

# Graphical Overlays: Using Layered Elements to Aid Chart Reading

Nicholas Kong and Maneesh Agrawala

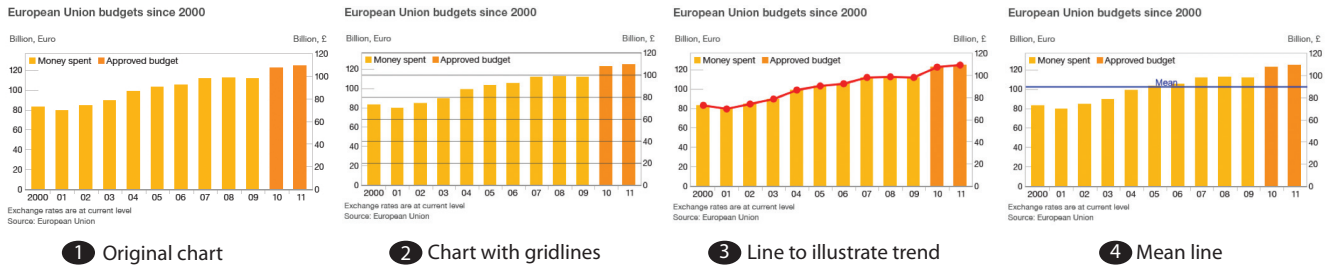


Fig. 1. In this chart of the European Union's budget by the BBC [7] the original design (1) forces viewers to mentally project a line to the y-axis to extract values. (2) A gridline overlay provides visual anchors, which can simplify the process of extracting values. (3) A line overlay encodes the data redundantly but better illustrates the trends in the data across time. (4) Finally, a statistical summary overlay depicts the mean value of the data so that viewers can easily compare each year's budget to the average budget across the years. All of these overlays were generated by our system without access to the underlying data, based on automatic extraction of the chart's mark and axis properties.

## Abstract—

Reading a visualization can involve a number of tasks such as extracting, comparing or aggregating numerical values. Yet, most of the charts that are published in newspapers, reports, books, and on the Web only support a subset of these tasks. In this paper we introduce *graphical overlays*—visual elements that are layered onto charts to facilitate a larger set of chart reading tasks. These overlays directly support the lower-level perceptual and cognitive processes that viewers must perform to read a chart. We identify five main types of overlays that support these processes; the overlays can provide (1) *reference structures* such as gridlines, (2) *highlights* such as outlines around important marks, (3) *redundant encodings* such as numerical data labels, (4) *summary statistics* such as the mean or max and (5) *annotations* such as descriptive text for context. We then present an automated system that applies user-chosen graphical overlays to existing chart bitmaps. Our approach is based on the insight that generating most of these graphical overlays only requires knowing the properties of the visual marks and axes that encode the data, but does not require access to the underlying data values. Thus, our system analyzes the chart bitmap to extract only the properties necessary to generate the desired overlay. We also discuss techniques for generating interactive overlays that provide additional controls to viewers. We demonstrate several examples of each overlay type for bar, pie and line charts.

**Index Terms**—Visualization, overlays, graphical perception, graph comprehension

## 1 INTRODUCTION

Reading a visualization involves a combination of perceptual and cognitive processes. Consider the task of extracting the value of the bar encoding the 2005 budget in Figure 1-1. The viewer must find the relevant bar, mentally project the top of the bar to a point on the y-axis, consider the nearest labeled tick marks and interpolate these numerical labels to compute the value at the projected point. While other types of charts (e.g., pie charts, line charts, scatterplots, etc.) use different visual encodings for the data, common chart reading tasks such as extracting, comparing and aggregating values, usually involve a similar sequence of perceptual and cognitive processes [26, 18, 20, 22]. However, many visualizations published in newspapers, reports, books and on the Web only support a subset of these processes.

