Interaction+: Interaction Enhancement for Web-based Visualizations

Min Lu[1]* Jie Liang[2]† Yu Zhang[1]‡ Guozheng Li[1]§ Siming Chen[1]¶ Zongru Li[1]‖ Xiaoru Yuan[13]**

1) Key Laboratory of Machine Perception (Ministry of Education), and School of EECS, Peking University, China
2) School of Engineer and Information Technology, The University of Technology, Sydney, Australia
3) Beijing Engineering Research Center of Virtual Simulation and Visualization, China

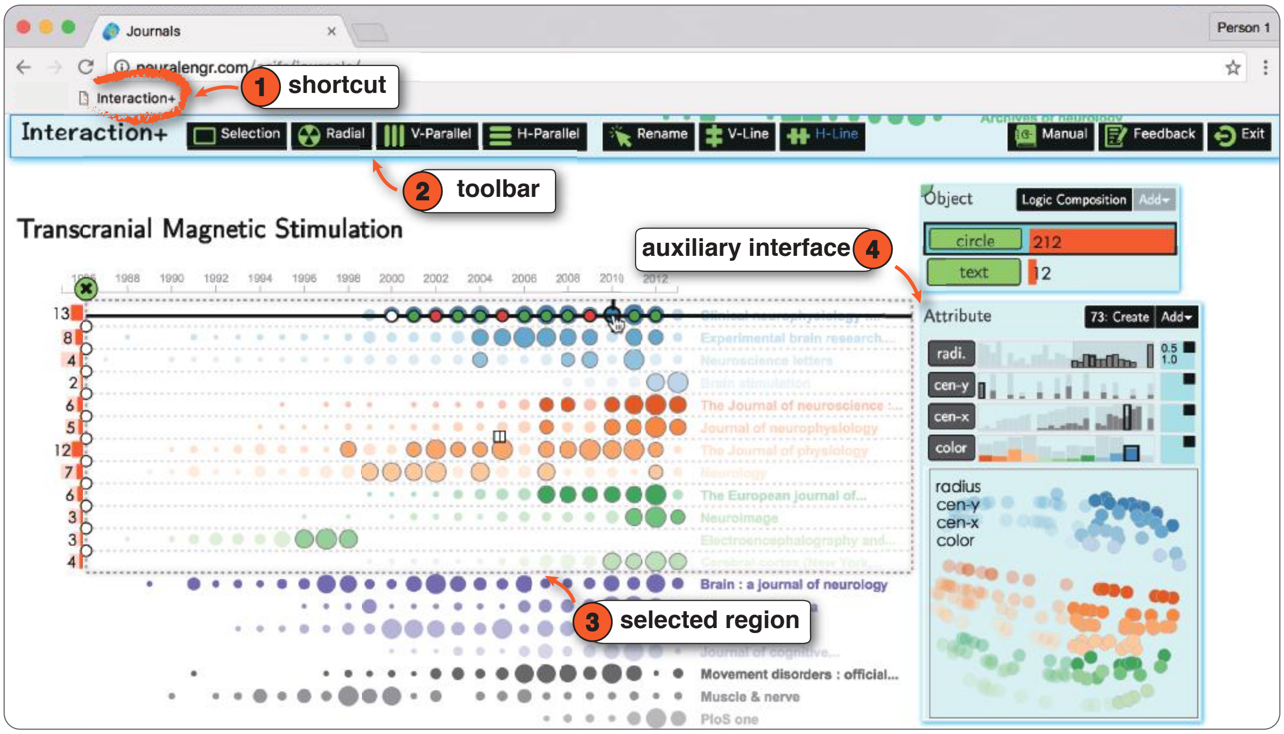


Figure 1: Usage of *Interaction+*: 1. with a click on the shortcut, *Interaction+* is applied in an existing visualization; 2. *Interaction+*'s toolbar is added on the top; 3. the user brushes a region of interest on the webpage; 4. *Interaction+* adds an auxiliary visualization of the extracted visual information; interactions can be performed either in the auxiliary interface (e.g., filter by radius) or the original visualization (e.g., hover to compare).

Abstract

In this work, we present *Interaction+*, a tool that enhances the interactive capability of existing web-based visualizations. Different from the toolkits for authoring interactions during the visualization construction, *Interaction+* takes existing visualizations as input, analyzes the visual objects, and provides users with a suite of interactions to facilitate the visual exploration, including selection, aggregation, arrangement, comparison, filtering, and annotation. Without accessing the underlying data or process how the visualization is constructed, *Interaction+* is application-independent and can be employed in various visualizations on the web. We demonstrate its usage in two scenarios and evaluate its effectiveness with a qualitative user study.

Index Terms: H.5.2 [Information Interfaces and Presentation]: User Interfaces—Graphical user interface

*e-mail: min.lu@pku.edu.cn
†e-mail: jie.liang@uts.edu.au
‡e-mail: yuzhang94@pku.edu.cn
§e-mail: guozheng.li@pku.edu.cn
¶e-mail: siming.chen@pku.edu.cn
‖e-mail: zongru.li@pku.edu.cn
**e-mail: xiaoru.yuan@pku.edu.cn

1 Introduction

With the flourish of web-based visualization toolkits, more and more people deliver information by visualizations on the web. Journalists publish news with embedded visualizations to tell the story. Statisticians attach visualizations of data in reports to express their observations.

Interactions are essential for efficient visual analytics [17] [19]. However, many online visualizations provide insufficient interactions, sometimes even no interactions. Interactive graphical editors, such as Adobe Illustrator and iVisDesigner [20] support users to generate eye-catching visualizations. However, they do not allow users to customize interactions in visualizations. Powerful visualization toolkits, such as D3 [6] and InfoVisTool [1], provide the application programming interface for interactions. However, programming with the low-level event handlers makes it laborious for visualization developers to develop interactions. Although some mixin libraries, like VisDock [9], have been designed to facilitate the programming of interactions, the requirement of programming expertise still sets the bar of interactive visualizations at a high level.

Based on these observations, we aim to enhance the interactivity of existing visualizations on the web in this work. To add interactions to a visualization which has been already created, one of the biggest challenges is the absence of visualizing context, i.e., there is no guarantee of the accessibility of the underlying data and code. However, most existing toolkits for authoring interactions during the visualization construction require the awareness of the visualization pipeline or the back-end data.

Different from the conventional constructing approach, we propose a novel method *Interaction+* which constructs interactions the other way around. Given an existing visualization, *Interaction+* takes its data and visualization process as a black-box, i.e., without the knowledge of how the visualization is created. Specifically, *Interaction+* extracts the visual information (e.g., visual objects and visual mappings, etc.) from the visualization and provides a suite of interactions driven by the information. Users' understanding of the visualization is leveraged to settle down the meaning of those interactions in a certain context. In return, *Interaction+* supports users to make the best of existing visualizations with the following benefits: **General** Without touching the underlying code and data, *Interaction+* can be applied to a wide range of visualizations on the web; **Light-weight** As an auxiliary enhancement, *Interaction+* provides an instant shortcut to a suite of interactions without much effort; **Explicit** *Interaction+* externalizes the visual information of existing visualizations and provides interactions which facilitate the visual reasoning which otherwise would be conducted mentally; **Flexible** *Interaction+* provides a suite of interactions which support users to explore the visualizations with different demands.

The contributions of this work are:

- An interaction enhancement approach, which augments the interactive capability of existing visualizations on the web.
- A prototype *Interaction+*, which can be applied in a variety of existing web-based visualizations.

