, P_{h,m}^k)}. \tag{4.2}$$

In IMF, I calculate the similarities off-line and record each movie's inner-cluster neighbors in each cluster during preprocessing.

4.3.3 Example-Based Cluster Refinement and Movie Ranking

Among inner-cluster neighbors, movies are ranked with the following considerations: (1) Movies closer to the example movie should be more visible and (2) A user may prefer to see popular or obscure movies first. Therefore, the inner-cluster neighbors are sorted by the following priority score in cluster k:

$$R_k(j,h) = Sim_k(j,h) + \alpha w_h^k, \tag{4.3}$$

in which j is the example movie and h is the movie to be ranked. w_h^k is the normalized weight of movie h in cluster k, reflecting its popularity in the cluster. I set the weight of a movie in a cluster as the probability of the movie term associated with the topic assigned by the topic model. Its value ranges vary a lot among the clusters. Without normalization, a single α is not able to adjust the ranking in multiple clusters at the same time. α is a parameter returned from the user interface. It is ranged from -1 to 1. When α is positive/negative, popular/obscure movies in the fan group get higher priority scores. When $\alpha = 0$, the popularities of the movies are not considered in the ranking. Users can interactively change α through a popular-obscure scale in the interface to bring out popular or obscure movies first. The non-neighbor movies are placed after the inner-cluster neighbors, ranked by their weights in the topic.

4.3.4 Cluster Recommendation and Redundancy Reduction

Among the large number of clusters, which clusters should be presented to users? An intuitive solution is to select the top (IMF sets it to be 10% of the total number of clusters) clusters with the largest numbers of inner-cluster neighbors with the example movie, since they are the most relevant clusters to the example movie. However, there are two problems. First, clusters with significant overlaps are not desired in visual explorations. They increase cognitive load and reduce screen usage efficiency without introducing new information. Second, displaying the same movie in the most visible locations of multiple clusters should also be avoided in order to use the screen more efficiently.

To tackle the problems, a redundancy reduction approach is designed. First, IMF sets a rule that a movie is displayed as the example movie's inner-cluster neighbor at most once in a view. If a movie is an inner-cluster neighbor of the example movie in multiple selected clusters, it is removed from the inner-cluster neighbor lists except in the cluster where it has the highest priority score. In this way, a movie is displayed in a high priority position in the visualization at most once.

Second, if the number of inner-cluster neighbors of the example movie in a selected cluster changes significantly after the first step, it overlaps considerably with other selected clusters and thus can be removed from the selected cluster list. IMF removes the clusters with less than 50 inner-cluster neighbors after the first step, since top clusters typically have much more inner-cluster neighbors than 50 before the first step.

4.3.5 Cluster Ordering

The selected clusters are sent to the visualization in an ordered list, which decides their visibility in the view. In addition, clusters adjacent in the list will be adjacent to each other in the visualization. By placing the most relevant cluster to the example movie in the most visible position, users can see it at the first glance, and thus gain trust in the recommendation. I define the cluster where the example movie has the biggest weight as the most relevant cluster, and rank it first in the list.

For the rest of the clusters, I considered two options for the ordering. The first option was to diversify adjacent clusters so that users can quickly access movies with different tastes. However, this option was not appreciated in our preliminary user tests, since the users could not make sense of the order and thus felt confused. The second option was to sort the clusters by their distances to the top cluster from small to large. A simple algorithm was used for the sorting: count the number of movies shared in the top 30 inner-cluster neighbors (before redundancy reduction) by a cluster and the top cluster. The larger the number, the closer the cluster is to the top cluster. The layout generated with this method was appreciated by the users. They commented that the themes gradually changed with new variance introduced and that the layout helped them identify the diversity. Therefore, this option is used in IMF.

4.3.6 Keyword-Cluster Relationship Discovery

Keywords are used in the visualization to help users learn the clusters and conduct operations such as filtering and cluster reordering. I define that a keyword is featured in a cluster if it has a heavier weight in this cluster than in the global space. The weight of a keyword in a cluster is calculated by dividing the sum of the weights of movies relevant to this keyword in the cluster by the sum of the weights of all movies in the topic. The weight of a keyword in the global space is the percentage of movies relevant to the keyword in all movies. IMF groups keywords by their categories. Within each category, the top four featured keywords are displayed in the strip summary.

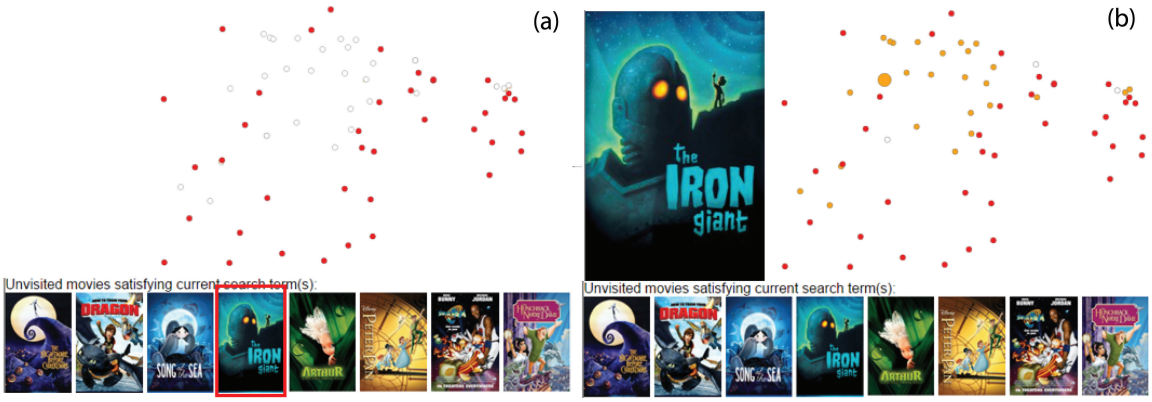


Figure 4.4: The Serendipitous Map: The scatter plot shows significant fantasy animation movies. The posters in the bottom show representative unvisited movies. The large golden dot represents the selected movie, "The Iron Giant (1999)", while other golden dots represent unvisited movies that will be recommended if "The Iron Giant (1999)" is set as an example movie. White dots represent other unvisited movies and red dots indicate movies that have been visited.

4.3.7 Creating the Serendipitous Map

The Serendipitous Map displays a 2D projection of movies in the dataset (see Figure 4.4). It allows users to spot explored and unexplored regions in the dataset and directly access representative movies. To avoid clutter and reduce the cognitive load of users, only the most important movies are displayed in the map, in the hopes that other movies can be reached using them as example movies in the exploration.

The top 1,000 movies with the highest salience scores are used. Chuang et la.[62] proposed salience score to select salient terms in one or more topics. Please refer to [62] for the algorithm. The selected movies are projected onto a 2D space using spectral embedding[63], which preserves the local distances of movies in the high dimensional space. To create the affinity matrix, for each pair of movies, I count how many times each movie appears as the other's neighbor. Therefore, movies close to each other in more clusters will be closer to each other in the map.

4.3.8 Visualizations and Interactions

A typical visual exploration pipeline users may use in IMF is: (1) setting a favorite movie as an example; (2) browsing clusters recommended by IMF; (3) examining clusters of interest; (4) setting a movie of interest as a new example; (5) using the serendipitous map to explore unfamiliar regions in the dataset. During the process, users can examine movies of interest at any time and save them for future watching, jump from one step to any other steps, or go back to a previous view through the history management interaction. I briefly introduce the visualizations and interactions in this section.

Setting example movies Clicking any posters in IMF opens up a movie detail window allowing users to set the movie as the example movie. Users can also search a movie using keywords in its title through a search box to pop out the movie detail window.

Browsing clusters As discussed in Sections 4.3.4 and 4.3.5, a set of clusters are returned by IMF in an ordered list after an example movie is set. They are displayed in the middle of the interface (see Figure 4.1). The top three clusters are displayed as three horizontal strips in the view, and other clusters can be accessed using the down key in the keyboard to scroll the strips vertically. There is always one strip brighter than the other strips - the focus cluster that users can examine in detail. Uses can use up/down keys to switch focus among the clusters.

The movies in a cluster are also returned in an ordered list (refer to Section 4.3.3). Their posters are displayed in the strip from left to right with the same order. Users can use the left/right keys to scroll the brighter strip to examine more movies in the focus cluster.

Figure 4.3 (1)-(3) shows several views with different example movies. Readers who are movie fans can tell that the examples clearly demonstrate the effectiveness of IMF in generating accurate yet diverse recommendations to users.

Accessing and reordering clusters with the dot matrix There is a row of dots following each keyword in the left panel (Figure 4.1, on the left). Each dot represents a cluster and the dots have the same order as the order of the clusters in the strips. The size of a dot represents the number of movies in the cluster associated with the keyword. If a user finds a keyword of interest, she/he can click a big dot to bring the focus to a cluster with lots of movies with the keyword. By clicking the sort button before a keyword, users can sort the clusters by the number of movies associated with the keyword in them in descending order.

Filtering Uses can click any keyword visible in the interface or type a keyword in an input box to set it as a filter. Movies not associated with the keyword will be removed from the clusters. Clicking the keyword again will remove it from the filter. Multiple keywords can be used in a filter at the same time and only movies with all those keywords will be remained in the view.

Movie reordering Moving the obscure-popular scale slider changes the order of movies in the clusters (popular movies first or obscure movies first). Please refer to Section 4.3.3 for more details.

Examining a cluster When the tag of the right panel is "strip summary", the right panel presents a summary of the focus cluster (see Figure 4.1 on the right). The featured keywords of the cluster are displayed. The numbers after a keyword are the number of movies with the feature in the cluster, and the total number of movies with

the feature in the whole dataset, respectively. A star highlights a keyword not in the example movie, so that a user can notice the difference. The most popular movies in the cluster are displayed at the bottom of the panel as posters so that users can quickly access them.

Examining a movie A user can hover the mouse over a poster in the middle view to highlight it. Its title and rating are overlaid on the poster. The keywords shared by the movie and the example movie/the focus cluster are highlighted by red dots in the left/right panels, so users can effectively find their shared features.

The movie detail window triggered by clicking a poster presents more details of the movie. From this window, users can set a movie as the example movie, watch its trailer, access websites where the movie can be watched directly, add it to a catalog and watch it later, or remove the movie from all clusters.

Serendipitous Map Clicking the "more movies" button will bring up the Serendipitous Map (see Figure 4.4). The map presents a scatterplot overview of the major movies in the dataset, where co-liked movies are close to each other (see Section 4.3.7 for the major movie selection approach and the projection algorithm). A movie is considered as "unvisited" if none of the clusters it belongs to have been recommended to users in their previous explorations. Otherwise it is a "visited" movie. Unvisited/visited movies are displayed as white/red dots in the scatterplot. Users can find unvisited movies by looking for white dots. A listed of unvisited major movies are displayed as posters at the bottom of the map for quick access.

Clicking a dot or dragging-and-dropping a poster into the scotterplot will make the movie appear on the top left of the scatterplot as a poster. Once a movie is displayed on the top left, the scatterplot provides a preview of the recommendations with this movie as the example - unvisited movies in the recommendations are colored in golden. If users see a lot of golden dots, they know that this movie is a good choice for the next example movie for good surprises.

History management At the bottom of the interface, there is the History Panel. All the example movies a user has chosen are displayed here as posters. The poster of the current example movie is colorful while other posters are in gray. Users can click a gray poster to make the movie an example again. Users can also remove a poster from the history or clear the whole history.

Insight management From the movie detail window, users can save a movie of interest to a catalog and watch it later. Users can click the "my catalog" tag on the right panel to see posters of the movies in the catalog. Users can also drag-and-drop a poster to the panel to add the movie into the catalog. Users can clear the catalog or remove a movie from it at any time.

4.4 Accessibility and Diversity Evaluation

In two experiments, I compared IMF with a baseline approach (BA), where an item-based collaborative recommendation algorithm [58] is used. Our hypotheses were that IMF has:

Better Accessibility: In average, movies can be accessed with less efforts from the example movies in IMF than in BA;

More Diverse Recommendations: The movies users explored have a higher profile dissimilarity in IMF than in BA.

Accessibility is important in recommendation since a system with better accessibility can allow users to find interesting items more efficiently, even if she/he starts from a poor example. Diversity is desired in recommendation systems.

4.4.1 The Approaches

BA recommends the nearest neighbors of the example movies in the user rating space to users. According to Arsan et al. [64], the distance between two movies is measured using the Jaccard distance[65] of their fans vectors in the global space. Similar to IMF, a BA user can select a movie in the visited neighborhood and set it

as a new example movie. In addition, I assume that a user would never be patient enough to go through a neighborhood of more than 30 items. Therefore, an item can have 30 neighbors at most. To make a fair comparison, I only keep the first 30 or less items in each cluster in IMF.

A user's efforts is measured by counting clicks. In BA, a user may use a click to access and examine the next movie in the current neighborhood, or to set the movie under examination as a new example movie and bring in its neighborhood. In IMF, a user may use a click to access and examine the next movie in the current cluster, to access the first movie in the next cluster in the cluster list, or to set the movie under examination as a new example movie and bring in its relevant clusters.

I created a directed graph in which the nodes are movies and the edges link a movie and its neighbors, for BA and IMF, respectively. In the BA graph, the length of the edge that links a movie and its Nth neighbor is N clicks (1 click to set the example and N-1 clicks to access the Nth neighbor). In the IMF graph, the length of the edge linking an item and its Nth member in its Mth topic is N+M clicks. Therefore, the efforts from Movie A to Movie B can be calculated as the length of the shortest path from node A to node B. Both graphs are connected, which means that there is always a shortest path from one node to another.

4.4.1.1 Accessibility Test

I measured the accessibility of a movie using 1 over the average of the shortest path lengths to it from all other movies. The result is shown in Figure 4.5. In each chart, the x axis represents the ratio of each movie's accessibility in IMF to that in BA. A value larger than 1 means that a movie is more accessible in IMF than in BA. Movies fall into the same ratio bin are stacked up. The color represents number of ratings 4 or higher each movie has, indicating how popular the movie is. Red shows the movie has more ratings and green shows the movie has less ratings.