In this paper, we introduce *graphical overlays*—visual elements that are layered onto a chart to facilitate a larger set of perceptual and cognitive processes involved in chart reading. We identify five main types of overlays, each designed to support different processes. (1) *Reference structure* overlays, such as gridlines (Figure 1-2), aid

the viewer in extracting and comparing values. (2) *Highlight* overlays draw the viewer's attention to certain marks by creating perceptual groups of marks. (3) *Redundant encoding* overlays allow viewers to extract numerical data in multiple ways and can be used to better depict some aspects of the data such as trends (Figure 1-3). (4) *Summary statistics* overlays depict aggregate information about the data set such as its mean, median or maximum (Figure 1-4). (5) *Annotation* overlays help viewers communicate and collaboratively analyze charts. While all five overlay types add visual elements to aid chart reading, they can also increase visual clutter. As a result, many published visualizations include few if any such overlays.

We present an automated system that applies user-chosen graphical overlays to existing chart bitmaps. By targeting existing charts, our system allow users to tailor published visualizations to better support the chart reading tasks they wish to complete. Our system takes a bitmap chart, an overlay type, and an optional set of user-specified parameters as inputs and outputs a graphical overlay for the chart. Our approach is based on the key insight that generating most graphical overlays requires only knowing the properties of the visual marks and axes that encode the data and does not require access to the underlying data values. Thus, our system analyzes the chart bitmap using existing tools (ReVision [25] and Datathief [33] in our implementation) to extract only the properties necessary to generate the desired overlay. We also show how our system can be used to generate interactive overlays and how it can place overlays underneath the marks in a chart.

• Nicholas Kong and Maneesh Agrawala are with the Computer Science Division of UC Berkeley, E-mail: {nkong, maneesh}@cs.berkeley.edu.

Manuscript received 31 March 2012; accepted 1 August 2012; posted online 14 October 2012; mailed on 5 October 2012.

For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

## 2 RELATED WORK

Our work is most closely related to two main areas of prior work: 1) layering information in visualizations, and 2) perceptual and cognitive models of graph comprehension.

### 2.1 Layered Visualizations

Layering information and visual elements is a common approach to designing visualizations. For example, William Playfair’s economic charts from the 18<sup>th</sup> century include gridlines layered onto line charts [23]. Gridlines are especially useful for extracting values from charts and commonly appear in visualizations published today. Other visual elements that are frequently layered onto visualizations include labels and highlights. Visualizations published on the Web sometimes provide interactive overlays such as the mouse-over data tooltips in Google Finance charts [12] and drop-down guidelines that follow the mouse cursor in the Wall Street Journal’s Foursquare visualization [29]. Web-based maps often layer additional information such as the crime incidents in Stamen’s Crimespotting [21] or building heat consumption in the UK government’s National Heat Map [34].

In the academic literature, researchers have developed specialized graphical overlays for specific tasks, including illustrating links between data tuples in a treemap [11]; illustrating links between multiple visual entities, such as text, images, or marks on a chart [27]; illustrating set relations [10]; providing references to offscreen items [4, 13]; and providing annotations [2, 15, 17]. In all of these examples, both published and academic, each overlay is designed for a specific application and is created based on knowledge of the data underlying the visualization. As a result these overlays cannot be applied to other existing charts. In contrast, our work focuses on providing a general-purpose tool for applying graphical overlays to existing chart bitmaps.

Although layers are commonly used in visualizations today, Tufte was amongst the first to investigate the use of “layering and separation” in visualization design [32]. He presents a number of visualizations that carefully use color, size and other retinal variables to visually distinguish different visual layers. For example, he explains how a musical staff composed of thin gray gridlines rather than thicker black lines allows viewers to visually separate the staff from the notes. Such layering allows viewers to easily direct their attention to the layer of their choice. Recently researchers have begun investigating how layered gridlines should be designed to facilitate such visual separation. They have conducted experiments to determine the minimum discriminable alpha value for layered gridlines [3, 28] and the optimal distance between gridlines [14]. We apply some of these results in the design of our overlays.