The rest of the paper is organized as follows. We discuss the related work in Section 2. Section 3 gives an overview of *Interaction+*. Section 4 presents the design details. We demonstrate its usage in Section 5 and report evaluation result in Section 6. In Section 7, we discuss the limitations and future work. In Section 8, we present the conclusion.

2 Related Work

We discuss the work related to *Interaction+* in three topics: general interactions that facilitate visual exploration, toolkits that build up web-based visualizations, and surface enhancement techniques that take the surface of interface as analysis basis.

2.1 Interactions

The importance of interactions in efficient visual analysis is widely recognized [19] [30]. There are various categorizations of low level interactions in the InfoVis field [4] [28]. For example, Roth [21] develops a functional taxonomy of interaction primitives for map-based visualization. Besides the low level categorization, Brehmer and Munzner [7] summarise a multi-level typology model which expresses complex tasks as sequences of simpler tasks. They distinguish *what*, *how*, and *why* a task is performed. In this work, *Interaction+* concerns how to assist users to perform tasks in existing visualizations with a suite of interactions.

Liu et al. [17] discuss the functionalities of interactions in three-fold, i.e., to enable external anchoring, information foraging, and cognitive offloading. Simkin and Hastie [26] describes the elementary perceptual processes that viewers use when extracting and comparing values, including anchoring scanning, projection, superimposition, detection, etc. *Interaction+* aids user to conduct efficient visual reasoning in web-based visualizations with add-on interactions, such as explicit comparing, quick counting, etc. Heer and Shneiderman [14] summarise the Interactive dynamics which contribute to successful analytic dialogues in three levels, i.e., view specification, view manipulation, and process & provenance. The interactions provided by *Interaction+* cover all the three levels, such as filtering to specify the view, selection to manipulate the view, and annotation to preserve analytic provenance.

2.2 Web-based Visualization Toolkits

Various visualization toolkits have been proposed to facilitate the construction of web-based visualizations. Among those toolkits, one major effort goes to developing the interactive environment for visualization construction. Lyra [22] allows users to author visual designs by drag-and-drop interactions. iVisDesigner [20] provides the interactive design of visualizations based on conceptual modularity. This kind of toolkit mainly focuses on visualization customization rather than interaction development. The other type is the client-side visualization libraries [1] [5] [6]. ProtoVis [5] defines a simple grammar of the graphical primitives and allows visualization designers to focus on the visual configuration. D3 [6] directly maps data attributes to elements and dynamically transforms them by updating the data. Reactive Vega [24] supports the declarative design of interactive visualization where the input data, scene graph elements, and interaction events are treated as first-class streaming data sources. Furthermore, Vega-lite [23] is a high-level grammar that enables rapid specification of interactive data visualizations, with which classical layouts require typically 1-2 lines of codes. Those libraries provide event handling APIs for interactions. Coding expertise is required to enable the interactivity of visualizations. VisDock [9] provides a set of cross-cutting interactions for SVG-based visualizations. As an external JavaScript mixin library, it still requires coding, although at minimum. *Interaction+* is intended to make the interactions of visualizations more accessible to the casual end users who don't know how to program. Instead of invoking the interaction APIs during visualization construction, *Interaction+* builds up the interactions after the visualization construction, i.e., to activate the existing web-based visualizations.

2.3 Surface Enhancement

One type of research effort goes to developing enhancement tools independent from applications by only scratching the surface of systems or interfaces. One of the earliest works is the observational attachment proposed by Olsen et al. [18]. Observational attachment operates primarily by observing and manipulating the surface representations of applications. Later, Dixon and Fogarty [11] provide three advanced interactions (target-aware pointing techniques, etc.) in a wide range of application interfaces by analyzing the pixels of interface.

In the field of visualization, some researches enhance the created visualizations to improve their legibility and interactivity. ReVision [25] extracts the underlying data from bitmaps and suggests more effective visual designs. Kong and Agrawala [15] present a technique that covers five graphical overlays to existing chart bitmaps to facilitate reading tasks. Brosz et al. [8] propose an interface which enables the graphical transformation from one shape to another, such as a bar chart transmogrified to a pie chart. Steinberger et al. [27] introduce the context-preserving visual linking to connect related elements across heterogeneous visualization views, which is independent from applications because of the image-based analysis.

Instead of enhancing the bitmap images with static transformations or overlays, *Interaction+* enhances the visualizations with dynamic visual exploration. One highly related work is the D3 restyle proposed by Harper et al. [12], which allows users to restyle the D3 visualizations without examining the underlying codes. The data of every visual object is retrieved from the _data_ property. By changing the visual mapping from data to visual object, the style of D3 visualizations is modified. Compared to their work, *Interaction+* does not require access to the underlying data, and therefore it is not limited to visualizations created using D3. *Interaction+* is applicable to general SVG-based visualizations including infographics. Meanwhile, different from restyling the visualization, *Interaction+* emphasizes on augmenting the interactions.

3 Overview

In this section, we have an overview of *Interaction+*, including the motivating scenario that drives the idea of *Interaction+*, the workflow of *Interaction+* and the data model of web-based visualizations.

3.1 Motivating Scenario

Helen, a policy analyst, is evaluating the budget proposal of the United States in recent years. She surfs the Internet and finds that there is a report in *The New York Times* [2], about President Obama's budget proposal in 2013 [1]. Instead of classical news, the report employs visualizations to deliver the information (Figure 13). It visually represents every project in the proposal as a circle and draws them together. Helen finds she can check the information of an individual project by hovering over the circle. However, she hopes to dig deeper and gain a sharper understanding of the proposal, such as to know the top 100 projects with the maximum spending, etc. She is disappointed that there is no more interactions to support these tasks. Instead of putting the visualization aside, Helen wants to make full use of it. She applies *Interaction+* in the webpage to have a discourse with the circles on the webpage. With *Interaction+*, she gets all the circles and their sizes. Then Helen sorts those circles by size and filters the desirable top 100. Quickly she gets the projects that she needs.

3.2 Workflow

Figure 2 illustrates the workflow of *Interaction+*. Initially, the input is a webpage with visualization. Then *Interaction+* parses the webpage's HTML Document Object Model (DOM) to extract visual information. The visual information contains visual objects and their attributes, as well as advanced knowledge about the visualization, such as the size distribution of an object group, etc. After that, *Interaction+* links and visualizes the extracted information on the webpage. With the added-on interface, users are able to perform a set of interactions with the existing visualizations, including selection, comparison, filtering, etc. Interactions are bidirectional, which can be performed either in the original visualization or in the auxiliary interface.

Figure 1 shows the interface of *Interaction+*. With an opened webpage, users can easily access the *Interaction+* service by a click on the shortcut. Then the toolbar of *Interaction+* is appended on the top. After selecting the region of interest, *Interaction+* adds an auxiliary visual interface on the webpage, which visualizes the extracted visual information. Then users can perform a variety of interactions in both the existing visualization and the auxiliary interface, such as hovering in the original visualization to compare visual objects, cross filtering in the auxiliary interface, etc. Users can exit *Interaction+* by clicking the right-top exit button.

[1] http://www.nytimes.com/interactive/2012/02/13/us/politics/2013-budget-proposal-graphic.html?_r=0

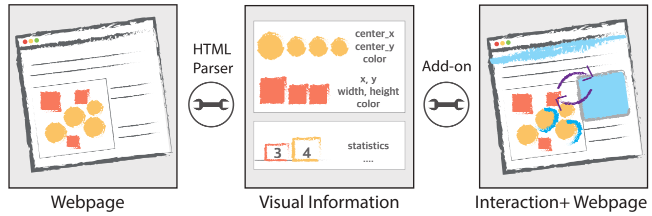


Figure 2: *Interaction+*'s Workflow: taking a webpage with existing visualizations as input, *Interaction+* extracts the visual information by parsing the HTML. Then *Interaction+* adds on a set of interactions driven by the visual information, which can be performed in the original visualization or the auxiliary interface.