Each chart is generated under a different popular-obscure setting. In Figure 4.5(a),

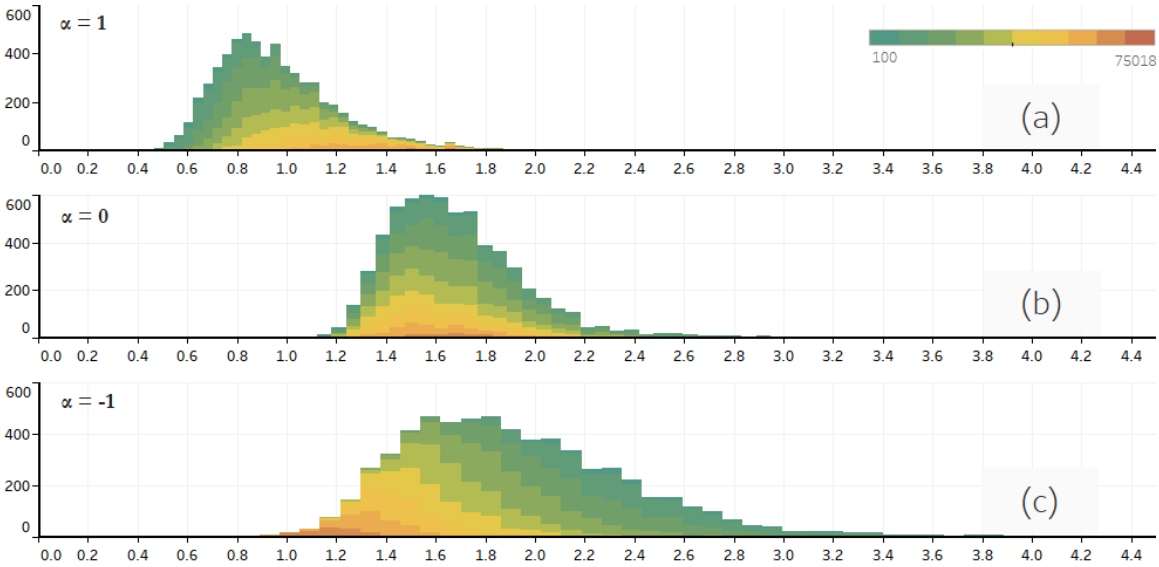


Figure 4.5: The accessibility test results. (a). The results when the rankings of popular movies are boosted. (b). The results without considering movie popularity in ranking movies in a cluster. (c). The results when the rankings of obscure movies are boosted.

the popular-obscure slider is set to the "popular" end (e.g. $\alpha = 1$), meaning that popular movies will have higher ranks in the recommendation. In Figure 4.5(b), the slider is set in the middle (e.g. $\alpha = 0$). In Figure 4.5(c), the slider is set to the "obscure" end (e.g. $\alpha = -1$) so that movies with less ratings will have higher priorities.

From these figures we can see that in the "popular" setting, IMF boosts the accessibilities of the popular movies while reducing the accessibilities of obscure movies. In the middle setting, the accessibilities of most movies are boosted. While in the "obscure" setting, IMF not only significantly boosts the accessibilities of obscure movies, but also boosts the accessibilities of most popular movies, although not as significant as in the middle setting. According to the experiment results, I set the default position of the popular-obscure slider to the middle of the bar (e.g. $\alpha = 0$) in IMF.

4.4.1.2 Diversity Test

This test evaluated whether IMF can provide users more diverse recommendations. For each movie, I collected the first N movies that can be reached from it, and

calculated the recommendation diversity as the diversity among these movies. In this test, it was not allowed to change example movies and there were no limitations to the number of neighbors could be accessed from the example movies in both IMF and BA.

The ground truth of diversity is collected form movie profiles. I measured the difference between two movies using the Jaccard dissimilarity of their profiles. Keywords that appeared in less than 1% of all movies were discarded since they were uncommon. Since the profiles of some movies were not available in The Movie Database (TMDB) [57], not all pair-wise difference values could be calculated. Thus I took the highest 10 difference values and calculated the recommendation diversity as the average of those 10 values. The result is presented in Figure 4.6.

In Figure 4.6(a), the y axis is the ratio of the average recommendation diversity starting from all movies in IMF to that from BA. The x axis shows the number of top recommended movies considered. From this figure we can see that the ratio is always bigger than 1, which means in average, exploration in IMF leads to higher diversity. The advantage of IMF gets bigger as a user accesses more recommendations. I then considered the individual movies. In Figure 4.6(b), y axis shows the percentage of movies that can trigger recommendations with better diversity in IMF. A percentage above 50% means that users can have a higher chance to get more diverse recommendations if they randomly pick an example movie. As we can see from Figure 4.6(b), IMF is better than BA since the percentages are all above 50%. The advantage of IMF gets bigger as a user accesses more recommendations. However, IMF can't guarantee that every example movie can lead to more diverse recommendations. How to improve it remains in the future work.

4.5 Preliminary User Study

I evaluated IMF with a k-12 math teacher, a CS master student, and a senior CS PhD student majoring in visualization. All the subjects claimed that they had

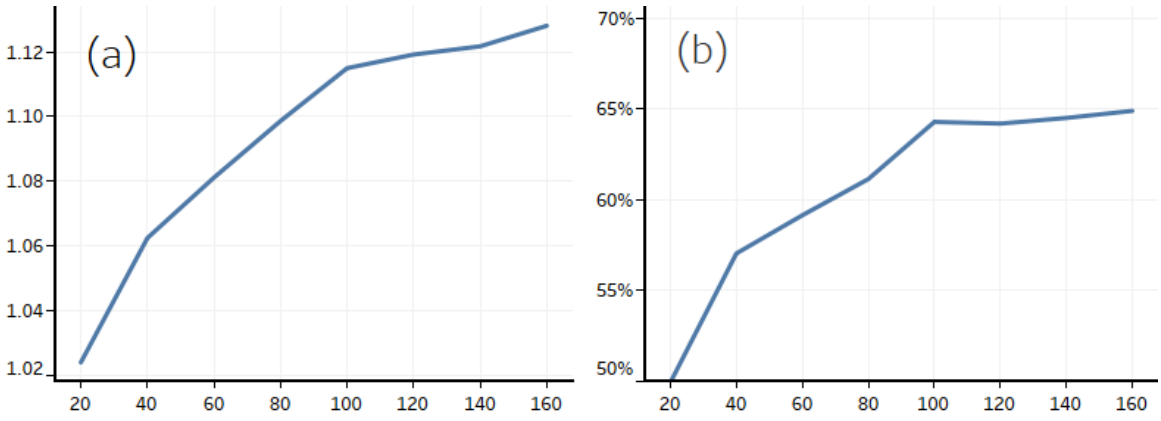


Figure 4.6: The diversity test results. (a). The average diversity gain of IMF. (b). The percentage of example movies with diversity gains in IMF.

watched a large number of movies. I intentionally invited them so that I could collect reliable opinions about the clustering and recommendation quality in a short amount of time. The other purposes of the studies were to observe how IMF was used and to collect user feedback. The studies were conducted one-on-one in a think-aloud, open-ended manner. The instructor and the subject sat together in front of a 22 inch computer monitor with the keyboard and the mouse controlled by the subject. The studies with the subjects lasted 120 minutes, 50 minutes, and 25 minutes, respectively. The time variance was only related to the available time the subjects could contribute to the evaluation.

There were no formal training sessions in these studies. All subjects immediately started to use the system since the interactions were very simple. The instructor provided hints during the studies to remind the subjects about available functions. Whenever the subjects found something interesting or had suggestions, they talked with the instructor immediately. At the end of each study, the instructor asked the subject to compare IMF with other recommendation systems they were familiar with and provide comments about IMF. All the studies were audio-recorded.

Each subject conducted many use cases. I select three cases different from our previous examples and discuss them in Section 4.5.1. They illustrate how the subjects

used IMF in different ways to satisfy different information needs. In Section 4.5.2, feedback from the subjects is presented.

4.5.1 Use Cases

4.5.1.1 Navigation from a Rough Start

A subject was looking for movies with awesome fighting scenes. However, the only relevant movie he could immediately think of was "Kung Fu Panda (2008)". So, he searched for it and set it as the example movie. The top cluster was filled with animation movies but the other clusters were not. From the left panel, he found the key "Jackie Chan" in the cast list. He reordered the clusters by how many Jackie Chan movies they had. He immediately found that the top cluster contained many action movies, as indicated by the cluster summary in the right panel. Next, he scanned the posters in the right panel. They were the most popular movies in that cluster. He picked "The Bourne Ultimatum (2007)" as the new example movie. With this example movie, he found lots of action movies, along with all three other Bourne movies in the view.

Discussion: This case demonstrated how IMF narrows down a search from a rough start. The diversity of the clusters recommended allowed the subject to have a broad view even when he started from an example movie only loosely related to his interests. The capability of progressive exploration by selecting new example movies enabled him to quickly drive the search to his real interest.

4.5.1.2 Getting More Tarantino Movies

A subject was interested in Quentin Tarantino's movies. But he forgot how to spell the name. So, he set "Pulp Fiction (1994)", a famous Tarantino movie, as the example movie. The initial view is shown in Figure 4.3 (2). He saw other works by Quentin Tarantino in the first cluster, such as "Reservoir Dogs (1992)", along with other similar movies such as "Fight Club (1999)" and "Se7en (1995)". He was impressed and said it

was the best results he had ever seen among all the recommendation systems he had used. The subject also showed interest in the third cluster, in which more comedies were displayed. He pointed to "Twelve Monkeys (1995)" and said, "It has something in common (with Pulp Fiction), but it is actually a comedy movie. I don't know why it is here but I am a big fan of this movie." He then set this cluster as the focus and continued his exploration in this cluster. He found many more interesting movies there.

Discussion: Other subjects also had similar positive comments about the cluster/recommendation quality. Note that IMF only used user ratings in clustering and the clustering algorithm was not aware of the director information. The comments indicated the effectiveness of our example-based cluster refinement algorithm, since the raw clusters from topic modeling do not have such a high quality.

I note that the subject at first was looking for Tarantino's movies and got very good recommendations. The subject then shifted his interest after seeing a movie from a different category, which is desired in recommendation systems (the diversity works!). When the subject changed his interest, IMF allowed him to focus on the interesting theme easily.

4.5.1.3 A Good Surprise

A subject was looking for crime movies with "Chinatown (1974)" as the example movie. He agreed with the recommendation results as they contain many classical crime movies. However, he was not interested in them because he had watched all of them. He clicked the "More Movies" button to fine other crime movies he "left out". He tried his luck and randomly clicked a white dot in the unexplored area and found it was "The Shawshank Redemption (1994)". This movie was different from what he was searching, but something he would definitely watch. He was impressed by this function because it helped him directly access the representative movies and found a good surprise. Thus he kept clicking other white dots in the scatter plot to find more

good surprises. He was surprised that he could find movies he was interested in from every corner of the scatter plot.

Discussion: The Serendipitous Map supports discovery for surprises because it makes the representative movies in the dataset accessible by a single click. Since the map also recorded the exploration history of the subject, he was able to control how "surprised" he wanted by clicking an unexplored movie closer or further away from the explored movies.

4.5.2 User Feedback

The subjects stated that IMF provides good recommendations and they liked the interactions. Below are their comments comparing IMF with existing commercial recommendation systems:

"I've used movie finders, but what happened is usually they give a selective movie list. For example, when I search Pulp Fiction, the only recommendation I can get are the big Tarantino movies, because they are similar and all very famous. So here, using the obscure bar, I can quickly get the similar movies that I haven't seen. I haven't seen other recommendation systems that have this wide of a range. It makes me want to explore more."

"The views are broader, which people will like. Amazon and Netflix make it too strict. I look at their results and find I have watched all of them."

"I find the new topic-based movie recommender offers me a much broader and richer set of choices than the IMDb recommender system. I particularly like the ability to look for more obscure movies to help me find the hidden gem."

Following this statement, I asked the subjects what made them confident about the obscure recommendation. Two of them said because they were mixed with movies clearly relevant to the example movie. One subject actually pulled the obscure-popular bar to the popular side every first time he dumped into a cluster so that he could see the movies he knew and learn what the movie cluster was about. Then he

pulled it back to the obscure side to find rare movies. Those comments showed that the example-based cluster refinement and the interactions that allow users to flexibly sort movies within a cluster helped the subjects understand unfamiliar movies based on the knowledge they already gained in a dataset.

The subjects also commented on other features of IMF:

About understanding latent characteristics of movies "I found 'Crouching Tiger, Hidden Dragon' from a romance movie strip. I thought it was about martial arts. Now I know there is a romance element in it and I want to watch it again." "I know I like these movies, but I didn't know why. Putting them together makes me think about the common elements in them that make me like them."

About cluster browsing "The top row has the most accurate recommendations. The other clusters gradually add variance in, which is good."

About history and insight management "I am very glad with the history management. It allows me to go back to a previous view easily.""This (my catalog) is a good function."

4.6 Related Work

Recommendation Systems Recommendation is a significant area in computer science, where researchers study how to recommend the most suitable items to particular users by predicting their potential interest [66]. It has many applications, such as movie search, online shopping, and vacation recommendation [67]. A large number of recommendation approaches have been proposed. Content-based approaches [68] attempt to recommend items that are similar to items a user liked in the past. The similarity is calculated based on content such as profiles of the items. Collaborative Filtering-based (CF) approaches [69] make recommendations based on the interactions (e.g., ratings) between users and items, and thus could still work when contents are not available. User-based CF approaches calculate similarities among users based on their common preferences. Items are recommended to a target user if they are

preferred by users similar to him or her. Item-based CF approaches [58, 70] compute the similarities among items based on the common interactions by users. Those items that are similar to a target user's historically consumed items are recommended to the user.

Our approach falls into the category of item-based CF approaches, since I cluster movies based on user ratings. However, I do not collect user consumption histories or user profiles. Rather, users actively improve the recommendation results through interactions. Meanwhile, IMF allows users to filter the recommendation results using keywords from an example movie, which provides the benefit of content-based recommendation.

A few efforts have been made to enhance recommendation systems with visualization. Chen et al. [71] project recommended web services into a 2D map which shows their underlying structure with very limited interactivity. They comment that users are less likely to trust recommendation results from a black box without knowing the reason behind them. PeerChooser [72] is a movie recommendation system that visually explains the underlying CF process to facilitate exploration. It provides information about the user community upon which the recommendation is made and allows users to interactively change it. An interactive node-link diagram is used in it. SFViz [73] recommends friends to a user based on his/her social information. A radial, space-filling tree visualization [74] and a circular graph visualization are used to help users understand their friendship patterns and the categories of interest of their friends. The grid layout used in our approach is much simpler than the visualizations used in those approaches. However, the rich set of interactions and the tight integration with automated algorithms make IMF a powerful tool that promotes accurate, novel, diverse, and serendipitous discoveries.

Topic Modeling Topic models, such as latent Dirichlet allocation (LDA) [60] and Latent Semantic Analysis (LSA) [61], are statistical models of document collections

that utilize the fact that documents exhibit multiple topics and an article blends multiple topics with different proportions [75]. Since user-item pairs can be mapped to document-term pairs in a document corpus, topic models have been used in recommendation systems, where each topic infers a user community with their referred item group. Zhang et al. [76] merge top items from various models, including a user-based LDA model and an item-based LDA model, to enhance serendipity of recommendation results. They also identify items on the edge of topics and avoid heavy concentration items to promote novelty of recommendation. Ziegler et al. [77] employ topic modeling and evaluate and re-rank recommendation results by their community differences. As with most recommendation systems, these works try to create a single list with a limited length, where accuracy, novelty, diversity, and serendipity need to be balanced.