### 2.2 Models for Graph Comprehension

Researchers have developed a variety of models of the perceptual and cognitive processes viewers perform as they comprehend a graph [8, 19, 20, 22, 24]. All of these models share a similar high-level structure; the viewer first perceives the visual elements of the chart, associates the visual elements with graphical concepts (e.g., recognizing a bar or set of bars), and finally extracts the desired information from the chart. This process may iterate multiple times. For example, in a line chart with multiple intersecting lines, viewers may first process each line individually, then combine that information to complete their task. Our overlays are designed to support the early stages of this process; the perception of visual elements and the association of elements with graphical concepts. We consider previous work that has examined these two stages in more detail.

**Perceptual processes.** Many researchers have examined the perceptual processes that viewers perform when reading a visualization. Cleveland and McGill [9] introduced graphical perception – the study of how people estimate numerical values from visual encodings. They conducted a series of experiments to determine the accuracy of these estimates for position, length, and angle encodings. Simkin and Hastie [26] describe the elementary perceptual processes that viewers use when extracting and comparing values, including anchoring,

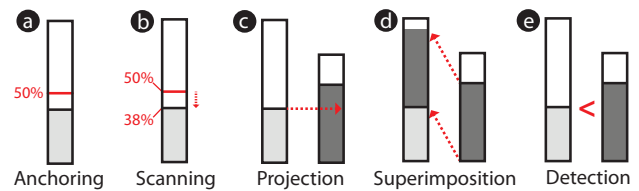


Fig. 2. Simkin and Hastie’s [26] elementary processes. (a) Anchoring occurs when a viewer segments a mark to extract a value. (b) Scanning occurs when a viewer estimates a mark’s value based on the anchor. (c) Projection occurs when a viewer extends a horizontal or vertical line. (d) Superimposition occurs when a viewer mentally moves a mark’s location. (e) Detection occurs when a viewer decides which of two marks is larger or smaller.

scanning, projection, superimposition, and detection (Figure 2). Our overlays are designed to facilitate these elementary processes.

Anchoring occurs when the viewer mentally segments a mark into regular divisions. The segmentation is based on the encoding type; a bar may be divided in half as in Figure 2-a while an entire circular pie might be divided into quarters. Scanning occurs when the viewer mentally interpolates from an anchor to the mark’s encoded value. In Figure 2-b, the viewer scans from the anchor to down to the value of the lower gray bar. Projection occurs when the viewer mentally draws a horizontal or vertical line to facilitate comparison of values. In Figure 2-c, the viewer projects the top of the left bar to the right bar to compare their heights. Superimposition occurs when the user mentally moves a mark near another mark. In Figure 2-d, the viewer superimposes the right dark gray bar onto the unfilled portion of the left bar to make an aggregate judgment. Detection is a quick process in which the viewer compares the relative magnitudes of two marks. In Figure 2-e, the viewer can quickly tell that the right, dark gray bar is larger than the left, light gray bar.

**Graph schemas.** A graph schema is a learned cognitive structure that describes the components of charts of different types [18, 20, 22]. For example, a schema for a bar chart might contain axes and bars. Viewers instantiate a graph schema based on the visual elements they perceive and use the schema to query the chart. If the viewer’s query cannot be answered by retrieving a component in the schema, the viewer may need to use additional cognitive processes (e.g., aggregation) to answer the query. Overlays provide additional visual elements that can extend viewers’ graph schemas. For example the overlay might provide numerical data labels for each mark or depict the mean value of the data. By extending the graph schema and providing direct access to additional information such overlays reduce cognitive load.

## 3 TAXONOMY OF OVERLAYS

Graphical overlays are designed to support the perceptual and cognitive processes used in chart reading tasks, such as extracting, comparing, or aggregating numerical values. We have analyzed a variety of charts published in books [31, 32] and online to identify five common types of overlays that each support a subset of these processes: reference structures, highlights, redundant encodings, summary statistic, and annotations. Figure 3 contains two examples of each overlay type for bar, pie, and line charts. We first consider how these overlays support cognitive and perceptual processes when they are added to a visualization in static form (Section 3.1), and then discuss how adding interaction to these overlays can further facilitate chart reading tasks (Section 3.2).

Although we focus on overlays that are layered on top of the marks in the base visualization, most overlay layers could alternatively appear below the marks as *underlays*. While overlays emphasize the content of the overlay, underlays emphasize the marks of the base visualization. Underlays support layering and separation better than overlays as viewers can more easily choose to ignore the underlay layer or direct attention their attention to it as necessary [28, 32] (Figures 8-5 and 8-6).