3.3 Webpage with Visualization

Interaction+ takes the webpage with visualization as input and performs information extraction on it. Before jumping into detail of the method, we introduce the data model of webpages in this section, serving as the preliminary.

When a webpage is loaded, the browser creates an HTML DOM of the page. The DOM is constructed as a tree of objects. Figure 3 illustrates the DOM of a webpage with bar chart. Starting from the *document* object, each object in the webpage is represented as a node in the DOM tree, including both the displayed and not displayed ones. Specially, the DOM tree stores the visual objects which compose the visualization, such as *rect*, *text* and *line* objects of the bar chart in Figure 3. An object is delineated by a tag name and a list of attributes. For example, the *rect* object is wrapped by a tag name *rect*. Its attributes describe the information of the rectangle, such as the visual attributes *width*, *height*, *fill*, etc.

The basic idea of extracting visual information is to analyze the visual objects by parsing the DOM tree. Webpages in real world can be much more complex than the conceptual model in Figure 3. We have tackled a series of problems that we encountered during *Interaction+*'s prototyping. We will introduce them along with the design in Section 4.

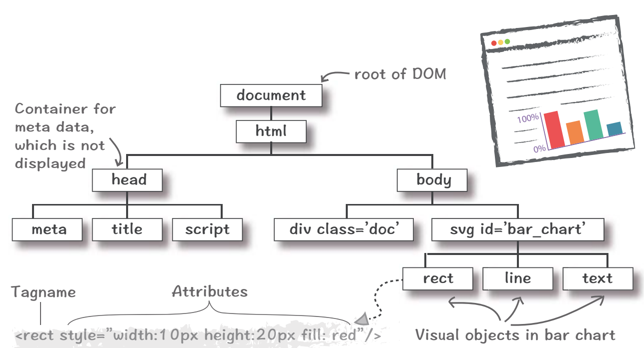


Figure 3: Illustration of DOM: DOM tree of a webpage with bar chart, note that it is not fully expanded.

4 *Interaction+*

Following the multi-level typology model of visualization tasks by Brehmer and Tamara [7], *Interaction+* enables users to manipulate the existing visualizations via selection, aggregation, arrangement, comparison, filtering, as well as annotation. In this section, we introduce the design of *Interaction+* in detail.

4.1 Selection

Selection is to mark something as interesting, which is one of the most essential interactions [30]. *Interaction+* supports selection on the webpage. As Figure 4 shows, users can brush a rectangular region of interest directly on the webpage. Then *Interaction+* automatically extracts the underlying visual objects and classifies them into groups, and then visually summarises the visual objects in the object panel. By default, the objects are grouped by tag names. A group of objects is represented by a label and a population bar giving the number of objects in the group. A click on the label highlights the corresponding group of objects in the visualization. The selected region can be removed by clicking the top-left delete button.

Specifically, in the object extraction step, *Interaction+* traces down the DOM tree to retrieve all the visual objects within the selected region. Some webpages use the in-line frame (i.e., $<iframe>$ object) to embed an independent webpage within a webpage. *Interaction+* needs to keep track of the global position of the frame to correctly recover the global position of objects within the frame. Meanwhile, we define a pruning set P_set of objects which are not possible containers for a visualization, e.g., $<script>$, $<audio>$, etc. Tracing is not conducted for the pruned objects and their children. Pseudocode 1 describes the algorithm.

Algorithm 1 Visual Object Extraction Algorithm

```
procedure EXTRACTOBJ(rect)                    ▷ input the selected rect
    visual_objects ← [ ]                       ▷ output the with-in objects
    children ← getChildren('body')
    while len(children) > 0 do
        child ← children.pop()
        if child is not in P_set then
            candidates ← getChildren(child)
            if len(candidates) == 0 then
                visual_objects.push(child)
                continue
            end if
            shift_pos ← (0,0)
            if child is 'iframe' then
                shift_pos ← getShift(child)
            end if
            candidates ← getInRect(candidates, shift_pos, rect)
            children.append(candidates)
        end if
    end while
    return visual_objects
end procedure
```

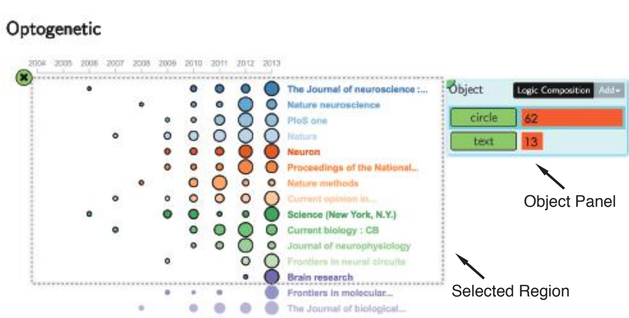


Figure 4: Selection: objects within a selected region are selected and summarized in the object panel. The corresponding objects in the original visualization are highlighted with a black border after clicking the group label in the object panel.

Meanwhile, *Interaction+* provides a special selection in the textual context on the webpage, where the users' understanding is leveraged to give semantic meaning to objects. As Figure 5 shows, users can brush a word on the webpage and use the word to rename a group of the detected objects. To rename a group, users can also directly type a name after double-clicking the label.

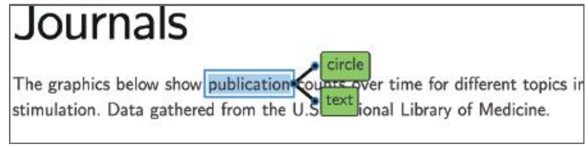


Figure 5: Selection in Textual Context: brush a word in the context to rename an extracted object group.

4.2 Aggregation

In *Interaction+*, we develop mask, a novel interaction, to aggregate the visual objects according to the defined layout structure. Mask acts as a template by which the selected region is sliced into sub-regions and the number of objects is counted in each sub-region. To maintain simplicity, we implement three types of masks for tabular and radial layouts, which are widely used layout structures in visualizations. Figure 6 lists the three masks implemented in the *Interaction+* prototype, i.e., radial, V-Parallel (vertically parallel), and H-Parallel (horizontally parallel) masks. The three masks slice the selected region with different schemas.

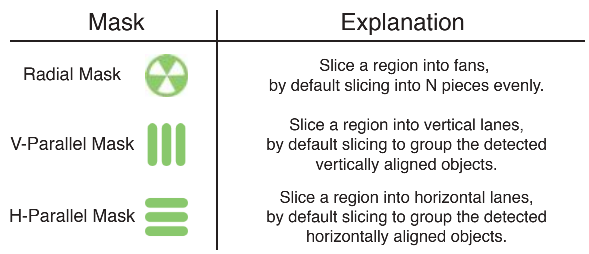


Figure 6: Three Masks: templates to aggregate the visual objects.

In Figure 7(a), the H-Parallel mask is covered over a visualization. Strictly speaking, vertically aligned objects should lie exactly on the same vertical line. However, in many cases, the objects that we perceive as vertically aligned may have little biases in the horizontal direction. To improve the generality of mask, our detected alignment is relaxed from the strict definition. For example, in Figure 7(a), stars are considered horizontal-aligned if their vertical distances are within a certain bias. Figure 7(b) illustrates the slicing procedure. Firstly, *Interaction+* extracts the boundary boxes of objects. Then objects are grouped by density clustering algorithm which only counts the distance along the orthogonal direction to the mask, e.g., vertical distance if using H-Parallel Mask. The centroids detected in the clustering are used to represent the position of each group. Then, slicing lines are computed as the perpendicular bisectors between each pair of adjacent centroids. After slicing, *Interaction+* counts the objects in each lane and shows the numbers as bars on the left in Figure 7(a). The bars will be dynamically updated when the objects are filtered (will be introduced in Section 4.5).