Many document visualization systems, such as the Topic Browser [78], Termite [62], and Serendip [79], allow users to interactively explore topic modeling results. For example, Chaney and Blei [80] display each topic as a card containing the top N words in the topic. User can interactively retrieve relevant topics according to a keyword or a document, or retrieve keywords and documents related to a topic. The Topic Browser [78] presents topics as tag clouds of the top keywords and provides extra information such as basic statistical analysis results. However, these approaches use the "overview first, zoom, detail on demand" paradigm [1] and are not suitable for a recommendation application.

Subspace Clustering-based visualization Clustering-based visualization approaches can benefit browsing and information gathering tasks when a large amount of information needs to be accessed [81]. For high dimensional data, subspace clustering is used since global clustering loses their effectiveness. For example, ClustNails [82] displays a matrix of thumbnails of the clusters as an overview and uses an extended heat map to reveal the relationships among the data points, dimensions, and

clusters. Ferdosi et al. [83] propose a method to rank subspaces and provide a graphical interface where users can interactively explore the ranked subspaces. Our approach is different in that it is customized for a recommendation application - subspace clusters visualized, data points examined, and redundancy reduction are all relevant to an example data point users select.

A few works have been done on visual explorations of overlapping biclusters in the application area of bioinformatics. For example, BiSet [84] uses semantic edge bundling to help users explore coordinated relationships revealed by the overlaps among biclusters. Rather than using complex visualization techniques such as edge bundling, our approach uses only simple visualization techniques so it can be used by non-visualization experts.

4.7 Project Conclusion and Future Work

In this project, I propose IMF, a fully working interactive movie recommendation system. It tightly integrates interactive visualizations with automated subspace clustering and cluster recommendation, refinement, and ranking algorithms. It allows users to observe the varying characteristics in movies and find movies of interest from a large number of diverse recommendations effectively and efficiently. It also promotes novelty and serendipitous discoveries through interactive visualizations powered by automatic algorithms. A set of evaluations have been conducted to illustrate its effectiveness and efficiency.

In the future, I would like to apply the proposed approach to not only recommendation applications, but also other applications where insights need to be extracted from data with complex relationships, such as streaming data visualization and dynamic graph visualization.

4.8 Mantra Reflection in This Project

Unlike the other two projects, the interactive movie finder (IMF) is focused more on navigating users to meet diverse objects and less on analysis. Egos are not analysis targets but movies that display users' implicit interests. The same as item-based recommendation systems, movies similar to an ego are brought to a user's attention. With no further knowledge on why the ego movie is chosen, the IMF relies on interactions to adapt users' drifting interests and deliberately increase diversity on demand.

First, a user can change the definition of similarity. High dimensional objects such as movies can be associated in different ways. In IMF, the similarities among movies in the same movie strip is calculated within a subspace. By selecting a movie strip, a user increases his/her chance to meet movies similar to the ego movie with regard to attributes featured in the subspace. In this way, the exploration direction is piloted. Second, a user can customize the browsing order within each movie strip through filtering and weighted sorting. For example, with the obscure slider a user can require the system to list more/less popular movies first. Third, a user can select a movie discovered in browsing as the new ego, if it better matches the user's intent. It further extends the potential visiting area in the database and thus increases diversity. A user can also retrieve an old exploration state and restart the exploration from there using the history tool.

The two side panels explain the relation between the ego movie and the movie strips. They help a user makes sense of the recommendation results and then makes decisions. The movie panel (left) shows how important each movie profile is in relation to the movie strips, and the movie strip panel (right) shows the characteristic movie profiles and movies in the strip.

Similar to the spot-tracking lens, an overview is added to show the exploration coverage and provide access points to unvisited objects. For these two proposes the

overview can only contain a sample rather than the whole dataset. In IMF, the overview contains the characteristic movies from all subspaces.

CHAPTER 5: EventPool

This section present the EventPool system. It is designed to answer low level comparative questions, such as finding extremes, identifying data cases satisfying certain conditions, and comparing between objects, on a time series datasets. It is challenge because the answers evolve over time and carry rich time-oriented insights that are important to analysts. Traditional time series visualization methods, such as line graphs and the horizon graph, do not support such activities well due to the clutter problem and the lack of efficient interactions for task conduction and result inspection. EventPool supports dynamic queries for visually conducting a variety of low level analytical activities on time series datasets. It uses semantic encoding to generate pre-attentive, uncluttered views where users focus on exploring the query answers. With a rich set of interactions and coordinated views, users can discover a wide variety of insights effectively and efficiently. We present three use scenarios and two user studies to illustrate the usefulness, effectiveness, and efficiency of EventPool.

5.1 Introduction

An ever-larger body of Time Series Datasets (TSDs) are generated nowadays from social studies, financial activities, scientific measurements, and other applications. When exploring TSDs, analysts often ask "which and when" questions and look for insightful patterns from the answers for hypothesis generation and evaluation. For example, with the World Life Expectancy (WLE) dataset (200 countries, 1900-2012) [25], analysts could ask: "which countries had longer Life Expectancies (LEs) than the United States (US)? and when?" (Q1) and "which countries were among the top 10 in the world with regard to LE? and when?" (Q2). A variety of patterns in

the answers can be interesting to the analysts, which include but are not limited to: patterns about ranking, such as countries staying in the top 10 for the longest/shortest amount of time and countries that joined the top 10 earliest or latest; patterns about temporal trends, such as whether the US was surpassed by more and more countries with regard to LE; and patterns about groups, such as continents with a lot of top 10 members and continents with more and more countries entering the top 10 as time goes on. Analysts are often unclear about which types of questions and which thresholds in the questions will bring them interesting insights. They are also often unaware of patterns of interest before they see them. Therefore, automated analyses do not work well and exploratory visualizations are important to the analysts.

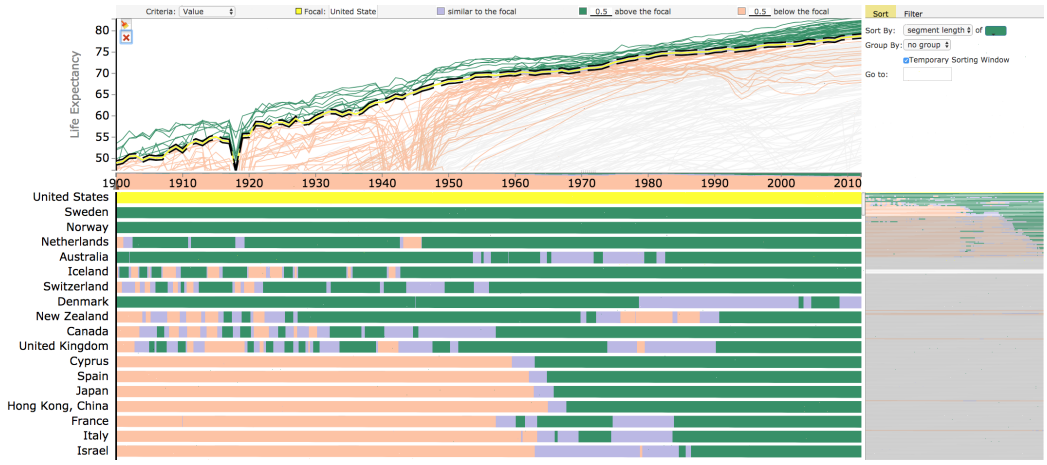


Figure 5.1: A screen shot of EventPool that answers the question "which countries had longer life expectancy (LE) than the US and when". The time periods when a country had a LE 0.5 year longer than the US are colored in green. When the LE was 0.5 year shorter than the US, the line is colored in orange. The line is purple when the LE is within 0.5 year of the US. Countries that had never had a close LE to the US are grayed out. The LE of the US are highlighted in yellow in the line graph. The small timelines on the right are an overview of the timelines on the left.

Existing time series visualization approaches, such as line graphs [85] and juxtaposed visualizations [86, 87], do not provide sufficient interactions and effective visual metaphors to help analysts investigate the "which and when" questions. For example, on the top of Figure 5.1, there is a line graph visually representing the answers to Q1. In this figure, a times series is colored in green in the time ranges when the

country had a LE 0.5 year longer than the US. It can be seen that the green lines are cluttered in a narrow region on the top of the line graph. It is difficult for users to find patterns without extensive human efforts. Time series do not overlap in juxtaposed visualizations, such as horizon graphs [86] and the line graph explorer [87]. However, research efforts on them are focused on how to use colors and positions to loyally represent the original data values. Very limited supports to answering the "which and when" questions are provided.

In this project, I propose EventPool, a new visualization system for exploratory analysis on univariate time series datasets. It helps users interactively construct a variety of "which and when" questions and visually discover insights from the answers. The "which and when" questions supported by EventPool include: **Extreme:** Questions about extremes, namely top/bottom N data cases with respect to an original or derived attribute, at each moment. An example question is Q2. **Condition:** Questions for data cases satisfying given conditions with regard to their original or derived values, at each moment. Example questions are: Which counties have LEs between 50 and 70? and When? Which countries' LEs decreased by 5 within one year? and when? **Comparison:** Questions for comparing a data case of interest with other data cases on their original or derived values, at each moment. An example question is Q1.

As pointed out by Amar et al. [88], finding extremes and data cases satisfying given conditions are two significant categories of low level analytical activities, and comparisons are an essential analytical activity at an even lower level. When TSDs are involved, "when" is always a critical consideration. Therefore, I argue that investigating the aforementioned "which and when" questions are essential analytical activities when analyzing time series datasets.

EventPool supports users in constructing queries and inspecting answers for the aforementioned "which and when" questions. Our approach consists of the following

steps:

Dynamic queries: A rich set of dynamic query functions are provided to users so that they can interactively "ask" questions and adjust thresholds with instant visual feedback. From the user's perspective, the queries are applied to the whole time range. However, readers can understand this step as constructing queries and gathering answers for the "which" part of a question at each time step. For example, to answer Q1, the following query is conducted at each year K: retrieve countries whose LEs are larger than the LE of the US in year K. The queries can be about values, ranking, changes, variance, or other derived values of interest. Queries at different time steps can have constant thresholds throughout all the time steps, such as "top 10" for Q2. They can also have varying thresholds that are functions of time, such as "LE of the US" for Q1.

Semantic encoding: Based on the query results, qualitative values are assigned to each data case at each time step. They provide question-related semantics to the data cases. For example, a qualitative value "longer LE than the US" will be assigned to a country in the years when it had a longer LE than the US. In the later visualization step, the time series are visually encoded by their semantics and other details are removed in several views. By suppressing details other than query answers through semantic encoding, EventPool allows users to focus on the answers and examine them in a more effective and efficient way with less clutter.

Comprehensible inspection: EventPool assembles query answers at individual time steps into views for the whole time span and provides a rich set of interactions to help users discover insights from them. In particular, timelines are used to display query answers where colors encode semantics of the time series. They are preattentive and can be browsed without clutter. A coordinated line graph, sorting, filtering, detail + overview, and grouping are provided to help users discover a wide variety of patterns.

I developed a fully working prototype of EventPool. It is a web-based system (http://bit.ly/2ok7W3V) developed in Javascript. Figures 5.1 and 5.2 show two screenshots of EventPool, answering Q1 and Q2 about the WLE dataset in green, respectively. The timelines on the bottom (left: a detail view; right: an overview) present the answers country by country. From Figure 5.1, it can be seen that the US is surpassed by more and more countries in recent years. From Figure 5.2, I can clearly see the consistent top 10 countries such as Sweden and Switzerland, the short time visitors such as Ukraine and Germany, and the newcomers like Israel and Singapore. Users can interactively changes the questions the visualization answers through a variety of dynamic queries for exploratory analysis.

The main contributions of this project include: (1) I identify a set of "which and when" questions, which are important low level analysis tasks that are largely ignored and hard to be conducted in existing time series visualization systems; (2) I propose EventPool, a new time series visualization system. It employs a dynamic query + semantic encoding + comprehensible inspection approach to answer those "which and when" questions. A rich set of interactions are provided in EventPool, which not only support users in effectively discovering insights from a given task, but also allow users to conduct exploratory analysis by inspiring new tasks and hypotheses; (3) I demonstrate the effectiveness and efficiency of EventPool in supporting the target tasks and exploratory analysis through three use scenarios and two user studies.

5.2 Related Work

Line graphs [85], where concurrent time series are displayed as lines overlaid in the same display space, become cluttered and comparing and following individual lines over a wide time span becomes difficult when there are more than dozens of time series [89]. Line graph variations, such as 3D line graphs [90] and the braided graph [89], also become unintuitive and misleading. Dynamic queries [91] have been provided in line graphs [92, 93]. TimeSearcher 1 [92] provided a rectangle widget named TimeBox.

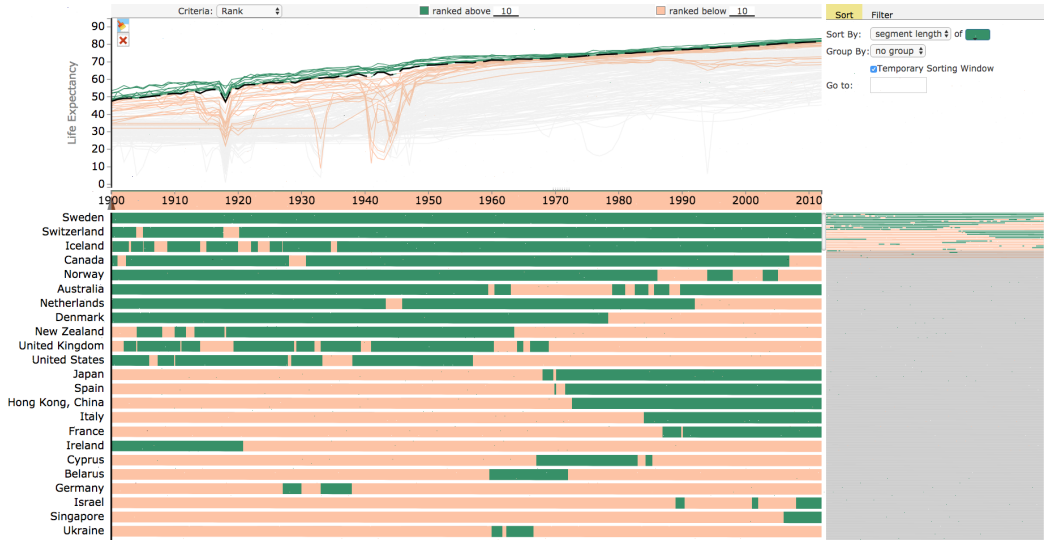


Figure 5.2: A screenshot of EventPool that answers the question "which countries were among the top 10 in the world with regard to LE and when?". The time periods when a country is among the top 10 are colored in green. Other time periods are colored in orange. Countries that had never entered the top 10 are grayed out.