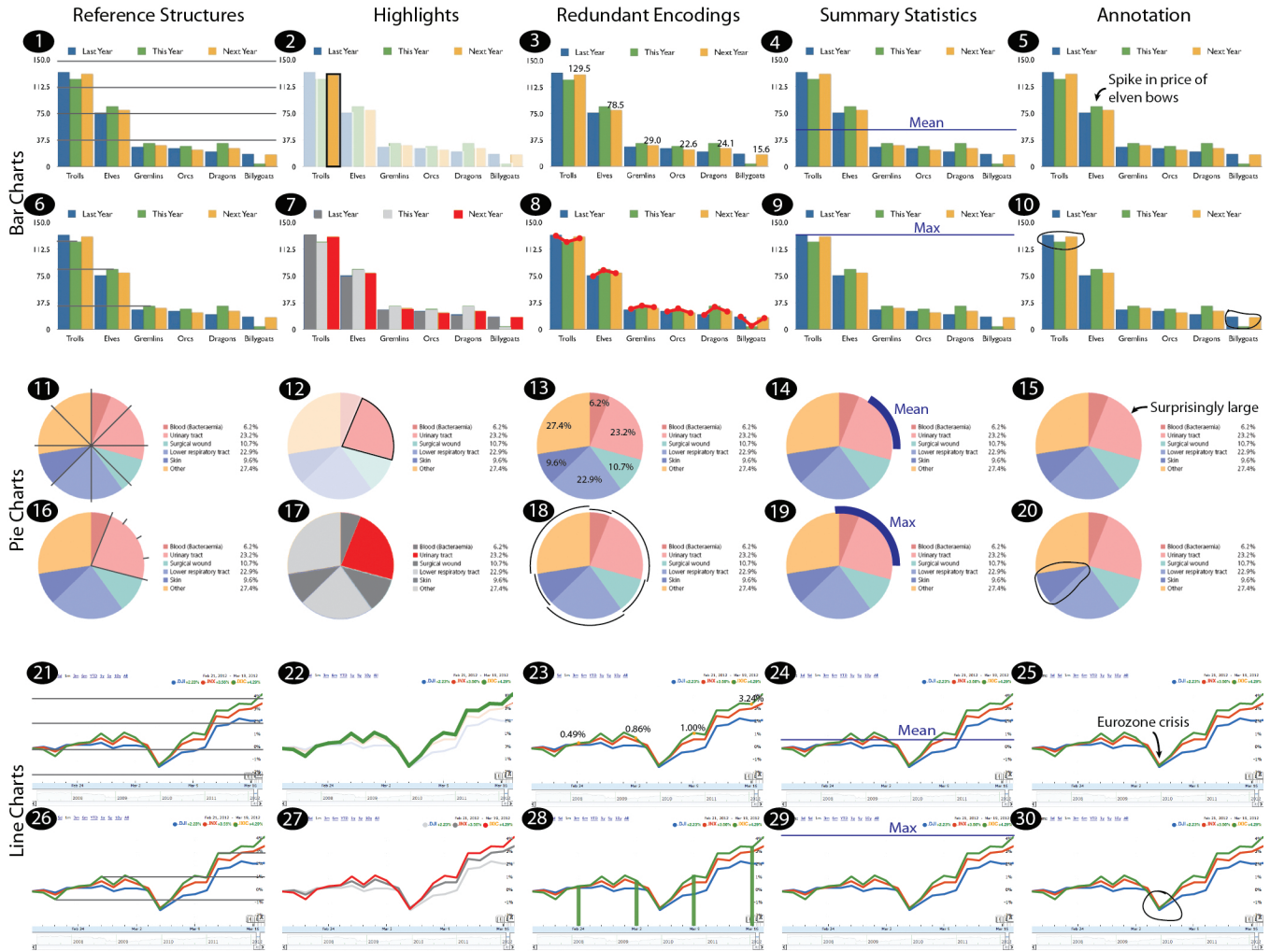


Fig. 3. Examples of visual overlays organized by type. These overlays were manually generated to illustrate the different overlay types. See Figure 9 for results generated by our system.