Figure 8(a) illustrates the Radial mask, which counts the stars in each sector and visualizes the number as surrounding arcs. Users are not required to place the mask precisely. When covered over objects, Radial mask is automatically calibrated to the center of the objects' boundary box (Figure 8(b)).

Moreover, *Interaction+* allows users to manually adjust the mask slicing. As Figure 7(a) shows, users can decrease/increase the slicing granularity by clicking the left/right Granularity Changer button in the center to observe in multiple levels. Users can also relocate a slicing line by dragging its handler.

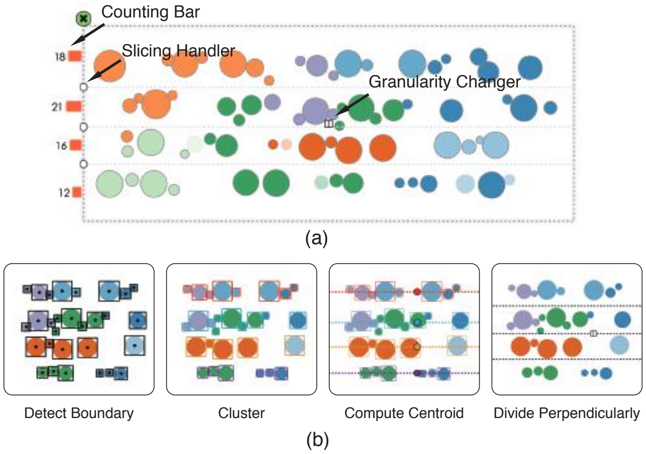


Figure 7: Aggregation by H-Parallel Mask: (a) H-Parallel mask which slices the visualization horizontally and counts objects in horizontal lanes; (b) Slicing procedure of H-Parallel mask.

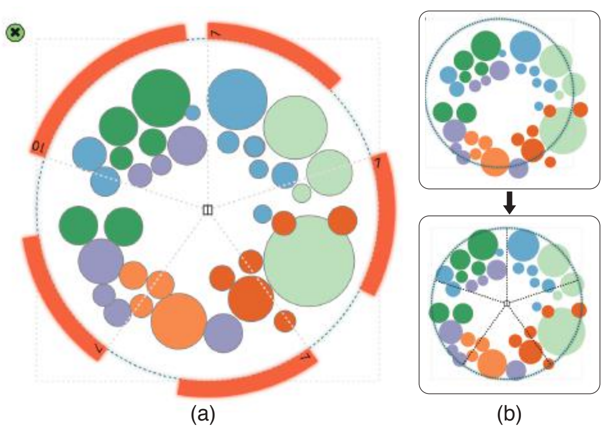


Figure 8: Aggregation by Radial Mask: (a) Radial mask which slices the visualization radially and counts objects in sectors; (b) Radial mask calibration.

4.3 Arrangement

Interaction+ allows users to use projection to arrange the visual objects in an auxiliary view with a different spatial layout from the original visualization. Visualization is to encode data to visual attributes of objects. To assist the visual perception, *Interaction+* parses the HTML of extracted objects to get the quantified visual information, such as the *width*, *height*, *fill* attributes of the *rect* object in Figure 3.

As Figure 9(a) shows, *Interaction+* visualizes the distribution of objects' visual attributes in the attribute panel. Users can select an object's group and examine the distribution of these visual objects over various attributes. The label gives the name of the attribute, which by default is the name detected from the HTML. Labels with obscure names are automatically replaced with intelligible names, such as *color* for *fill*, $center_x$ for *cx*, etc. Meanwhile, users can rename the attribute after double-clicking the label. The histogram shows the frequencies of objects in a certain range of the attribute. The histogram can be expanded to the one with ticks for more details (e.g., the *color* attribute in Figure 9(a)). Users can delete an unwanted attribute by clicking the delete button which appears when hovering over the name label.

Interaction+ supports users to arrange the visual objects by the joint distribution of objects across multiple visual attributes. By directly dragging attribute labels into a new row, users can create the projection of multiple attributes. Specifically, a scatter plot is created if two attributes are joined (Figure 9(b)) and MDS (Multidimensional Scaling [16]) projection if there are more attributes (Figure 9(c)).

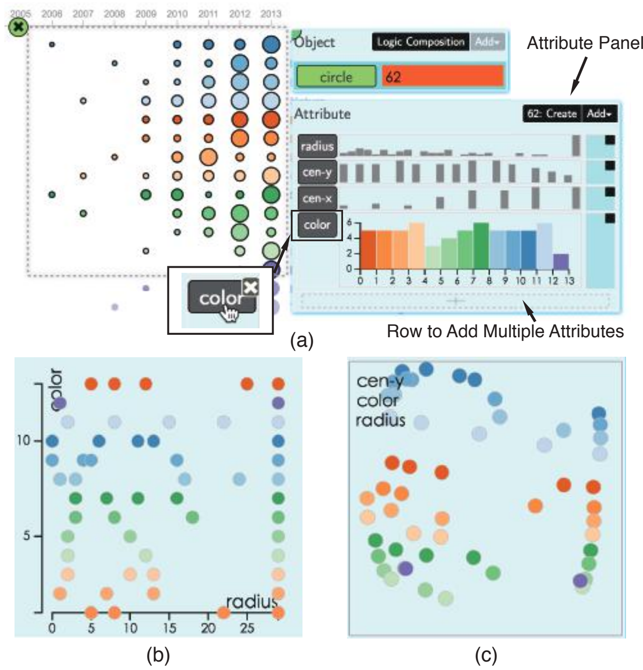


Figure 9: Arrangement: (a) the histogram shows the distribution of each visual attribute for the selected group of objects; (b) visual objects are plotted in a scatter plot which consists of two attributes; (c) visual objects are drawn in MDS projection of multiple attributes.

4.4 Comparison

Interaction+ supports users to compare the visual objects in the visualization. On one hand, *Interaction+* supports users to compare an individual object globally. When hovering on a single object, its corresponding visual information is highlighted in the attribute panel. It supports users to spot a specific visual object in the whole distributions. On the other hand, *Interaction+* provides a special comparison interaction, which allows users to do local comparison within a collection of objects. As Figure 10 shows, *Interaction+* provides reference lines to directly facilitate the elementary perceptual processes of anchoring and comparison [26]. Currently, *Interaction+* helps users scan the objects which line up horizontally or vertically. Taking the hovered object as the observed object, the reference line helps to read values, such as the anchoring on axis in left part of Figure 10(a). Meanwhile, it facilitates the comparison to others in the same horizontal or vertical line. The difference is explicitly encoded by a circular marker, i.e., green if larger, red if smaller, and white if equal.

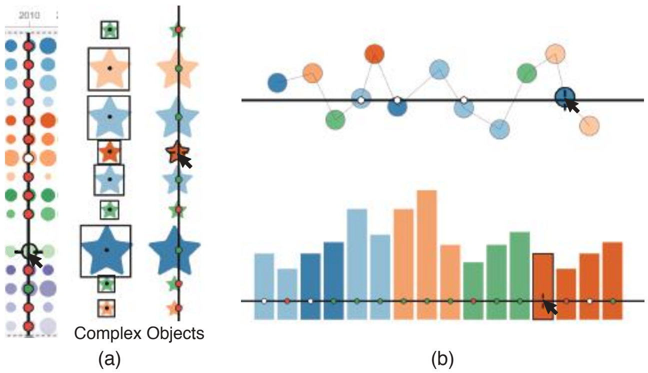


Figure 10: Comparison with Reference Line: (a) vertical line to compare objects which line up vertically, which also can handle complex shapes; (b) horizontal line to compare objects which line up horizontally.