Users can manipulate it to specify query constraints; time series passing through the timebox are selected and highlighted for detailed analysis. An angular query widget allows users to select time series with desired slopes. TimeSearcher 2 [93] provided a SearchBox widget, allowing users to select a segment of a time series and search for similar patterns in other time series. These approaches quickly get cluttered when the number of time series involved in the results increases. EventPool provides a wider range of queries, such as queries about ranking and queries comparing a focus time series with other time series throughout the entire time span. It uses semantic encoding and coordinated juxtaposed views to overcome the clutter problem.

Juxtaposed techniques, such as the horizon graph [86], the line graph explorer [87], and two-tone pseudo coloring [94], display time series side by side to avoid overlapping. However, they become less effective when the number of time series displayed increases due to the intensive cognitive efforts required to decode the colors and shapes [89]. Interactive Horizon Graphs [95] allow users to interactively adjust the baseline of the horizon graph to change colors and shapes for pattern discovery. The Qualizon graph [96] colors time series segments in different value ranges with

different color families. Data density of visualized quantitative values is increased and qualitative abstractions become visible. None of the above approaches provided queries with change, variation, ranking, and ego-centric thresholds. Their interactions were also quite different from those provided in EventPool. EventPool does not show quantitative values in its juxtaposed views as those approaches.

Exploring ranking of time series is a challenging task. To address this problem, RankExplorer [97] used a stacked graph [98, 99] to show how the counts and rankings of words searched in search engines change over time. It partitioned a large number of time series into a manageable number of ranking categories and visualizes their total counts using a stacked graph. Rather than displaying such aggregated information, EventPool focuses on individual time series that satisfy a given ranking threshold at each time step. It is effective in discovering ranking-related insights about individual time series, which is not a focus of RankExplorer. Perin etl al. [100] directly displayed the ranking of the time series as a line chart. This approach is intuitive, but it may suffer from the clutter problem when the number of series increases.

Tominski [101] presented a general event-based visualization model with three steps: specifying event criteria, finding events, and automatically adjusting visual representations according to the detected events. He pointed out that dynamic queries can be used for event detection and there are implicit and explicit event representations. The former alters visualization parameters to reveal events, and the latter visualizes events rather than data. EventPool fits well into that model, with dynamic queries for event specification and both implicit (colored line graph) and explicit (colored timelines) event representations. That is why it was named EventPool.

Shahar et al. [102] proposed knowledge-based temporal abstraction (KBTA) for clinical data analysis. The input includes a set of time series with multiple numerical attributes and events. The output includes a set of context-specific events presented as time intervals. Bade et al. [103] used the KBTA idea in visualization to im-

prove body temperature monitoring. The KNAVE II system [104] allows analysts to investigate multiple attributes simultaneously with event timelines. It supports semantic navigation of clinic events with a predefined ontology. VISITORS [105, 106] provided an exploratory interface to query time segments over time, patient and targeted events. In these approaches, event detection was tied to a priori or associated domain knowledge and was an automatic process separated from the visual analysis, which is different from the dynamic query driven visual exploration in EventPool. These approaches only allowed users to inspect one or a few time series at a time, with the exception of VISITORS [105, 106]. VISITORS allowed users to inspect the records of multiple patients at the same time by aggregating patients with similar events into bar charts. It is different from EventPool where the time series are not aggregated.

Both EventPool and ChronoLenses [107] conduct transformation on time series data, but along different directions. ChronoLenses apply transformations (e.g. 1st derivative) on data and display the transformation results, which are usually more cluttered than the original data, for more analytical power. EventPool simplifies data and visualizations so that users can visually discover insights more effectively.

5.3 EventPool

5.3.1 Design Requirements

Our approach consists of three steps: (1) dynamic queries; (2) semantic encoding; and (3) comprehensible inspection. The following requirements need to be addressed: In step 1, multiple types of queries should be provided to support a variety of "which and when" tasks (R1-1). Instant visual feedback of the queries is needed to help users adjust the thresholds efficiently (R1-2). In step 2, qualitative values are assigned to time series segments, which will be mapped to colors in the visualizations. Thus the number of qualitative values should be small to avoid visual clutter in the displays (R2-1). In step 3, the visualizations should scale to at least hundreds of time series over

hundreds of time steps (R3-1). They should be visually pre-attentive with semantics, such as when, who, and how long a pattern is, explicitly represented (R3-2). Users should be allowed to examine the raw values of time series identified from views where details are suppressed by semantic encoding (R3-3). Users should be allowed to change the views based on drifting interests (R3-4). In the following sections, I discuss how EventPool is designed according to these requirements.

5.3.2 Queries

To address requirements R1-1, a set of range queries that vary in the following factors are considered in EventPool: the number of ranges, original values or derived values, and constant thresholds or thresholds changing over the time span. I introduce the factors in this section. The case studies in Section 5.4 illustrates the usefulness of the varying queries.

Number of ranges: Two range queries can be defined by one threshold and its results can be represented using two qualitative values. For example, if I set a threshold of 50 years on LE, the two ranges can be values less than 50 and values larger than or equal to 50, respectively. A country whose LE is less than 50 in a year will be projected to the qualitative value "less than 50 years" and otherwise it is projected to "larger than or equal to 50 years" in that year. Similarly, three range queries can be defined using two thresholds which cut a value scale into three ranges. Their results are projected to three qualitative values. Currently, EventPool only supports queries with two ranges or three ranges according to R2-1, but the number can always go up at the cost of more qualitative values encoded in step 2.

Values: The values used in the queries can be either original data or any derived values. Here are some examples of derived values: **ranking** (the relative position of a time series in the original value space), **percentage change** (the relative gain/loss of the value compared with the previous time segment), **net change** (the gain/loss of the value compared with the previous time segment), and **variance** (the average

of the squared differences from the mean over a time window centered in this time segment).

Constant/varying Thresholds: The thresholds used in the queries can be constant or varying throughout the entire time span of a dataset. "Top 10" and "LE of 50 years" are examples of constant thresholds. EventPool also supports an ego mode, where the thresholds are tied to a referencing time series called an ego or a focal series. For example, Figure 5.1 shows an ego mode and the focal time series is the US. The thresholds are "0.5 year longer/shorter than the LE in the US".

Once a query is conducted, the numeric values of a time series are projected to a few qualitative values, namely the names of the ranges they fall in, such as "above 50 years" and "lower than or equal to 50 years".

5.3.3 Visualizations

In EventPool, several views are coordinated to tackle the requirements about visualizations. A line graph, which is intuitive and familiar to users, is used to provide instant visual feedback to dynamic queries (R1-2). In particular, the line segments are colored to encode the qualitative values in individual time segments. A consistent color coding is used in all the visualizations. Users can roughly estimate the size of the result set from the line graph and interactively adjust the queries based on the estimation. Different from existing approaches such as Time Searcher 2 [93], there is no need for users to examine the results in detail from the line graph (therefore the clutter is not a problem). That task is conducted in other views. When users find time series of interest from other views, they can examine the individual time series in the line graph (R3-3).

Juxtaposed timelines are used to represent the qualitative values resulting from a query, where each time step is represented by a colored small bar placed along the time axis. There are two timeline views in EventPool. One is a detail view and the other is an overview. In the detail view, each time series is represented by a timeline.

The colors in the timeline encode the qualitative values in individual time steps, with the same color coding as in the line graph. The timelines are assigned a constant height to ensure that their labels, colors, and patterns are readable (R3-2) and that users can interactively select a timeline with ease to examine it in the line graph (R3-3). A consequence of this design is that only a portion of the time series in a big dataset can be displayed in the view at the same time. Two approaches are used to increase the scalability (R3-1). First, an overview is provided where the timelines are displayed at a reduced height without the labels. It is a zoomed-out of the detail view with all time series displayed. Since only a small number of colors are used in the timelines, dataset-level patterns can be observed in this view. The time series currently displayed in the detail view are highlighted in the overview. Users can drag a scrolling bar to scroll the detail view to time series with interesting patterns according to the overview. Second, interactive sorting is provided to automatically bring time series of interest to more visible positions. The sorting function is discussed in Section 5.3.4.

The time axis is displayed between the line graph and the timeline views. A stacked bar chart (see Figure 5.4 (b)) is displayed within the time axis. It shows the percentages of time series with individual qualitative values at each time step. The higher a bar is, the higher the percentage of the series with the qualitative value encoded by the color of the bar is in that time segment.

5.3.4 Interactions

Dynamic Queries: Users can interactively construct a query through a row of widgets displayed on the top of the line graph (see Figure 5.1). They can use the widgets to choose the type of the query and set the thresholds. In the ego mode, users can either type the name of a series in an input box or double click it from any views to set it as the ego.

In some special cases, boundaries can be drawn in the line graph to separate line

segments falling into different ranges. Examples include queries evaluating ranking and values (see Figures 5.1 and 5.2). For these queries, EventPool explicitly draws the boundaries in the line graph and allows users to directly drag the boundaries to change the thresholds. This interaction is intuitive and convenient for users. For all queries, a slider will pop up if users click a threshold input box. Users can directly drag the slider to modify the query with instant visual feedback from the line graph. This function is especially useful when statistics values such as variance are used in the query, since it is difficult for users to set an appropriate threshold in these queries.
Sorting: EventPool provides a sorting function to help users organize time series in the timeline views based on query results and other measures. It serves for two tasks: first, it allows users to access the most interesting time series in the detail view without scrolling the screen. Second, sorting often reveals interesting dataset-level patterns since they can bring similar series near each other.

The following are our considerations when I designed the sorting function: to allow users to focus on a qualitative value of interest and keep the ranking intuitive, the sorting should be focused on one qualitative value at a time; since patterns are often time-relevant, users should be allowed to rank the series based on a desired time span; and users may be interested in how long a series keeps a qualitative value, how much they change with that value, or how much they change during sorting time window.

To address these requirements, users can set a focus qualitative value, a time window, and a sort criteria in sorting. From the "sort by" widget, users can select a color to encode the qualitative value of interest. They can click the triangles on the right of the time axis to turn on/off a time window on the time axis. Moving it horizontally and dragging and dropping its handles will change its size and position. If the time window is turned off, the ranking will be conducted over the whole time span. From the "sort criteria" widget, users can select from "segment length", "segment average", or "total difference". EventPool will rank the time series based on the total length of

time segments with the selected color in the time window, average values of the time series on the segments with the selected color in the time window, or the difference of the time series values at the two ends of the time window, respectively. The ranking decides the order of the timelines in the detail view and the overview.

For example, in Figure 5.1, green, which represents "0.5 year higher than the US", is selected. The sort criteria is "segment length" and the time window is turned off. Therefore, the timelines are sorted based on the number of years when a country has a LE 0.5 year higher than the US, namely the total length of the green bars, over the whole time span. Users can find countries consistently with a better LE than in the US from the top of the detail view. From the overview, users can notice that most countries that have relatively small numbers of qualified years surpassed the US in recent years.

In addition, the "sort by" color also decides the visibility of the line segments in the line graph. Since users have expressed a strong interest in the qualitative value (color) they selected, line segments with this color will always be drawn on the top of other line segments so that they can be seen clearly.

Grouping: Time series in a dataset often belong to different categories. For example, in a stock price dataset, the stocks belong to different sectors; in a world health dataset, the countries belong to different continents. Such categorical information is often important in insight discovery. EventPool provides a grouping function to cope with this need. When users turn on the group mode through a widget in the interface, the timelines are sorted first by their categories and then by the sorting criteria. The categories are also sorted by their average values of the sorting criteria. Figures 5.4 (a) and 5.4 (b) show examples of grouping. In these figures, countries are grouped by continent. The color legend of the groups are displayed at the bottom of the interface. When users hover the mouse over a series, its group label is highlighted in the legend, as shown in Figure 5.4 (b).

The grouping function organizes time series into a one level hierarchy. It can be further extended to support hierarchies with multiple levels. The visualization can be similar to Timeline Tree [108], where a node-link tree visualization is attached to the timelines.

Filtering: EventPool allows users to remove line segments (continuous time segments with the same color) that are too long or/and too short from the visualizations through a filtering tool (see Figure 5.3 (b) on the top right). If users are looking for continuous patterns, they can remove short segments. If they are looking for transient patterns, they can remove long segments. They can remove segments with all colors or with a specified color using the dropbox on the top of the tool.

Two horizontal sliders are used to set thresholds (upper limit and lower limit) for filtering. Segments whose lengths are out of the range are removed from the display. A linear value mapping from the range of segment lengths to the sliders is used as a common practice. However, since the lengths of the sliders are limited, it is difficult for users to make fine adjustments desired when removing tiny segments. To address this issue, I use function $y = c^n - (c-x)^n, x < c$ to project small changes in segment length x to big changes in y when x is near zero. c is the maximum length of a segment, which equals to the length of available time period. I visualize the projection function using a curve with two horizontal bars over it. The horizontal offset from the left end of the curve to the upper/lower bar's intersect with the curve determines the lower/higher limit of the filtering. Users can interactively drag the bars up and down to change the limits. Obviously, at the pointy top, a tiny horizontal change maps to a much bigger vertical change. Therefore, users can make fine adjustments for a threshold near zero precisely by vertically moving the bar.