### 3.1 Overlay Types

**Reference structures.** Overlays can provide reference structures [3] that are designed to help viewers recover the mapping between the visual encoding and the data, as shown in the first column of Figure 3. Reference structures such as gridlines directly facilitate the elementary perceptual processes of anchoring and projection [26], as shown in Figure 2, and thereby help viewers extract and compare numerical data values. Gridlines can be placed at regular intervals along an axis as in Figures 3-1 and 3-21, or emanate from specific marks to provide a more direct reference for the mark value as in Figures 3-6 and 3-26. Figures 3-11 and 3-16 show polar gridlines for pie charts. The former example is a direct analog of the regular gridlines in the context of pie charts, while the latter example regularly divides a specific pie slice at 5 degree increments to facilitate reading the percentage value of the slice relative to the whole pie.

**Highlights.** Overlays can highlight important marks, as shown in the second column of Figure 3. Such highlights draw the viewer’s attention to specific marks by forcing them to share one or more distinguishing visual attributes (e.g., hue, saturation, texture, border color, drop shadows, etc.). The highlighted marks form a perceptual group based on the Gestalt principle of similarity [36]. Highlights can also act as deictic references for specific marks and thereby aid communication; viewers can refer to all the “red” highlighted marks rather than describing each one independently [16]. The overlays in Figures 3-2, 3-12, and 3-22 highlight certain marks by desaturating the

other marks so that they appear closer to the color of the background. Figures 3-2 and 3-12 add a high contrast black border to a highlighted mark while Figure 3-22 thickens the highlighted line to further emphasize the chosen mark. The overlays in Figures 3-7, 3-17, and 3-27 re-color specific marks to a pre-specified highlight hue while setting the color of the other marks to different shades of gray.

**Redundant encodings.** Overlays can provide alternative encodings of the data, as shown in the third column of Figure 3. Such redundant encodings can help viewers extract values or they can emphasize trends in the data. Figures 3-3, 3-13, and 3-23 show overlays containing numerical data labels. Viewers can directly read the data value from the label instead of activating additional cognitive processes [22] to infer the encoded mark value using axis labels. Figure 3-8 shows an overlay that joins the tops of a group of bars together using lines. Figure 3-28 shows the converse for a line chart, where bars are drawn underneath data points. Lines communicate trends better than bars, whereas bars communicate individual data values more effectively than lines [22, 38]. Although redundant encodings show the same data, each encoding may be best suited to a different chart reading task. Finally, Figure 3-18 shows an overlay that contains arcs outside pie slices that emphasize the length of the outer edge of the slice. Viewers may perceive such length encodings more accurately than the angle or area encoding provided by the original pie slices [9].

**Summary statistics.** Overlays can contain visual elements that illustrate summary statistics of the numerical data, such as the mean,

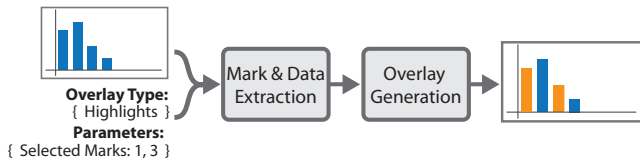


Fig. 4. Overview of the system. Our system takes a chart bitmap, overlay type, and author-specified design parameters (e.g., grid spacing, highlight hue, selected marks to highlight, etc.) as inputs. It then extracts the necessary marks and data and passes this information to the overlay generation component, which outputs a graphical overlay on the input bitmap.

median, standard deviation, or global or local maxima or minima, as shown in the fourth column of Figure 3. Such overlays allow viewers to visually compare individual marks to a statistic based on the complete distribution of data values. These overlays save the user from performing time-consuming cognitive functions to mentally aggregate the data and compute the summary statistics [22, 30]. Overlays presenting the mean and max values of the data set are shown in Figures 3-4, 3-14, and 3-24 and Figures 3-9, 3-19, and 3-29 respectively.