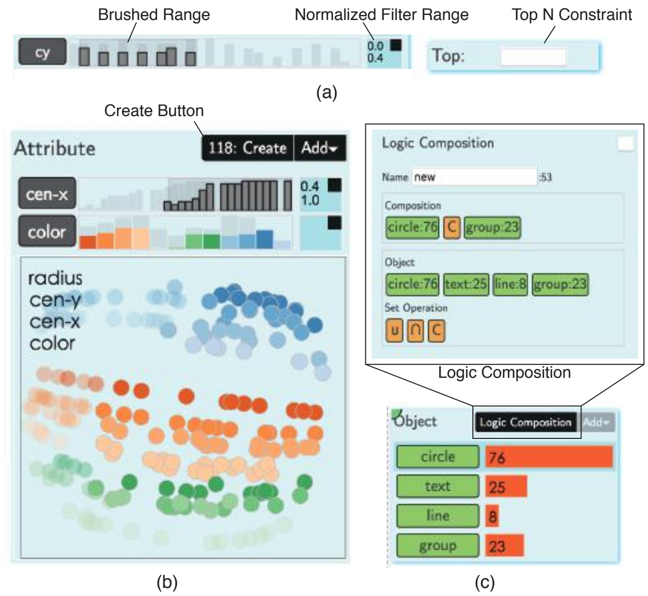


Figure 11: Filtering: (a) either brushing the range of interest or defining the top N constraint; (b) multiple filtering criteria combined via cross-filter mechanism; (c) group composition by logic operation.

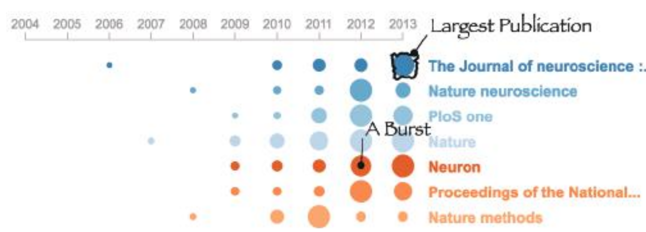


Figure 12: Annotation: textual tag connected to the visual objects, with border as a highlight option.

4.5 Filtering

Filter is an essential interaction, which allows users to peel off the irrelevant objects and focus on relevant ones. *Interaction+* enables users to conduct dynamic query [3], which makes the existing visualizations on web more responsive and live. Explicit filtering by *Interaction+* facilitates the object retrieving tasks, which would otherwise be performed mentally and cost users much effort.

As Figure 11(a) shows, users can either brush a range of interest on a single attribute or directly lasso the objects in scatter plot and MDS projection. The normalized selected range is given in the right to indicate the filtering quantitatively. Moreover, users can retrieve objects with the top N values in an attribute, such as the top 10 objects with the maximum radius. As Figure 11(b) shows, multiple filtering settings are combined via cross-filter mechanism [29]. Whiling filtering, objects are dynamically highlighted in the existing visualization and updated in *Create* button. By clicking the button, users can save the filtered objects as a group so that it can be explored further. In a higher level, an object group can be derived by applying logic operations, i.e., union, intersection, and complementary operations to the originally detected groups. Figure 11(c) shows the interface of logic composition, where users can define the operators and operands by drag-and-drop.

4.6 Annotation

Annotation is one of the most commonly used approaches to facilitate the introduction or sharing of visualizations. *Interaction+* supports users to add additional textual annotations to the visualizations. After double-clicking on the visual object of interest, users can type words to make a label. Figure 12 shows annotations with different styles. Specifically, border with hand-drawing style is provided as an option to highlight the annotated visual object. Moreover, users can manually drag the label around to adjust its location.

5 Usage Scenarios

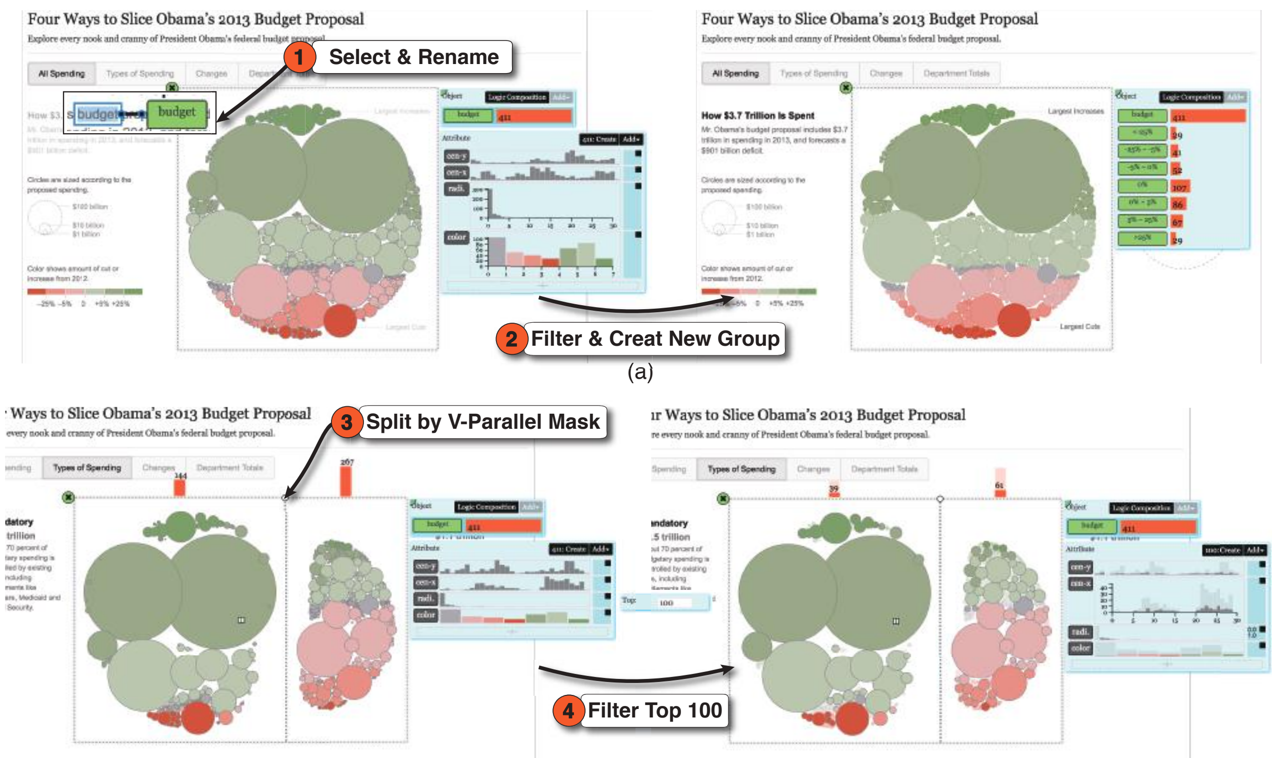
In this section, we demonstrate the two usage scenarios of *Interaction+*. The first one is our motivating scenario, to apply *Interaction+* in *The New York Times*. The second one is to give life to infographics using *Interaction+*. Interaction+ can be accessed at http://vis.pku.edu.cn/interaction+.