The filtering tool is coordinated with the visualizations and instant visual feedback is provided when users move the bars or the sliders. If segments with different colors have different thresholds, each threshold will be displayed in the tool with the

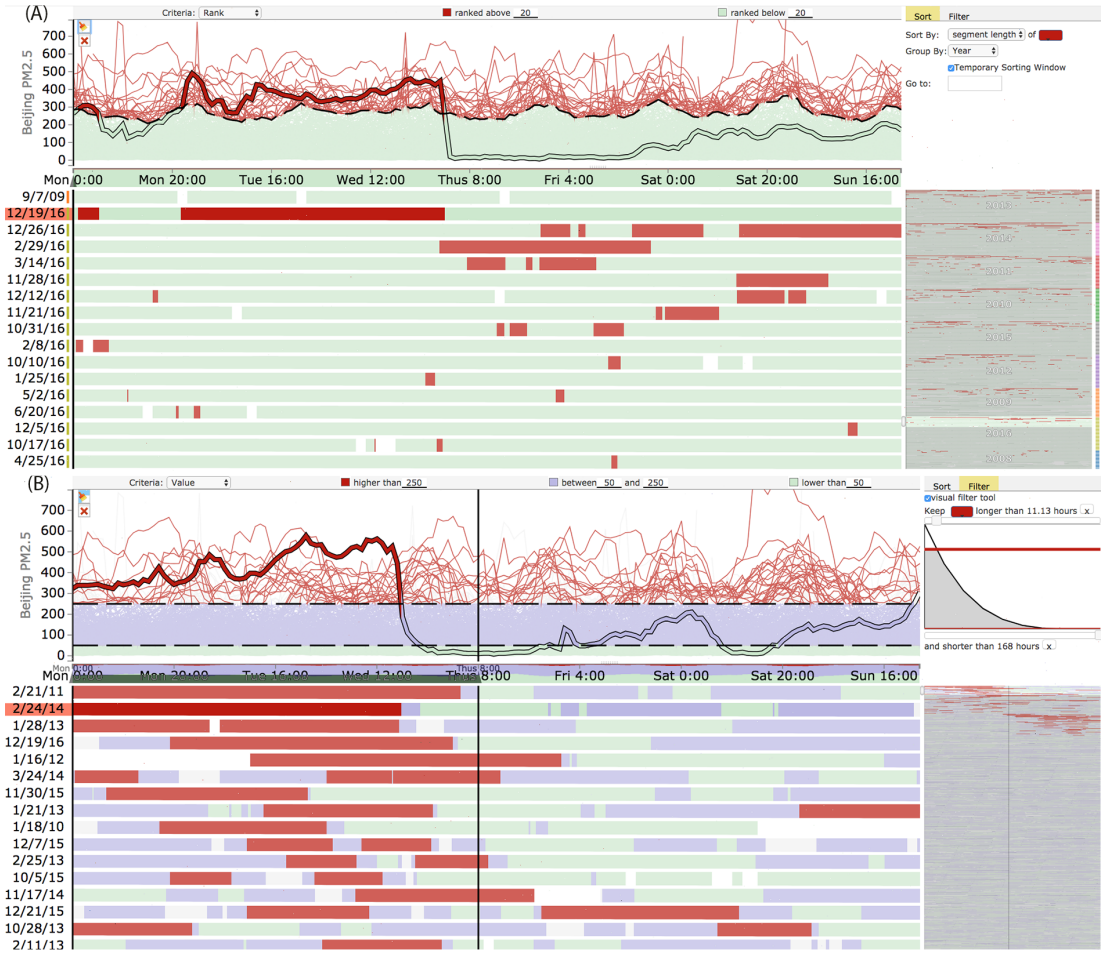


Figure 5.3: The PM2.5 dataset containing information about atmospheric particulate matter (PM) with a diameter less than 2.5 micrometers displayed in EventPool. (a) Red - the top 20 time series in PM2.5 value in each hour; Green - other time series. (b) Red - hazardous conditions with PM2.5 values higher than 250; Green - good conditions with PM2.5 values lower than 50; Purple - other conditions.

corresponding color.

Highlighting and Selection: Users can hover the mouse over a series in any views to highlight it in all visualizations. If the timeline is not in the detail view, a preview of the timeline and its neighbors are overlaid on the detail view to allow users to examine them in detail without scrolling the detail view (see Figure 5.4 (b) for an example). By clicking a timeline, users can select it and keep it highlighted after the mouse is moved out. With the shift key pressed, users can select all series between two series they click in the detail view. It is called a **bulk selection**.

In the line graph, the selected series are drawn in thicker lines than other series and displayed on the top layer of the visualization. In the timeline views, since they are already compact, yellow tags are attached to selected series without changing the timelines themselves.

5.4 Use Scenarios

5.4.1 Scenario 1: Analyzing Beijing Air Quality

In this Scenario, Xin, a resident of Beijing, explored her city's records of PM 2.5 density, an important measurement of air quality. The dataset contained hourly records of PM 2.5 density from April 2008 to January 2017. To study weekly patterns, the time series was divided into 440 weekly time series (weeks without any data were removed). Each of them has 168 hourly records, covering seven days from Monday 12:00 am to Sunday 11:00 pm.

Xin started her exploration by continuously dragging the black borderline of the rank query. She noticed that it always had a wave shape before the threshold went too low. She found that the peak of the black line always happened during midnight (Figure 5.3 (a)). It meant that the peak PM2.5 densities always occurred during midnight. She guessed that it may have been due to the low temperature during the nights or discharges from illegal plants running during the nights.

She was also interested in yearly patterns, so she grouped the timelines by years. The weeks were automatically ranked by the number of hours in the top 20 PM2.5 readings within a year, and the years were ranked by average number of hours in the top 20 PM2.5 readings. From the overview of the timelines (Figure 5.3 (a), bottom right), Xin found that there were weeks with heavy pollution levels (timelines with lots of red) in each year. She hovered the mouse across the overview to preview the timelines. She found that year 2016 was next to year 2008 (2008 only has 9 months of records) on the bottom, which means it has lower average number of peak hours than most years. She also noticed that there were fewer weeks with heavy pollution levels

in 2016. "Wow, air quality was actually getting better in 2016, although there were still bad weeks." Clicking the 2016 region in the overview, Xin brought the detail view to 2016 data (Figure 5.3 (a), bottom left), and hovered the mouse over the timelines to examine the heavily polluted weeks in 2016 one by one in the line graph.

Immediately, Xin noticed an interesting pattern in the week starting from 12/19/2016. That time series is highlighted in the visualizations in Figure 5.3 (a). It can be seen that there was a dramatic drop in the PM2.5 records in the middle of that week. Xin recalled that the Beijing government enforced emergency policies during that period of time, such as shutting down heavily polluting factories. It seemed that those policies effectively reduced the air pollution.

Then Xin wondered whether there were similar patterns in other weeks. She switched EventPool to the three ranges mode and used PM2.5 values 250, the official threshold for hazardous conditions, as the upper limit and 50, a threshold for good conditions, as the lower limit. Figures 5.3 (b) show results of this query. The green color indicates good conditions, the red color indicates hazardous conditions, and the purple color indicates other conditions. Since long hazardous conditions are more likely to invoke government actions, Xin filtered out short red segments and only kept hazardous conditions lasting more than 10 hours using the filtering tool. Following the pattern observed, Xin set the time window to cover the left half of the time axis and then sorted all series by hazardous hours in the time window (see Figure 5.3 (b)). Multiple weeks with long hazardous conditions in the first half of a week immediately popped out. One of them, the week starting from 2/14/2014, is highlighted in the Figure 5.3 (b) as an example. From the line graphs in Figures 5.3 (a) and (b), Xin saw that this series had a similar pattern with the week starting from 12/19/2016. The same pattern was also discovered from multiple other weeks in the detail view. They show the impact of the emergency policies by the government.

Figure 5.4: The WLE dataset displayed in EventPool. (a) Green/orange/purple indicates a country had higher/lower/approximately the same LE with the US at the time. (b) Green indicates a country was experiencing significant changes of LE at the time. A preview of three timelines in the overview was overlaid on the detail view. (c) Orange indicates a country had a continuous downtrend in LE at that time.

5.4.2 Scenario 2: Exploring the WLE Dataset

In this scenario, Tom, a middle school student, wondered if the WLE dataset might reflect historical events. He loaded the dataset into EventPool and selected the two-range mode. Assuming that certain historical events would make LE unstable, he set the target to be variance and dragged the threshold slider until he got Figure 5.4 (a). From the stacked bar chart in the time axis, he noticed that almost all countries were unstable around 1918, the year of a flu pandemic. Tom then found that World War II (WWII) also heavily impacted many countries in the 1940s. He set the time window to 1938-1952 and, from the variance slider, sorted the countries by the number of years

Figure 5.5: The stock price dataset displayed in EventPool. The stocks are sorted by price increases from more to less. Stocks with price plunges are highlighted. Orange - price plunges; Green - price hikes.

they suffered. From the top of the detail view, he identified many Eastern European countries, which means they suffered longer than other countries. Meanwhile, from the overview, Tom noticed that not all European countries were influenced by WWII that much. He hovered the mouse over those countries in the overview, and found countries such as Greenland from the preview overlaid on the detail view (see Figure 5.4 (b)). He found that several countries from both South Asia (blue) and East Asia (brown) were also influenced by WWII using a similar approach.

Now, Tom wanted to capture the long term impacts of historical events, especially when a country went consistently downwards. He set the query in "net change" mode and the threshold on -0.1 year. He then filtered out segments shorter than 3 years. The result is shown in figure 5.4 (c), where the downward trends are displayed in orange. He quickly noticed from the line graph that there are several time periods when many countries had a downward trend. He set the time window to 1983-1993 to cover one of them. From the overview, he found that the countries decreasing the most in this time period are from Africa and Europe. He scrolled the detail view to these European countries. From their timelines, he found that many of them also

experienced large decreases in the 1960s and 1970s.

5.4.3 Scenario 3: Stock Screening

In this Scenario, Jenny, a newbie to the stock market, used EventPool to analyze the stock market and find stocks for investment. The stock price dataset she used contained the closing price of 292 actively traded NASDAQ companies over a span of 300 days (from 1/20/2016 to 3/28/2017). Each company was represented by a time series with 300 time steps.

Jenny started from finding price hikes and price plunges in the stocks. She conducted a three range query on the dataset. Percentage change was used with -3% and 3% as the thresholds. She also used the filtering tool to remove line segments whose lengths are smaller than 2 days. As shown in Figure 5.5, each green segment is a price hike when the price of a stock experienced a change of more than 3% for more than two days; while each orange segment is a plunge when the price of a stock experienced a loss of more than 3% for more than two days.

Jenny then sorted the stocks by the total lengths of green segments. In the detail view, Jenny found many stocks with long price hikes. However, she also noticed that most of them also contained orange segments. Jenny realized that there were risks in these stocks. To examine their long term performance, Jenny highlights the top 30 series in the detail view (the ones with more price hikes) using a bulk selection. Then, she reordered the stocks by their total changes in the whole time span. From the overview, Jenny found that the highlighted stocks were evenly distributed in the ranked list, which means that a stock‘s long term performance is not related to how many price hikes they have.

To reduce risks, Jenny decided to stick with stocks without price plunges. She sorted the stocks by the total lengths of orange segments and dragged the detail view to the bottom. There, she found many stocks without any plunges. She highlighted them using a bulk selection and reordered the stocks by their total changes again.

Stocks whose prices increased the most are brought to the top of the detail view. Jenny happily found several highlighted stocks in them (see Figure 5.5). She hovered the mouse over them to examine them in the line graph, and found they were stocks with stable increases. One of them is highlighted in the line graph of Figure 5.5 as an example. Jenny decided that they are candidate stocks for her investment.

5.5 User Study 1

To evaluate the effectiveness and efficiency of EventPool in examining query results, our first user study compared it with a line graph system (Baseline for short) I developed. Besides a selection box similar to the TimeBox in TimeSearcher 1 [92], all the dynamic query functions available in EventPool were provided in Baseline. It is difficult to conduct the queries needed for answering most tasks in existing systems. The semantic color encoding was also provided in Baseline to help the subjects examine query results. Since EventPool has a grouping function, I provided one in Baseline. By clicking it, a list of group names is displayed and users can select one or more groups to display and hide time series in other groups in the line graphs. Users can thus examine the groups one by one in a less cluttered view. Another option was to display group information using colors. In addition, if multiple time series were selected by the selection box in Baseline, their labels are listed on the right side of the line graphs. When users click a label, its line graph is highlighted. To make it fair, Baseline enlarges the line graphs so that it occupied the same amount of space as all the views in EventPool.

Twelve seniors and graduate CS students from UNC Charlotte participated in the study. Since all of them were taking the same visual analytics class, I assumed they all had a similar background in visualization. The subjects were randomly assigned to two groups of 6 subjects. One group tested Baseline first and EventPool second. The other group tested EventPool first and Baseline second.

Two five-question sets about the WLE dataset were used. They contained tasks of

similar types, similar difficulty levels, and different parameters. Half of each group worked on set 1 using Eventpool and set 2 using Baseline. The other half answered set 2 using Eventpool and set 1 using Baseline. Here are the tasks: Task 1 (Comparison): "Which country exceeded Korea (set 1)/the US (set 2) most recently (T1)?"; Task 2 (Extreme): "Which country stayed in the top 10 (set 1)/15 (set 2) for the shortest amount of time? "; Task 3 (Extreme): "Who is the newest member of the top 10 (set 1)/15(set 2)?"; Task 4 (Extreme): "Which former member of the top 10 (set 1)/15 (set 2) have never joined it again since 1950? "; and Task 5 (Condition): "Identify at least 3 countries that were not affected by Wold War II (1939-1945) but suffered at some time after it (set 1)/Identify at least 3 countries that suffered three decreasing periods (set 2)".

In this user study, I aim at examining 1) whether EventPool users capture details of interest more effectively and efficiently than Baseline users; 2) whether EventPool users judge the accuracy of their answers better than Baseline users. Goal 1 was evaluated by the accuracy of the answers and the task completion time, which were captured with LimeSurvey[109], an online survey tool. To evaluate goal 2, I asked the subjects to rate their confidence level on their answers and compared the ratings with the actually accuracy of their answers.

The subjects conducted the study one by one in front of a 27" screen with 2560*1440 resolution, where the interface of EventPool or Baseline occupied the whole screen. Two other screens were placed beside the 27" screen. LimeSurvey was used on one screen to show questions and capture answers and completion time through a survey during the formal study. The other screen displayed text instructions for interactions during the formal study.

Each study consisted of two consecutive sessions, one for each system. Each session ran as follows: first, an instructor introduced the visualizations and interactions of a system to a subject through a demo exploring the world GDP dataset [25], which

contains the GDP values of 197 countries from year 1990 to 2012. Then, the subject spent 15 minutes to become familiar with the system, when they addressed a set of tasks similar to the test tasks on the world GDP dataset with the help of the instructor. After that, the subject conducted the test tasks on the WLE dataset independently. The answers and the completion time were collected through LimeSurvey. Immediately after answering each question, the subject was also asked to provide a confidence measure about his/her answer in a scale of 0-5 (5 is the most confident). After the second session, the subject was asked to orally compare the two systems. Their oral comments were recorded.

Table 5.1: Average and stand deviations of the results in user study 1.

	Time (sec)	Accuracy	Confident 1	Confident 2
Task 1				
EventPool	73 (27)	96% (0.14)	4.83 (0.41)	4.67 (0.52)
Baseline	178 (98)	63% (0.48)	4.5 (0.84)	3.8 (0.45)
Task 2				
EventPool	53 (25)	100% (0)	4.83 (0.41)	4.83 (0.41)
Baseline	171 (73)	0% (0)	4.83 (0.41)	3 (1.67)
Task 3				
EventPool	43 (26)	92% (0.29)	5 (0)	4.8 (0.41)
Baseline	124 (30)	50% (0.52)	4.5 (0.84)	3.67 (1.63)
Task 4				
EventPool	68 (25)	94% (0.2)	4.83 (0.41)	4.67 (0.52)
Baseline	230 (51)	66% (0.45)	4 (1.1)	3.2 (2.05)
Task 5				
EventPool	88 (42)	91% (0.22)	4.83 (0.41)	4.67 (0.82)
Baseline	185 (69)	74% (0.35)	4 (0.89)	4.17 (1.17)

Table 1 shows the results of the study. In all the tasks, EventPool users achieved better accuracy in less time than Baseline users, suggesting that EventPool users examined the query results more effectively and more efficiently. In particular, Baseline users took three times longer than EventPool users on average to answer Task 2, but they had a zero accuracy rate compared to 100% accuracy rate of EventPool users.