5.1 *Interaction+* Visualization News

As introduced in Section 3.1, Helen wants to dig a visualization published in *The New York Times* for more information. The visualization depicts the President Obama's federal budget proposals in 2013 (Figure 13). In the visualization, each circle represents a budget project, whose size encodes the spending. Its color indicates the amount of increase-cut compared to 2012, green for more money and red for less.

With *Interaction+*, Helen initially brushes a rectangular region to select the whole visualization. Then *Interaction+* automatically detects 411 circles in the visualization. After reading the textual context of the visualization, Helen brushes the word *budget* and assigns it as the name of each circle. Now Helen learns that there are 411 budget projects proposed in 2013. To dig deeper, Helen clicks the budget group to examine its attributes in detail. With the understanding of the visual encoding, Helen expands the meaningful attributes *radius* and *color* (left part in Figure 13(a)). In the *radius* histogram, Helen learns that distribution of spending is a classical long tail distribution. In the *color* histogram, Helen can see the frequency of different proposal categories in different cut-increase levels. Helen finds that the gray category, i.e., the budgets with stable spendings in 2012 and 2013, makes up the largest population, which would be overlooked in the original visualization because of the small size of the gray circles. By brushing the color attribute, Helen filters the projects in certain increase-cut level and creates a new group with the filtered projects. In the right part of Figure 13(a), categories of projects are listed with population bars, which make it easy to compare. Helen can examine each of them and conduct the exploration further.

Helen is also interested in the difference between mandatory



(a)

(b)

Figure 13: Usage Scenario 1 *Interaction+* Applied in *The New York Times*: (a) *Interaction+* is laid over the whole visualization, which extracts 411 budget projects; histograms of visual attributes (i.e., color and radius) indicate projects' distributions over increase-cut levels and spendings respectively; Projects are filtered by different increase-cut levels, each of which is created as a new group for further examination. (b) changed to *types of spending* layout, *Interaction+* divides proposals into mandatory and discretionary ones by V-Parallel mask. *Interaction+* does quick count on the two types of proposals, i.e., 144 mandatory proposals and 267 discretionary ones. Filtering the top 100 spending projects, 61 of them are discretionary and 39 are mandatory.

projects and discretionary ones. So she makes changes to the visualization in the *types of spending* layout. Helen covers the V-Parallel mask over the visualization. She adjusts the mask to separate projects into the mandatory and discretionary categories. In the left of Figure 13(b), she learns that the discretionary projects are almost 2 times more than the mandatory one. Furthermore, she examines how the projects are distributed in these two categories. She retrieves the proposals with the top 100 biggest spendings, i.e., sets the top constraint in *radius* to 100. The result is shown in the right part of Figure 13(b). Among the top 100 projects, there are 39 mandatory projects and 61 discretionary proposals.

This scenario showcases how *Interaction+* can support the interactions when the existing visualization has very limited interactions. *Interaction+* provides a set of interactions by which the visual process is externalized. Additionally, the auxiliary interface provided by *Interaction+*, such as the list of projects in different increase-cut categories (the right part of Figure 13(a)), also serves as a visualization which shows the data from another view.

5.2 *Interaction+* Infographics

Infographics are vivid visual representations of data. John, a teacher, crafts an infographic with Adobe Illustrator to present the information on the four groups of students in his classroom, each of which is encoded in one color [2]. As Figure 14 shows, the infographic depicts the students from multiple facets, including their geographical distribution, performance over courses, health status, and attendance. Although the inforgraphic is very eye-catching, lacking interactions holds readers back from exploring the data further. So John wants to exploit *Interaction+* to give life to the infographic.

Figure 14(a) is the map part which visualizes the students' distribution over the US. With *Interaction+*, John is able to make selection on the map. For example, in Figure 14(a), John brushes students over the whole US and it is quickly counted that there are 38 students in total and its distribution over groups. By color filtering, John checks the geospatial distribution of a specific group.

The infographic also gives a calendar-based view to show students' attendance in the semester, as Figure 14(b) shows. With the filtering function, John is able to check up the days with different attendances. For example, John filters the days in March with the top largest/smallest width value, i.e., the highest/lowest attendance. To better memorize his exploration, John makes annotations on the filtered result as Figure 14(b) shows.

The inforgraphic enumerates the health status of students one by one in three levels (Figure 14(b)). With the H-Parallel Mask in *Interaction+*, John is able to sum up the students in each level quickly.

Figure 14(d) depicts the scores of different courses. John applies the horizontal reference line in *Interaction+* to make explicit comparison among scores. In Figure 14(d), John takes the *Art* score from *Apple Group* as reference and compares others to it.

In this scenario, we demonstrate how *Interaction+* gives life to infographics. With the infographics produced by vector image editors, e.g., Adobe Illustrator, *Interaction+* provides a set of basic interactions on existing infographics without programming.

[2]Design based on Freepik http://www.freepik.com/

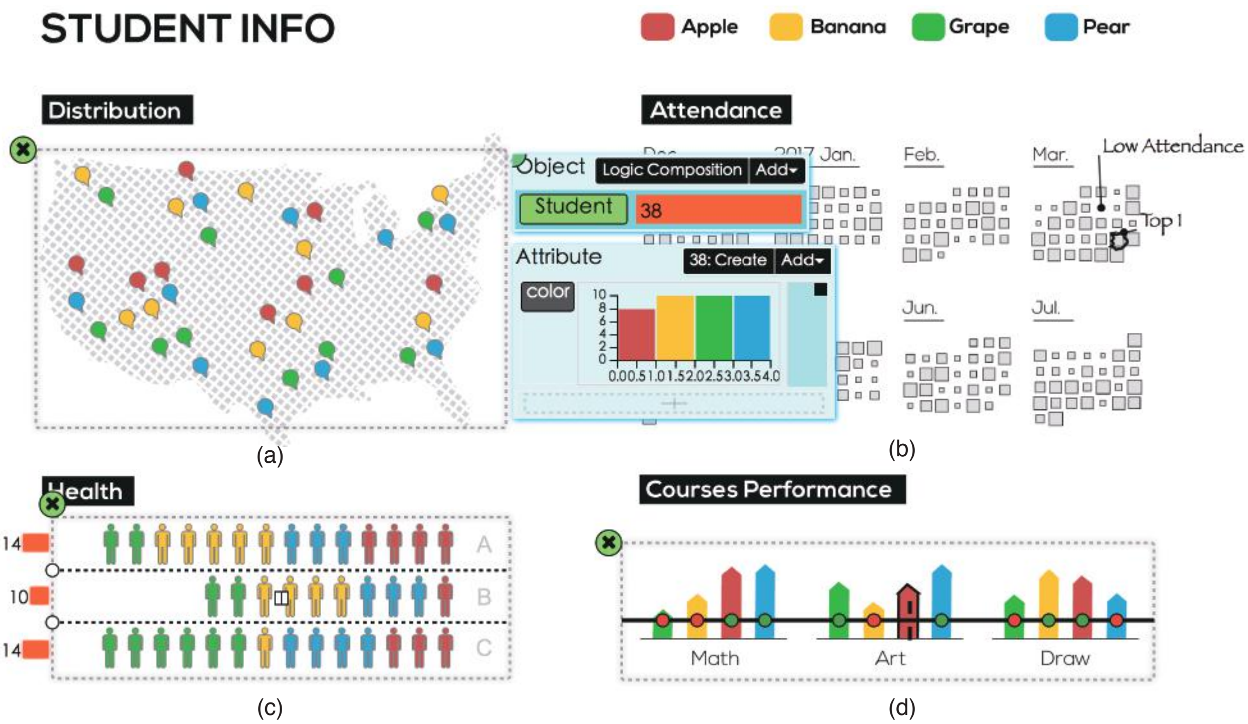


Figure 14: Usage Scenario 2 *Interaction+* Applied in the Infographic about Students: (a) *Interaction+* helps with spatial selection; (b) days with the top highest and lowest attendance are filtered by *Interaction+*, annotated with labels; (c) the H-Parallel Mask in *Interaction+* aggregates the students by three different health statuses; (d) *Interaction+* does explicit comparison among the scores.

6 Evaluation

Interaction+ is proposed to enhance the interactivity of visualizations. As discussed in Section 2, there is no similar tool or technique serving as counterpart. The result of comparison between the interactive capability with and without *Interaction+* is easy to anticipate. We performed a pre-test where we recruited two participants and asked them to perform some tasks with and without *Interaction+*, including counting, comparing, etc. The pre-test confirms that it is obviously more time-consuming, somtimes even impossible, to finish the tasks without *Interaction+*. For example, without *Interaction+*, the two participants gave up counting the circles in the budget visualization of *The New York Times* as shown in Section 5.1. Hence, we believe it is more meaningful to focus on a qualitative study to evaluate the effectiveness of *Interaction+* itself rather than a comparative study with others.

6.1 Study Design

A laboratory study was performed to systematically evaluate the effectiveness of each interaction in *Interaction+*. We recruited 14 participants in our university, including 7 undergraduate students and 7 graduate students. All of them are from the School of Computer Science and have at least a basic understanding of visualization. Four of them have expertise in visualization field. Before evaluation, we had checked the participants for color blindness.

Each participant underwent an one-on-one evaluation with a supervisor, around 45 minutes. Each participant was rewarded with a gift. At the beginning, a participant received a 15-minutes training. During the training, the supervisor introduced the operation of *Interaction+* and walked the participant through all the interactions provided by *Interaction+*. The participant could ask questions and try *Interaction+* to warm up. After the training phase, the participant is required to apply *Interaction+* to one or more websites and conduct open-ended exploration covering all interactions in *Interaction+*, including selection, aggregation, filtering, etc. A list of websites was provided for the participant to choose from. The participant could also explore any other new websites. The participant was encouraged to think aloud during the exploration. The supervisor took notes on how the participant behaved with *Interaction+*. After the exploration, the participant was asked to fill in a questionnaire. As Table 1 shows, the questionnaire surveys the opinion of *Interaction+* from both the functional and overall aspects in 7 likert agreement scale (1 for totally disagree, 7 for totally agree). In the end, the supervisor had a semi-structured interview with the participant to ask about impression, potential improvements, and general comments of the tool.

6.2 Results

The feedback from participants was positive overall. All participants showed great interest in *Interaction+* and felt excited being able to apply it to different visualizations (Q12). Averagely, each participant applied *Interaction+* in 3 webpages from the provided list. Some participants even actively sought other new webpages. One participant commented, "*Interaction+* provides a shortcut to the interactions of visualization and it should be used widely." Other participant said, "*Interaction+* provides an active and quantitative way to understand the static visualizations on the web." It is generally easy for participants to use *Interaction+* (Q11). Most of participants (11/14) expressed their impression on the light-weight feature of *Interaction+*. "It takes no effort to have a try", one participant commented. Although it is generally easy to use, the feedback varies from function to function. The following is the feedback on each interaction.

Selection All participants could make selection in the visualizations easily (Q1). Counting objects after selection received lots of praise. One participant said it really alleviated his mental burden. Besides the rectangular selection, several participants sug-

Table 1: Summary of the evaluation questions and average rating in 7 likert agreement scale (1 for totally disagree, 7 for totally agree).

Category		Questions	Rating
Functional	Select	Q1: Easy to select and count the number of the objects I concerned with this tool	6.7
		Q2: Convenient to assign semantic meaning to the objects (e.g., budget proposal to circle) with the two rename functions (double click and pick up)	6.2
	Arrange	Q3: Easy to explore and understand the attribute distribution of objects	6.4
	Filter	Q4: Easy to filter the objects by attributes	6.3
		Q5: Easy to filter the top objects	6.4
	Annotate	Q6: Easy to make annotation	6.4
	Aggregate	Q7: Easy to apply the mask on the visualization to fit in the layout of visualization	5.9
		Q8: Easy to understand the aggregation result displayed with histogram in the mask function	6.5
		Q9: Easy to divide visual objects into different groups on demand by adjusting the mask lines	5.9
	Compare	Q10: Easy to use the vertical/horizontal line comparison function for size comparison	6.0
Overall		Q11: Easy to interact visualization with the tool in general	6.4
		Q12: This tool could be applied in various visualizations online	6.4

gested more flexible selection, e.g., lasso. Some participants complained that the selection window is unchangeable after a selection is created. Most participants were satisfied with the selection in textual context (Q2). A participant suggested that *Interaction+* should detect and rename objects automatically.

Arrangement With the basic understanding of visualization, we found all participants could easily understand the histogram, scatter plot as well as MDS (Q3). They all could create the plot via drag-and-drop easily.

Filtering All participants had no difficulty in filtering (Q4, Q5). Some of them considered filtering as the highlight of *Interaction+*. The only one participant who held slightly negative opinion about filtering criticized that the extent of the brushing sometimes was too weak to recognize which one was selected.

Annotation All participants could easily add an annotation and drag it around as they liked (Q6). Several participants said marking helped them memorize the important. One participant advised that *Interaction+* should also provide the annotation function for object group.

Aggregation Compared with other functions, the utility of aggregation received a lower approval (Q7, Q9). Participants had no difficulty in understanding the aggregation (Q8). However, most of the participants identified the limitation of current three masks and suggested more masks for other classical visualizations, such as graph. One participant suggested to show a ghost mask during dragging the selection region, which would help with the definition of selection region. Meanwhile, another participant suggested to upgrade the dividing lines to polylines controlled by anchors.

Comparison Almost all the participants appreciated the explicit comparison in *Interaction+*. Some of them experienced a perceivable lag when comparing visual objects in large scale.

7 Discussion

Interaction+ can be successfully applied to a variety of existing visualizations on the web. In this section, we first discuss it in the context of activating the interactions and also identify limitations that imply the potential future research efforts.

Activating Web-based Visualizations *Interaction+* adds auxiliary interactions on the existing visualizations, which is one of the approaches to activate the visualizations. Table 2 summarises the representative toolkits which can make web-based visualizations interactive. Most of the existing toolkits (e.g., VisDock [9], Vega-Lite [23], etc.) are intended for the client programmers, who develop the interactions during the visualization construction by programming. In this manner, interactions are often customized case by case. The other type of activating toolkits are those designed for the casual end-users, who read and interact with visualizations using the non-programming tools. *Interaction+* and *D3 Restyle*[12] are this kind of tools. In this manner, interactions are often generally applicable, such as *D3 Restyle* for visualizations created by D3. The biggest difference between *Interaction+* and all existing activating toolkits, including the *D3 Restyle*, is that *Interaction+* does not access or parse the underlying data. Untouching the data benefits *Interaction+* in activating a border range of visual graphics including the infographics, as demonstrated in Section 5.2.

Table 2: Visualization Activating Toolkits

Toolkits	Targeted User	Access
VisDock	Programmer	Data
Vega-lite	Programmer	Data
D3 Restyle	Casual End-User	Data Parsed from D3 + Visual Objects
Interaction+	Casual End-User	Visual Objects

Connection to Data Although there is no requirement of knowing the underlying visual mapping from data to visual objects, in some cases, without any connection to the data, it would be hard for users to perform certain analysis tasks. *D3 Restyle* [12] recovers the mapping from data to visual objects in D3 visualizations, which inspires us to enhance *Interaction+* with data hints in the future work. Web-based visualizations are often embedded in the context which probably provides explanation about the visualizations. Now *Interaction+* provides the simple semantic renaming function. But it would be more powerful to leverage users' understanding of visualizations to recover the data-visual-mapping from the context and automatically rename objects semantically.