According to the confidence scores, EventPool users were highly confident about

their answers, which was in line with their high accuracy rate. When subjects used Baseline first, they were also highly confident about his/her answers with Baseline, but they had many wrong answers. Interestingly, when the subject used EventPool first, their confidence scores with Baseline were generally lower and aligned better with the accuracy measure. A possible explanation is that subjects realized that in Baseline, they missed many details visible in EventPool, and thus became less confident with their answers.

5.5.1 User Feedback

The subjects consistently preferred EventPool in their oral comments. Here are some quotes of representative comments: "The second one (EventPool) is better because it was less time consuming and visualization was more accurate. The first one (Baseline) needs really a lot of operations, and the second one is more direct to get the answer. So, less time consuming less pressure." "I feel the first one (EventPool) is good. The selection is far easier. It's very easy and fast. You would have everything in one glance." "The line chart can be helpful sometimes but sometimes it's really hard to pick up the specific range you're looking for, and having the bar as below is easier to see where the green is and orange is for that specific country. The second one (Baseline) is very hard to do that because you have to go through each of the line." "I like the first one (EventPool) that you can see all the different countries to compare. If you have a prior knowledge you might have an understanding like why it happened." "The visualization (of EventPool) is more intuitive and accurate. The second (Baseline) is just the lines so it was hard to click one particular line identified that country. It required more time and I was not sure if I selected the correct line. It was confusing." "I think the second system (EventPool) is more intuitive and it is easier to understand and check the data, because this system represents very neatly."

5.6 User Study 2

Another user study was conducted to (1) learn whether novice users can effectively use EventPool in exploratory analysis, and (2) learn the exploration strategies users may have with EventPool. Ten graduate students in CS major (7 without visualization background and prior knowledge of EventPool, 3 with 1 year experience of visualization but no prior experience with EventPool) participated. The study was open ended and the subjects were asked to find interesting patterns from the WLE dataset with a limited amount of training.

The subjects conducted the study one by one. Each study had a 15-minute training session followed by a 15-minute exploration session on EventPool. In the first 10 minutes of training, the instructor introduced the interactions using the World GDP dataset. Then, a subject had 5 minutes to become familiar with the system. Then, the subject spent 15 minutes to explore the WLE dataset. During the exploration process, the subject was asked to say the motivations of his/her activities, tasks he/she wanted to conduct, and interesting insights he/she discovered. The instructor sat beside him/her and provided suggestions on which interactions he/she could use to conduct his/her tasks. The audio was recorded along with the system screen for result analysis. At the end of the study, the subject provided oral comments, which were recorded.

The following observations, comments, and insights were collected: **Effectiveness**: The subjects discovered many types of insights. For example, six subjects found that Russia, Ukraine, and Kazakhstan had correlated LEs before WWII (correlation). Most subjects noticed the global drop of LE caused by the 1918 flu pandemic (trend), and four of them discovered that there were only six countries not significantly influenced by it (outliers). Most subjects identified the big drop of LEs in a large number of countries during World War II (trend). A subject further found that Switzerland and the United Kingdom suffered from the war less than many other European coun-

tries (comparison). Two subjects spotted that countries such as France decreased and increased LE twice during the war. Six subjects pointed out that Europe was damaged the most by the war, followed by East Asia and South Asia, while LEs of countries in the Middle East and America increased rapidly during the war (group). **Strategies**: Many interesting exploration strategies were observed from the subjects. Five subjects continuously lowered the value or ranking threshold. They found patterns such as the 1918 flu pandemic and World War II in this way. Three subjects placed the ranking borderline very low, and continuously moved a narrow time window to find the countries with the lowest LEs in different time periods, and generated many hypotheses to explain it. One subject further used filtering to find out who was always at the bottom.

The subjects often looked for insights from the overview. For example, a subject fixed the value threshold, looked for "gaps" (orange segments within long green segments) from the timeline overview, and examined them one by one in other views for insights. A subject commented that "(Eventpool) has the overview I can click and quickly access my desired parts. It makes me able to think about the information I want". Comparisons were frequently conducted on the timeline views for insights. The subjects often vertically scanned the timelines to compare the widths of gaps in different countries to find countries more or less influenced by a mass event. They also horizontally scanned the timelines to find countries involved in multiple events.

I observed that all the subjects changed the query threshold or switched to another type of queries when there was no more interesting patterns in the visualization, which often led to new patterns, new tasks, or new hypotheses. Many of them were able to use multiple interactions together, sometimes in a creative way. For example, a subject queried on the top 10 countries and sorted all countries by the total difference in LE. He found that South Korea, who joined the top 10 in 2008, was also the country with the biggest LE increase, while other top 10 countries didn't increased so much. A

subject applied filtering on speed query results to find long-term increasing segments. She found two groups of segments: one from 1920 to 1950, and the other from 1951 to 1985. Then she learned that only less than half of the countries appearing in trend 1 also exhibited trend 2 using the time window and bulk selection interactions. Three subjects highlighted a country of interest and changed the rank threshold to learn when the country reached different ranks.

The above findings suggest that the usage of EventPool is not limited to answering given "which and when" questions on extreme, condition, and comparison. As a subject commented, "I felt I was motivated by the functions and visualization (of EventPool). I was not interested in the dataset at first. However, the built-in analysis framework is easy to follow. Then I got interested.", EventPool is an exploratory analysis tool that triggers unexpected discoveries. Three subjects were so engaged that they requested for more time to explore with EventPool after the exploration session ended.

Another gain from this study was an observation about EventPool's learning curves. I noticed that the first two subjects started using EventPool from different queries and experienced quite different learning curves. The first subject started with two range queries on value, which is the simplest and most intuitive query in EventPool. He was able to find insights immediately with simple interactions. Later, when he was more familiar with the views, he switched to more complex queries and interactions with no difficulties. He discovered many high quality insights and provided highly positive comments. As a contrast, the second subject started with a complex three ranges queries with net changes. She struggled quite a while before she became productive. Based on this observation, I suggested to the remaining subjects that they start with the two range queries on value. This strategy worked well and none of them experienced a learning curve as steep as the second subject experienced. Once this strategy is proven by further user studies, it will used to guide the design

of EventPool training.

5.7 Discussion on Scalability

To scale EventPool to larger datasets (say, thousands of time series and thousands or more time steps), two bottlenecks need to be addressed. First, queries need to be conducted fast enough for interactive exploration. Second, visualizations should not be cluttered. The first challenge may be addressed using advanced indexing techniques, such as Nanocubes [110] or parallel query computation. To address the second challenge, a multiple screen design may be used, where the overview can be displayed in more space. Filtering functions can be provided to automatically hide time series without user-specified patterns, such as a long gap. Sampling techniques, such as the blue noise sampling [111] or density-based sampling [112], can also be used to preserve patterns and outliers. Techniques for exploring time series with large numbers of time steps, such as SignalLens [113] and Kronominer[114], can be modified and adapted in timelines for enhanced scalability on time steps. It is interesting to extend EventPool for multidimensional time series, where the sortable table and semantic zooming ideas of LiveRAC [115] are inspiring. Besides timelines, calendar views [116] also may be adapted in EventPool for discovering date-related patterns.

5.8 Project Conclusion and Future Work

In this project, I propose EventPool, a new time series visualization system for datasets with hundreds of time series. Its core is the novel ideas of conducting queries at individual time steps and comprehensively inspecting the qualitative query results in a familiar line graph and timelines leveraged with semantic encoding. EventPool allows users to effectively and efficiently conduct low level analytical tasks about extreme, condition, and comparison. A rich set of interactions are provided for exploratory analysis. Users can interactively change the type of queries, thresholds, and a focus time span for inspirations of new insights, tasks, and hypotheses.

5.9 Mantra Reflection in This Project

The spot-tracking lens and the interactive movie finder both use an object as the focal and extract its related objects. It can be understood as looking at the details through a microscope. However, for synopsis tasks where all data entries need to be considered at once, a telescope is needed to intelligently discard all irrelevant details and deliver the big picture. The Eventpool system makes users explore time series datasets with a telescope. The semantic encoding technique projects the numerical data into a semantic space which is more manageable and meaningful in analysis. It is the ego in the "land, modify, and shift" mantra. The encoding result is visualized using timelines, along with a linechart to explain the encoding and support visual threshold adjustment.

The modify step includes all interactions to reorder the timelines to support event-based tasks, such as finding time series staying the longest in one event value and finding time series that change states simultaneously. With the rich interactions, a user can sort and compare the timelines within a given time window, within and across groups, and based on three different criteria.

The challenge is that all critical information needs to be demonstrated in the plain view so that information mining would not interfere the ego shifting, or adjusting the semantic encoding thresholds in this case. I created two views to allow users to see important details immediately as well as to get an overview with less details: the main view is devoted to show the top ranked time series clearly with labels, while the overview shows how each semantic value is distributed in the whole dataset so that user can notice interesting shapes.

CHAPTER 6: Design Space and Design Guidelines of the "Land, Modify, and Shift" Mantra

To recap, to ensure the overall effectiveness of an exploration, I need to address the following challenges: Showing diverse information (C1): users should encounter diverse information during exploration. Building user confidence (C2): users move forward in an exploration willingly and with confidence. Allowing users to move fast (C3): users should be able to judge the worthiness of a visualization and make decisions on the next move quickly. Supporting information diet (C4): users should be able to collect essential insights from one view quickly. Showing unvisited area (C5): users should be aware and able to access the unvisited information portions.

Each of the three projects addressed these design challenges in different ways. But many design concerns and approaches were shared. This chapter concludes these practices as the design space and design guidelines for the "land, modify, and shift" exploration mantra.

6.1 Design Space of the "Land, Modify, and Shift" Mantra

The aforementioned "land, modify, and shift" exploration is a chain of close-up analysis tasks to collect insights from different parts of a dataset. However, finding these close-up tasks is not easy. An effective task should be both meaningful to analysts and restrained for effective visualization. I separate task-searching process into two parts: First, users decide the type of insights wanted, such as objects matching certain criteria, outliers, similar objects, shared trends, etc. The type of insights determines how the visualization is shaped. Second, users decide where to find them. With the type alone, one can extract a large number of insights, which won't be able to fit into

the display. In my mantra, users define the scope of extracted information through an ego. The ego carries a semantic meaning for users to hold on while filtering and browsing the insights. In the previous projects, the ego is the focal object followed by the spot-tracking lens in the first project, the example movie in the item-based recommendation in the second project, and the semantic query that projects the time series into event sequences in the third project.

When users shift their ego, the visualization stays the same so that users always see the same type of insights. Meanwhile, by modifying the visualization, users change what kind of insights to see, which are still associated with the ego. These two interactions work together to keep the exploration smooth and flexible. How to select the ego and how to modify the visualization are two independent design spaces that determine the "land, modify, and shift" exploration.

6.1.1 Design Space of Egos

An ego should be a valuable analysis target so that each close-up analysis is meaningful and effective(C2, C3). Besides, the ego should be associated with different egos under different conditions so that modifying the visualization brings in new ego candidates intuitively (C1).

There are two approaches to break down a big information space. One is through a microscope. Namely, users look at the details of a small set of individuals and leave the rest invisible. The other is through a telescope. Users look at a highly simplified version of the dataset, as long as it supports their task. In my previous projects, the spot-tracking lens and the IMF use the microscope. The analysis is always about showing a small subset of objects at a time. They both use an object as the ego, which sets the analysis scope within its information neighborhood based on similarity. An object intuitively links to other objects conditionally close or similar to it. By modifying the visualization, users change conditions and extract different information.

The EventPool uses a telescope, as all analysis activities are conducted on a simplified version of the original dataset. The ego is a semantic query that transforms the numerical data into a more memorable and task-relevant semantic space. The query is naturally linked to other queries with differences in parameter settings. In EventPool, users can directly manipulate these parameters visually and see results immediately in the visualization.

6.1.2 Design Space of Visualization Modification

Theoretically, users can modify any elements of visualization, such as color, shape, layout algorithm, and coordinates. However, to make users focus on an exploration(C2, C3), a view should not be dramatically changed. In addition, regarding intuitiveness and effectiveness (C2, C3, C4), basic visualization techniques should be the best pick because they are universally recognized and approved. For each data structure, there are corresponding visualization techniques for it, such as scatter plot for two dimensions dataset and line graphs for time series. Fortunately, there is still a considerable design space left for modification. The options are as follows:

- Changing the way to extract an information portion: It is the most obvious way to replace the information in a view while keeping the visualization technique the same. For example, the IMF system shows multiple movie strips that a user can select. Each leads a user to movies under a different topic.

- Changing visual priority: Interactions such as sorting, filtering, and highlighting can be provided to control visual priorities without dramatically changing a view. For example, the order of movies in the IMF and the timelines in the EventPool are both results of multiple customizable metrics working together. The "spotlight" highlighting technique proposed in the spot-tracking lens allows users to not only see important bubbles but also follow them easily in a dynamic display.

- Changing the boundary of extracted information: Extending a view increases the scope and diversity of the encountered information without changing the visualization design. In spot-tracking lens, the spotlight function not only navigates a user's attention but also determines how large the visible area is. In the IMF and the EventPool, users can change the size of the visible part of the list based on tasks and screen size.

- Adding a side view for other tasks: An additional view panel can support tasks not covered by the visualization, such as providing access points to the information out of a view. In the EventPool, an overview shows the whole results in a compact view. A side view can also record exploration history. The IMF and spot-tracking lens both have an overview that allows users to examine the coverage of an exploration and access the unvisited part.

6.2 Design Guidelines of the "Land, Modify, and Shift" Mantra

The previous section discussed the design space for interactions. There are several other design concerns invoked by the aforementioned three projects.

6.2.1 Building Upon Basic Visualizations

Using the basic visualization techniques minimizes learning curves (C3) and builds user confidence (C2). Moreover, it opens the possibility of integrating the "land, modify, and shift" mantra to existing visualization systems using the same techniques. For example, the spot-tracking lens are designed as an optional tool of the animated bubble chart. The highlighting techniques, including the spotlight, selective labeling, and filtering, would not interfere with the inherited visual design. In the EventPool, a linechart is kept in the system because it is more recognizable to the general public than the event sequence view. So it will help users understand how the semantic encoding works.