SVG Rendering vs. Canvas Rendering *Interaction+* is currently designed for web-based visualizations using SVG technology, i.e., object-model based visualization. *Inateraction+* is unable to handle the web-based visualizations drawing on a Canvas, which is pixel-based. The difference in rendering approaches of SVG and Canvas causes the difference in the extraction of visual objects and attributes. In the SVG-based visualization, objects are accessible in DOM. However, Canvas-based visualizations are bitmaps where the visual objects need to be visually recognized. Several previous work [10] [25] successfully apply machine learning techniques to identify 10 classical chart types and extract visual objects. However, how to identify objects from more general visualization bitmap is one subject that remains to be explored.

Visual Information Extraction *Interaction+* is able to extract objects from DOM elements and quantify their attributes. *Interaction+* can handle the scope extension of attributes. For example, given a DOM element $< circle\ fill = "red"/ >$, which is moved by the transformation performed on its parent $< g\ transform = "translate(10,20)"/ >$, *Interaction+* can extract the attribute *fill* as well as the global position of $< circle >$. However, *Interaction+* is unable to parse the attributes declared in CSS. For example, given a DOM element $< circle\ class = "obj1"/ >$, which is styled by the declaration of *obj1* in CSS, *Interaction+* can not retrieve the detailed attributes in the CSS now. Also, *Interaction+* can not handle too complex attributes. For example, the shape of object $< path >$, which is described by the attribute d. In current version, *Interaction+* only distinguishes an object is $< path >$ or not but can not analyze its geometry from d further. *Interaction+* can be extended to detect the boundary box of $< path >$ element to compute its width and height.

Extension to Other Interactions In this work, we propose six types of add-on interactions to support users to conduct basic visual tasks. In the next stage, we identify two possible extensions of the interactions. One is to consider other crucial interaction techniques and analysis tasks in visualization [7] [30]. For example, an add-on view linking tool that helps users link multiple originally separated views in existing visualizations. It would work in this way: with users defined multiple regions, objects are extracted in each region and an automatic linking is built among different regions. The other extension is to go further with current interactions. Besides changing none-geometric properties of the objects in this work, it would be interesting to rearrange the objects in a different layout. Meanwhile, current interactions can be improved further, such as the selection function empowered with query relaxation engine [13], etc.

8 Conclusion

In this work, we propose a novel interaction enhancement method *Interaction+* to augment interactions in the existing visualizations on the web, where most of the existing interaction construction toolkits fail because of the inaccessibility to the visualization context (the back-end data, visualizing process, etc). Different from the conventional interaction constructing approaches, *Interaction+* takes the existing visualizations as input and performs the extraction of the visual information, by which a suite of flexible interactions are driven. The suite of interactions are seamlessly integrated in the existing visualizations, which allows users to perform direct manipulation in the original visualizations, including selection, comparison, filtering, etc. *Interaction+* can be applied to a variety of existing visualizations on the web from the well-developed visualizations to hand-made inforgraphics. We demonstrate its usage in two detailed scenarios and evaluate its effectiveness in a laboratory user study. The result shows that *Interaction+* enhances the interactivity of visualizations in an efficient and light-weight manner.

Acknowledgements

The authors wish to thank the anonymous reviewers for their valuable comments. The authors credit the help from Zhibang Jiang on visual design. This work is supported by NSFC No. 61672055 and the National Program on Key Basic Research Project (973 Program) No. 2015CB352503. This work is also funded by PKU-Qihoo Joint Data Visual Analytics Research Center.

References

[1] Javascript infovis toolkit. http://philogb.github.io/jit/.

[2] The new york times. http://www.nytimes.com.

[3] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: An implementation and evaluation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619–626, 1992.

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

[5] M. Bostock and J. Heer. Protovis: A graphical toolkit for visualization. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1121–1128, 2009.

[6] M. Bostock, V. Ogievetsky, and J. Heer. D3: data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309, 2011.

[7] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2376–2385, 2013.

[8] J. Brosz, M. A. Nacenta, R. Pusch, S. Carpendale, and C. Hurter. Transmogrification: Causal manipulation of visualizations. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, pages 97–106, 2013.

[9] J. Choi, D. G. Park, Y. L. Wong, E. Fisher, and N. Elmqvist. Visdock: A toolkit for cross-cutting interactions in visualization. *IEEE Transactions on Visualization and Computer Graphics*, 21(9):1087–1100, 2015.

[10] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of The American Statistical Association*, 79:531–554, 1984.

[11] M. Dixon and J. Fogarty. Prefab: implementing advanced behaviors using pixel-based reverse engineering of interface structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1525–1534, 2010.

[12] J. Harper and M. Agrawala. Deconstructing and restyling d3 visualizations. In *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, pages 253–262, 2014.

[13] J. Heer, M. Agrawala, and W. Willett. Generalized selection via interactive query relaxation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 959–968, 2008.

[14] J. Heer and B. Shneiderman. Interactive dynamics for visual analysis. *Communications of the ACM*, 55(4):45–54, 2012.

[15] N. Kong and M. Agrawala. Graphical overlays: Using layered elements to aid chart reading. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2631–2638, 2012.

[16] J. B. Kruskal and M. Wish. *Multidimensional scaling*, volume 11. Sage, 1978.

[17] Z. Liu and J. Stasko. Mental models, visual reasoning and interaction in information visualization: A top-down perspective. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):999–1008, 2010.

[18] D. R. Olsen, Jr., S. E. Hudson, T. Verratti, J. M. Heiner, and M. Phelps. Implementing interface attachments based on surface representations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 191–198, 1999.

[19] W. A. Pike, J. Stasko, R. Chang, and T. A. O'connell. The science of interaction. *Information Visualization*, 8(4):263–274, 2009.

[20] D. Ren, T. Hollerer, and X. Yuan. ivisdesigner: Expressive interactive design of information visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2092–2101, 2014.

[21] R. E. Roth. An empirically-derived taxonomy of interaction primitives for interactive cartography and geovisualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2356–2365, 2013.

[22] A. Satyanarayan and J. Heer. Lyra: An interactive visualization design environment. *Computer Graphics Forum*, 33(3):351–360, 2014.

[23] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer. Vega-lite: A grammar of interactive graphics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):341–350, 2017.

[24] A. Satyanarayan, R. Russell, J. Hoffswell, and J. Heer. Reactive vega: A streaming dataflow architecture for declarative interactive visualization. *IEEE transactions on visualization and computer graphics*, 22(1):659–668, 2016.

[25] M. Savva, N. Kong, A. Chhajta, L. Fei-Fei, M. Agrawala, and J. Heer. Revision: Automated classification, analysis and redesign of chart images. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 393–402, 2011.

[26] D. Simkin and R. Hastie. An information-processing analysis of graph perception. *Journal of the American Statistical Association*, 82(398):454–465, 1987.

[27] M. Steinberger, M. Waldner, M. Streit, A. Lex, and D. Schmalstieg. Context-preserving visual links. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2249–2258, 2011.

[28] M. O. Ward, G. Grinstein, and D. Keim. *Interactive data visualization: foundations, techniques, and applications*. CRC Press, 2010.

[29] C. Weaver. Cross-filtered views for multidimensional visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 16(2):192–204, 2010.

[30] J. S. Yi, Y. ah Kang, J. Stasko, and J. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1224–1231, 2007.