6.2.2 Browsing Important Items in a Plain View

I recommend to present valuable insights in a plain view for information foragers to collect, so they would not need to spend too much time at one place in an exploration (C4). Users can apply different algorithms of filtering and highlighting to support various tasks.

It is important to reserve enough space for labels, since labels link discoveries in analysis with external knowledge. All my aforementioned projects show labels of important objects in an immediate view.

6.2.3 Summarizing Information Portion

Occupying a plain view with important details benefits information diet. However, users need more information to decide whether the portion is worth digging through (C3). One approach is providing a quick preview and the other approach is providing an overview. The fast-forward function in the spot-tracking lens allows users to quickly go through an extracted information portion, while the other two projects have overviews on the side which summarize it. In the IMF, each movie strip has a summary panel containing featured profiles and movies, giving users an idea about what to expect in the rest of the strip. The overview in the EventPool allows users to see the distribution of each semantic value over the whole dataset.

The detail view shows important details. The overview summaries the information portion. They are created for different purposes, and thus they are shown simultaneously in plain views. In this way, users can shift the ego and observe the changes seamlessly without other operations interfering.

6.2.4 Making Ego Candidates Accessible

Visualization using my mantra should intuitively provides access points to ego candidates. All my implementations have multiple ways to access ego candidates. Making ego candidates easy to find gives users a sense of control (C2). It is also a

strategy to increase diversity (C1). With a wide range of choices to move forward an exploration, users may shift an ego by accident or out of pure curiosity and get surprising discoveries. In the spot-tracking lens, all bubbles in the lens are accessible, even though most objects are visually insignificant. In the IMF, the summary panel provides access points to the centric movies in each topic. Access points to interesting objects can be preserved in a view to check out later. In the spot-tracking lens and the EventPool, users can highlight interesting items found in one view and trace them throughout an exploration. In the IMF, the exploration history at the bottom also provides access points to previous ego movies.

6.2.5 Recording the Exploration History with an Overview

The overview is important as it serves two purposes. First, it shows the coverage of an exploration and provides direct access to the unvisited areas (C5). Second, it shows where the current ego is in the global coordinates to keep users oriented in exploration. For example, the spot-tracking lens has an "overview" of the whole canvas of the animated bubble chart. The bubbles are initially black and will be lighted up by its length of appearance within the spotlight, aka the user's attention. The overview does not have to be perfect as long as it serves its design purposes well. The overview of the IMF is created from a small sample, which can be easily embedded into a 2D display.

6.2.6 Comparing the "Land, Modify, and Shift" Mantra to the "Overview First, Detail on Demand" Mantra

This section compares the "land, modify, and shift" exploration mantra to the most impactful and recognizable "overview first, detail on demand" exploration mantra. Both approaches have advantages and disadvantages regarding different data structures and exploration goals. To understand the difference better, imagine an explorer who is a tourist visiting a city. The "overview first, detail on demand" is like having

the city map to help navigation. While in the "land, modify, and shift" exploration mantra, an explorer is a flaneur guided by his/her inspiration.

The "overview first" mantra is preferred if the projection is effective, efficient, and reliable, like a well-designed city map, which will help the tourist quickly locate and check out interesting POIs. However, to achieve this goal, the dataset structure must be simple so that it can be projected and visualized in a 2/3D display without misleading users. It is less and less realistic to create such an overview nowadays because of the curse of dimensionality. Regarding the exploration goals, the "overview first" approach is more efficient if exploration coverage is a crucial goal. My mantra, on the contrary, has better scalability as it breaks down the complex information space into manageable information portions, but cannot guarantee the coverage as an efficient overview can do. However, an overview situates the users' mind to what the overview shows. It brings benefits only if the tasks are clear for the type of information. Tourists can quickly circle out places they want to go on a map. But they may miss lots of interesting sights on their way to these places. As a result, different tourists may all end with a similar image of the city. For these looking for unexpection and inspiration, they may want to personalize what they see, even though the "important insights" are not guaranteed. My mantra is devoted to extending potentially accessible areas by increasing the randomness in exploration and presenting details, where unexpected insights are usually hidden. Every time visualization will bring in something different for users to explore. For example, the spot-tracking lens makes my user study subjects to follow less popular countries and see their development. For the same reason, my mantra can be applied to build a stand-alone movie recommendation system emphasizing diversity and serendipity.

CHAPTER 7: Conclusion

This dissertation proposes a new exploration mantra that allows users to collect items and insights from a massive dataset in an uncluttered way. The exploration process is a chain of interest-driven close-up analysis tasks. Each step is targeting a different information portion to build a thorough knowledge. The concept of "ego" is introduced to help users build effective tasks. Placing an ego in the dataset defines the center of extracted information portion. Modifying the visualization changes the rules of what and how the information would be extracted. Each ego shifting and modifying bring users new insights to investigate, and what users see will change their mind and inspire them what to do next. Then repeating interactions will capture changes in their drifting mind during the exploration. In this way, the "land, modify, and shift" exploration mantra navigates inexperienced users to ask effective questions in an exploration and get answers so that they can build an understanding of an unknown information space.

Compared to the more impactful overview first mantra, the "land, modify, and shift" mantra provides users more confident and motivated exploration experience with an easier start point, interest-driven interactions, and less overwhelming information. The flexible ad-hoc interactions encourage serendipitous discoveries and long-term interaction, rather than framing user's analysis perspective with an overview. However, giving that users only look at a dataset in a close-up way, it will be hard to guarantee a concrete "big picture" from the exploration. However, rich interactions are integrated to ensure the overall diversity, coverage and user satisfaction of the exploration, as is demonstrated in my three projects using this exploration mantra.

7.1 Spot-tracking Lens

The spot-tracking lens is a new zoomable user interface for animated bubble charts. To solve the change blindness problem, the zooming is coupling with automatic panning, which follows one object during animation. A set of novel techniques, such as frames of reference, spotlight, and selective labeling, are also introduced to navigate users' attention in a dynamic view. The project was motivated to allow users to enjoy the expected benefits of zooming, such as focusing on focal objects and their context, examining details and labels in a less cluttered view. Even though the project was motivated by solving the change blindness problem in an animated bubble chart, the user study showed subjects were more likely to study unfamiliar countries and discover interesting details. Without these interactions, the subjects could only see details in the uncluttered area, and most analysis efforts were devoted to popular countries. These changes in behavior inspire the study of a new exploration mantra that merges unexpectedness and diversity through close-up studies.

7.2 Interaction Movie Finder

The Interaction Movie Finder is a fully working interactive movie recommendation system. It allows users to observe the varying characteristics in movies and find movies of interest from a large number of diverse recommendations effectively and efficiently. It also promotes novelty and serendipitous discoveries through interactive visualizations powered by automatic algorithms. It integrates interactive visualizations with automated algorithms such as subspace clustering, weighted ranking, and collaborative filtering through the "land, modify, and shift" exploration mantra. In IMF, a user lands on a movie and sees the recommendation related to it. A User can increase his/her chance to meet movies similar to the ego movie regarding specific features. Through filtering and weighted sorting certain movie features can be enhanced on demand within each movie strip. Either way it modifies the content and

order of the browse list. The user can also pick a movie discovered in browsing as the new ego. In addition, the two side panels explain the relation between the ego movie and the movie strips. They help users build confidence by helping them reason the results. An overview is added to show the exploration coverage and provide access points to unvisited objects. These interactions work together to ensure the diversity, coverage and user satisfaction.

7.3 EventPool

The two projects above use an object as the focal and extract its related objects. It can be understood as looking at details through a microscope. However, for synopsis tasks that usually consider all data entries at once, a telescope is needed to intelligently discard all irrelevant details and deliver a big picture. The Eventpool system allows users to explore time series datasets with a telescope. EventPool is a new time series visualization system for datasets with hundreds of time series. Its core is the novel ideas of conducting queries at individual time steps and comprehensively inspecting the qualitative query results in a familiar line graph and timelines leveraged with semantic encoding. User studies showed that analysis based on those qualitative values can be much more efficient and effective for a wide range of tasks. The query setting is the telescope in the "land, modify, and shift" mantra. Users can interactively change the query targets and adjust thresholds while witnessing the result being updated immediately. In addition to that, a rich set of interactions are provided to sort, group, and highlight the timelines based on different analysis focuses.

In addition to these three implementations above, this dissertation also includes a discussion about the design space of the "land, modify, and shift" exploration mantra and provide some hand-on design guidelines for future reference. In a nutshell, the interactions and visualization should help user to push the exploration forward more widely, quickly, and confidently.

REFERENCES

[1] B. Shneiderman, "The eyes have it: A task by data type taxonomy for information visualizations," in *Visual Languages, 1996. Proceedings., IEEE Symposium on*, pp. 336–343, IEEE, 1996.

[2] P. Pirolli and S. Card, "Information foraging in information access environments," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 51–58, ACM Press/Addison-Wesley Publishing Co., 1995.

[3] D. J. Simons and C. F. Chabris, "Gorillas in our midst: Sustained inattentional blindness for dynamic events," *perception*, vol. 28, no. 9, pp. 1059–1074, 1999.

[4] L. Backstrom and J. Kleinberg, "Romantic partnerships and the dispersion of social ties: a network analysis of relationship status on facebook," in *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pp. 831–841, ACM, 2014.

[5] C. Prell, *Social network analysis: History, theory and methodology*. Sage, 2012.

[6] D. Fisher, "Using egocentric networks to understand communication," *IEEE Internet Computing*, vol. 9, no. 5, pp. 20–28, 2005.

[7] E. Bellotti, "What are friends for? elective communities of single people," *Social Networks*, vol. 30, no. 4, pp. 318–329, 2008.

[8] S. G. Roberts, R. I. Dunbar, T. V. Pollet, and T. Kuppens, "Exploring variation in active network size: Constraints and ego characteristics," *Social Networks*, vol. 31, no. 2, pp. 138–146, 2009.

[9] M. J. Lubbers, J. L. Molina, J. Lerner, U. Brandes, J. Ávila, and C. McCarty, "Longitudinal analysis of personal networks. the case of argentinean migrants in spain," *Social Networks*, vol. 32, no. 1, pp. 91–104, 2010.

[10] G. Ahuja, "Collaboration networks, structural holes, and innovation: A longitudinal study," *Administrative science quarterly*, vol. 45, no. 3, pp. 425–455, 2000.

[11] A. Abbasi, K. S. K. Chung, and L. Hossain, "Egocentric analysis of co-authorship network structure, position and performance," *Information Processing & Management*, vol. 48, no. 4, pp. 671–679, 2012.

[12] B. Lee, C. S. Parr, C. Plaisant, and B. B. Bederson, "Visualizing graphs as trees: Plant a seed and watch it grow," in *International Symposium on Graph Drawing*, pp. 516–518, Springer, 2005.

[13] F. Van Ham and A. Perer, ""search, show context, expand on demand": supporting large graph exploration with degree-of-interest," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, 2009.

[14] T. Crnovrsanin, I. Liao, Y. Wu, and K.-L. Ma, "Visual recommendations for network navigation," in *Computer Graphics Forum*, vol. 30, pp. 1081–1090, Wiley Online Library, 2011.

[15] J. Heer and D. Boyd, "Vizster: Visualizing online social networks," in *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pp. 32–39, IEEE, 2005.

[16] Q. Liu, Y. Hu, L. Shi, X. Mu, Y. Zhang, and J. Tang, "Egonetcloud: Event-based egocentric dynamic network visualization," in *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*, pp. 65–72, IEEE, 2015.

[17] F. B. Viégas, S. Golder, and J. Donath, "Visualizing email content: portraying relationships from conversational histories," in *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pp. 979–988, ACM, 2006.

[18] T. Nakamura and T. Igarashi, "An application-independent system for visualizing user operation history," in *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pp. 23–32, ACM, 2008.

[19] P. Resnick, R. K. Garrett, T. Kriplean, S. A. Munson, and N. J. Stroud, "Bursting your (filter) bubble: Strategies for promoting diverse exposure," in *Proceedings of the 2013 Conference on Computer Supported Cooperative Work Companion*, CSCW '13, (New York, NY, USA), pp. 95–100, ACM, 2013.

[20] M. J. Bates, "The design of browsing and berrypicking techniques for the online search interface," *Online review*, vol. 13, no. 5, pp. 407–424, 1989.

[21] G. Marchionini, "Exploratory search: from finding to understanding," *Communications of the ACM*, vol. 49, no. 4, pp. 41–46, 2006.

[22] P. Pirolli and S. Card, "The sensemaking process and leverage points for analyst technology as identified through cognitive task analysis," in *Proceedings of international conference on intelligence analysis*, vol. 5, pp. 2–4, 2005.

[23] G. W. Furnas, "Effective view navigation," in *Proceedings of the ACM SIGCHI Conference on Human factors in computing systems*, pp. 367–374, ACM, 1997.

[24] S. K. Card, P. Pirolli, M. Van Der Wege, J. B. Morrison, R. W. Reeder, P. K. Schraedley, and J. Boshart, "Information scent as a driver of web behavior graphs: results of a protocol analysis method for web usability," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 498–505, ACM, 2001.

[25] H. Rosling, "Ted 2006." http://www.gapminder.org/videos/ted-talks/hans-rosling-ted-2006-debunking-myths-about-the-third-world/.

[26] B. B. Bederson and J. D. Hollan, "Pad++: a zooming graphical interface for exploring alternate interface physics," in *Proceedings of the 7th annual ACM symposium on User interface software and technology*, pp. 17–26, ACM, 1994.

[27] A. Cockburn, A. Karlson, and B. B. Bederson, "A review of overview+ detail, zooming, and focus+ context interfaces," *ACM Computing Surveys (CSUR)*, vol. 41, no. 1, p. 2, 2008.

[28] K. Perlin and D. Fox, "Pad: an alternative approach to the computer interface," in *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pp. 57–64, ACM, 1993.

[29] L. Nowell, E. Hetzler, and T. Tanasse, "Change blindness in information visualization: A case study," in *Information Visualization, IEEE Symposium on*, pp. 15–15, IEEE, 2001.

[30] T. Miller and J. Stasko, "The InfoCanvas: information conveyance through personalized, expressive art," in *CHI'01 Extended Abstracts on Human Factors in Computing Systems*, pp. 305–306, ACM.

[31] J. Alsakran, Y. Chen, Y. Zhao, J. Yang, and D. Luo, "Streamit: Dynamic visualization and interactive exploration of text streams," in *Pacific Visualization, IEEE Symposium on*, pp. 131–138, IEEE, 2011.

[32] S. Huron, R. Vuillemot, and J.-D. Fekete, "Visual sedimentation," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 12, pp. 2446–2455, 2013.

[33] B. Bach, E. Pietriga, and J.-D. Fekete, "Graphdiaries: animated transitions and temporal navigation for dynamic networks," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 5, pp. 740–754, 2014.

[34] M. Farrugia and A. Quigley, "Effective temporal graph layout: A comparative study of animation versus static display methods," *Information Visualization*, vol. 10, no. 1, pp. 47–64, 2011.

[35] G. Robertson, R. Fernandez, D. Fisher, B. Lee, and J. Stasko, "Effectiveness of animation in trend visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 14, no. 6, pp. 1325–1332, 2008.

[36] T. Tekušová and J. Kohlhammer, "Applying animation to the visual analysis of financial time-dependent data," in *Information Visualization, 11th International Conference*, pp. 101–108, IEEE, 2007.

[37] R. S. French, "The discrimination of dot patterns as a function of number and average separation of dots.," *Journal of Experimental Psychology*, vol. 46, no. 1, p. 1, 1953.

[38] J. S. Yi, Y. ah Kang, J. T. Stasko, and J. A. Jacko, "Toward a deeper understanding of the role of interaction in information visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 6, pp. 1224–1231, 2007.

[39] 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*, pp. 19–24, ACM, 2004.

[40] W. Javed, S. Ghani, and N. Elmqvist, "Polyzoom: multiscale and multifocus exploration in 2d visual spaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 287–296, ACM, 2012.

[41] J. J. Van Wijk and W. A. Nuij, "Smooth and efficient zooming and panning," in *Information Visualization, IEEE Symposium on*, pp. 15–23, IEEE, 2003.

[42] T. Igarashi and K. Hinckley, "Speed-dependent automatic zooming for browsing large documents," in *Proceedings of the 13th annual ACM symposium on User interface software and technology*, pp. 139–148, ACM, 2000.

[43] C. Plaisant, J. Grosjean, and B. B. Bederson, "Spacetree: Supporting exploration in large node link tree, design evolution and empirical evaluation," in *Information Visualization, IEEE Symposium on*, pp. 57–64, IEEE, 2002.

[44] A. Ion, Y.-L. B. Chang, M. Haller, M. Hancock, and S. D. Scott, "Canyon: providing location awareness of multiple moving objects in a detail view on large displays," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3149–3158, ACM, 2013.

[45] P. Baudisch and R. Rosenholtz, "Halo: a technique for visualizing off-screen objects," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 481–488, ACM, 2003.

[46] C. Schlienger, S. Conversy, S. Chatty, M. Anquetil, and C. Mertz, "Improving usersâ comprehension of changes with animation and sound: An empirical assessment," in *Human-Computer Interaction–INTERACT 2007*, pp. 207–220, Springer, 2007.

[47] D. Archambault, H. C. Purchase, and B. Pinaud, "Animation, small multiples, and the effect of mental map preservation in dynamic graphs," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 17, no. 4, pp. 539–552, 2011.

[48] B. Kondo and C. M. Collins, "Dimpvis: Exploring time-varying information visualizations by direct manipulation," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 20, no. 12, pp. 2003–2012, 2014.

[49] D. Frederick, J. Mohler, M. Vorvoreanu, and R. Glotzbach, "The effects of parallax scrolling on user experience in web design," *Journal of Usability Studies*, vol. 10, no. 2, 2015.

[50] P. S. Heckbert and M. Garland, "Survey of polygonal surface simplification algorithms," tech. rep., DTIC Document, 1997.

[51] C. de Boor, "A practical guide to splines," *Applied Mathematical Sciences 27*, 2001.

[52] NHSDesigns, "The five elements of shading." http://nhsdesigns.com/pdfs/graphic

[53] J.-D. Fekete and C. Plaisant, "Excentric labeling: dynamic neighborhood labeling for data visualization," in *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pp. 512–519, ACM, 1999.

[54] "Gapminder." http://www.gapminder.org/.

[55] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, pp. 19:1–19:19, Dec. 2015.

[56] J. Tang, Z. Meng, X. Nguyen, Q. Mei, and M. Zhang, "Understanding the limiting factors of topic modeling via posterior contraction analysis," in *Proceedings of The 31st International Conference on Machine Learning*, pp. 190–198, 2014.

[57] TMDB, "The movie database," 2008.

[58] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, ACM, 2001.

[59] A. K. McCallum, "Mallet: A machine learning for language toolkit." http://mallet.cs.umass.edu, 2002.

[60] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.

[61] S. T. Dumais, "Latent semantic analysis," *Annual review of information science and technology*, vol. 38, no. 1, pp. 188–230, 2004.

[62] J. Chuang, C. D. Manning, and J. Heer, "Termite: Visualization techniques for assessing textual topic models," in *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pp. 74–77, ACM, 2012.

[63] B. Luo, R. C. Wilson, and E. R. Hancock, "Spectral embedding of graphs," *Pattern recognition*, vol. 36, no. 10, pp. 2213–2230, 2003.

[64] T. Arsan, E. Koksal, and Z. Bozkus, "Comparison of collaborative filtering algorithms with various similarity measures for movie recommendation," *International Journal of Computer Science, Engineering and Applications*, vol. 6, no. 3, 2016.

[65] T. Tanimoto, *An Elementary Mathematical Theory of Classification and Prediction*. International Business Machines Corporation, 1958.

[66] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[67] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 734–749, June 2005.

[68] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*, pp. 325–341, Springer, 2007.

[69] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *Internet Computing, IEEE*, vol. 7, no. 1, pp. 76–80, 2003.

[70] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[71] X. Chen, Z. Zheng, X. Liu, Z. Huang, and H. Sun, "Personalized qos-aware web service recommendation and visualization," *IEEE Transactions on Services Computing*, vol. 6, no. 1, pp. 35–47, 2013.

[72] J. O'Donovan, B. Smyth, B. Gretarsson, S. Bostandjiev, and T. Höllerer, "Peerchooser: visual interactive recommendation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1085–1088, ACM, 2008.

[73] L. Gou, F. You, J. Guo, L. Wu, and X. L. Zhang, "Sfviz: Interest-based friends exploration and recommendation in social networks," in *Proceedings of the 2011 Visual Information Communication - International Symposium*, VINCI '11, (New York, NY, USA), pp. 15:1–15:10, ACM, 2011.

[74] J. Stasko and E. Zhang, "Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations," in *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on*, pp. 57–65, IEEE, 2000.

[75] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[76] Y. C. Zhang, D. Ó. Séaghdha, D. Quercia, and T. Jambor, "Auralist: introducing serendipity into music recommendation," in *Proceedings of the fifth ACM international conference on Web search and data mining*, pp. 13–22, ACM, 2012.

[77] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*, pp. 22–32, ACM, 2005.

[78] M. J. Gardner, J. Lutes, J. Lund, J. Hansen, D. Walker, E. Ringger, and K. Seppi, "The topic browser: An interactive tool for browsing topic models," in *NIPS Workshop on Challenges of Data Visualization*, vol. 2, 2010.

[79] E. Alexander, J. Kohlmann, R. Valenza, M. Witmore, and M. Gleicher, "Serendip: Topic model-driven visual exploration of text corpora," in *Visual Analytics Science and Technology (VAST), 2014 IEEE Conference on*, pp. 173–182, IEEE, 2014.

[80] A. J.-B. Chaney and D. M. Blei, "Visualizing topic models.," in *ICWSM*, 2012.

[81] Y. Liu, S. Barlowe, Y. Feng, J. Yang, and M. Jiang, "Evaluating exploratory visualization systems: A user study on how clustering-based visualization systems support information seeking from large document collections," *Information Visualization*, vol. 12, no. 1, pp. 25–43, 2012.

[82] A. Tatu, L. Zhang, E. Bertini, T. Schreck, D. Keim, S. Bremm, and T. von Landesberger, "Clustnails: Visual analysis of subspace clusters," *Tsinghua Science and Technology*, vol. 17, no. 4, pp. 419–428, 2012.

[83] B. J. Ferdosi, H. Buddelmeijer, S. Trager, M. H. Wilkinson, and J. B. Roerdink, "Finding and visualizing relevant subspaces for clustering high-dimensional astronomical data using connected morphological operators," in *Visual Analytics Science and Technology (VAST), 2010 IEEE Symposium on*, pp. 35–42, IEEE, 2010.

[84] M. Sun, P. Mi, C. North, and N. Ramakrishnan, "Biset: Semantic edge bundling with biclusters for sensemaking," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 22, no. 1, pp. 310–319, 2016.

[85] W. Playfair, *The commercial and political atlas: representing, by means of stained copper-plate charts, the progress of the commerce, revenues, expenditure and debts of england during the whole of the eighteenth century.* T. Burton, 1801.

[86] J. Heer, N. Kong, and M. Agrawala, "Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualizations," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1303–1312, ACM, 2009.

[87] R. Kincaid and H. Lam, "Line graph explorer: scalable display of line graphs using focus+ context," in *Proceedings of the working conference on Advanced visual interfaces*, pp. 404–411, ACM, 2006.

[88] R. Amar, J. Eagan, and J. Stasko, "Low-level components of analytic activity in information visualization," in *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pp. 111–117, IEEE, 2005.

[89] W. Javed, B. McDonnel, and N. Elmqvist, "Graphical perception of multiple time series," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 6, pp. 927–934, 2010.

[90] L. Chittaro, C. Combi, and G. Trapasso, "Data mining on temporal data: a visual approach and its clinical application to hemodialysis," *Journal of Visual Languages & Computing*, vol. 14, no. 6, pp. 591–620, 2003.

[91] B. Shneiderman, "Dynamic queries for visual information seeking," *IEEE software*, vol. 11, no. 6, pp. 70–77, 1994.

[92] H. Hochheiser and B. Shneiderman, "Dynamic query tools for time series data sets: timebox widgets for interactive exploration," *Information Visualization*, vol. 3, no. 1, pp. 1–18, 2004.

[93] P. Buono, C. Plaisant, A. Simeone, A. Aris, G. Shmueli, and W. Jank, "Similarity-based forecasting with simultaneous previews: A river plot interface for time series forecasting," in *Information Visualization, 2007. IV'07. 11th International Conference*, pp. 191–196, IEEE, 2007.

[94] T. Saito, H. N. Miyamura, M. Yamamoto, H. Saito, Y. Hoshiya, and T. Kaseda, "Two-tone pseudo coloring: Compact visualization for one-dimensional data," in *Information Visualization, 2005. INFOVIS 2005. IEEE Symposium on*, pp. 173–180, IEEE, 2005.

[95] C. Perin, F. Vernier, and J.-D. Fekete, "Interactive horizon graphs: improving the compact visualization of multiple time series," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3217–3226, ACM, 2013.

[96] P. Federico, S. Hoffmann, A. Rind, W. Aigner, and S. Miksch, "Qualizon graphs: Space-efficient time-series visualization with qualitative abstractions," in *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, pp. 273–280, ACM, 2014.

[97] C. Shi, W. Cui, S. Liu, P. Xu, W. Chen, and H. Qu, "Rankexplorer: Visualization of ranking changes in large time series data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 12, pp. 2669–2678, 2012.

[98] L. Byron and M. Wattenberg, "Stacked graphs–geometry & aesthetics," *IEEE transactions on visualization and computer graphics*, vol. 14, no. 6, 2008.

[99] S. Havre, E. Hetzler, P. Whitney, and L. Nowell, "Themeriver: Visualizing thematic changes in large document collections," *IEEE transactions on visualization and computer graphics*, vol. 8, no. 1, pp. 9–20, 2002.

[100] C. Perin, R. Vuillemot, and J.-D. Fekete, "A table!: Improving temporal navigation in soccer ranking tables," in *Proceedings of the SIGCHI conference on human factors in computing systems*, pp. 887–896, ACM, 2014.

[101] C. Tominski, "Event-based concepts for user-driven visualization," *Information Visualization*, vol. 10, no. 1, pp. 65–81, 2011.

[102] Y. Shahar and M. A. Musen, "Knowledge-based temporal abstraction in clinical domains," *Artificial intelligence in medicine*, vol. 8, no. 3, pp. 267–298, 1996.

[103] R. Bade, S. Schlechtweg, and S. Miksch, "Connecting time-oriented data and information to a coherent interactive visualization," in *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 105–112, ACM, 2004.

[104] Y. Shahar, D. Goren-Bar, D. Boaz, and G. Tahan, "Distributed, intelligent, interactive visualization and exploration of time-oriented clinical data and their abstractions," *Artificial intelligence in medicine*, vol. 38, no. 2, pp. 115–135, 2006.

[105] D. Klimov, Y. Shahar, and M. Taieb-Maimon, "Intelligent visualization and exploration of time-oriented data of multiple patients," *Artificial intelligence in medicine*, vol. 49, no. 1, pp. 11–31, 2010.

[106] D. Klimov, Y. Shahar, and M. Taieb-Maimon, "Intelligent selection and retrieval of multiple time-oriented records," *Journal of Intelligent Information Systems*, vol. 35, no. 2, pp. 261–300, 2010.

[107] J. Zhao, F. Chevalier, E. Pietriga, and R. Balakrishnan, "Exploratory analysis of time-series with chronolenses," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 12, pp. 2422–2431, 2011.

[108] M. Burch, F. Beck, and S. Diehl, "Timeline trees: visualizing sequences of transactions in information hierarchies," in *Proceedings of the working conference on Advanced visual interfaces*, pp. 75–82, ACM, 2008.

[109] "Limesurvey: the online survey tool - open source surveys," https://www.limesurvey.org/.

[110] L. Lins, J. T. Klosowski, and C. Scheidegger, "Nanocubes for real-time exploration of spatiotemporal datasets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2456–2465, 2013.

[111] M. Liu, J. Shi, K. Cao, J. Zhu, and S. Liu, "Analyzing the training processes of deep generative models," *IEEE transactions on visualization and computer graphics*, vol. 24, no. 1, pp. 77–87, 2018.

[112] C. R. Palmer and C. Faloutsos, "Density biased sampling: An improved method for data mining and clustering," in *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, pp. 82–92, 2000.

[113] R. Kincaid, "Signallens: Focus+ context applied to electronic time series," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 900–907, 2010.

[114] J. Zhao, F. Chevalier, and R. Balakrishnan, "Kronominer: using multi-foci navigation for the visual exploration of time-series data," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1737–1746, ACM, 2011.

[115] P. McLachlan, T. Munzner, E. Koutsofios, and S. North, "Liverac: interactive visual exploration of system management time-series data," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 1483–1492, ACM, 2008.

[116] J. J. Van Wijk and E. R. Van Selow, "Cluster and calendar based visualization of time series data," in *Information Visualization, 1999.(Info Vis' 99) Proceedings. 1999 IEEE Symposium on*, pp. 4–9, IEEE, 1999.