**Annotations.** Overlays can contain annotations that provide contextual information or comments, as shown in the fifth column of Figure 3. These overlays can aid users in communicating about charts or collaboratively analyzing charts as they allowing viewers to create arrows and other deictic references to marks [16, 17]. Figure 3-5, 3-15, and 3-30 show overlays that contain text annotations associated with marks. Figure 3-10, 3-20, and 3-30 show overlays that contain freehand annotations.

### 3.2 Adding Interaction to Overlays

Interaction in graphical overlays can help reduce visual clutter and increase the variety of tasks an overlay supports by allowing users to interactively specify which marks an overlay should target. For example, consider the overlay in Figure 3-6, where horizontal lines are drawn from bars to the vertical axis to support projection [26] (Figure 2-c). If lines were drawn from every bar, the display would quickly become cluttered and the viewer would need to perform additional cognitive processes to separate the lines from each other [20]. An interactive version of this overlay could allow the viewer to specify which of these lines to draw by clicking on bars, thus reducing visual clutter while still supporting the viewer’s specific task.

Interaction can also provide direct manipulation of overlay elements, which would support the superimposition and anchoring processes [26] (Figures 2-a and 2-d) and decrease the amount of spatial cognition the viewer must apply [30]. For example, the polar gridlines in Figures 3-11 and 3-16 could move with the viewer’s cursor and thereby allow the viewer to align the gridlines with a specific slice.

## 4 A SYSTEM FOR PRODUCING VISUAL OVERLAYS

We have developed an automated system that applies a user-chosen graphical overlay to existing charts. We refer to the person creating the overlay as the overlay author. Our implementation supports all of the overlays shown in Figure 3. Figure 4 shows an overview of our system. The input to our system is a chart bitmap and the specific overlay type. While our system provides default values for each of the overlay design parameters (e.g. gridline spacing, highlight hue, line thickness, and font style, etc.), authors can optionally set these parameters through a web-based user interface (Figure 5). Our system then extracts the necessary marks and data from the chart and outputs an overlay over the existing chart. We first discuss the properties required to construct each graphical overlay and the tools we use to obtain those properties. We then describe our implementation of the graphical overlays.

### 4.1 Chart Properties for Overlay Generation

One approach to building graphical overlays is to generate them directly from the underlying data values. However, correctly recovering

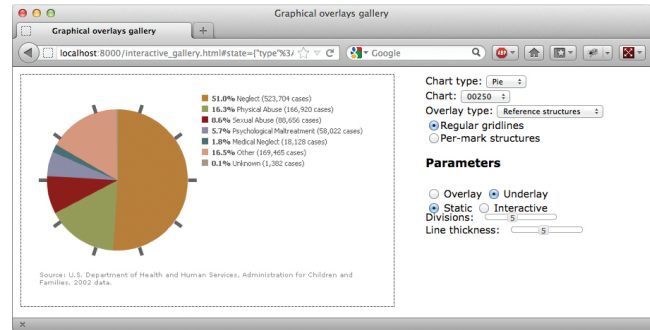


Fig. 5. User interface for authors to interactively modify graphical overlay design parameters such as line thickness, font size, the number of divisions in a gridline overlay, highlight hue, etc.

the data values from a chart bitmap is difficult; fully automated tools for recovering data from chart bitmaps such as ReVision [25], achieve only 47.9% accuracy in extracting data. Recovering the properties of marks and axes is much easier; ReVision achieves 71.4% accuracy in extracting these elements.

Our system takes advantage of the insight that most graphical overlays only require access to the mark and axis properties of the base visualization, and can be generated without access to the underlying data. For each type of overlay we consider whether it requires mark properties, axis properties or access to the underlying data.

**Mark properties.** Mark properties encompass the retinal variables used to visually encode the data [5]. Our system primarily uses two mark properties; (1) the location of the mark boundary and (2) the color of the mark. Knowledge of mark boundaries allows our system to generate pie chart gridlines (Figures 3-11 and 3-16). Similarly, the summary statistics overlays only require access to mark boundaries. For example, the position of a mean line in a vertical bar chart (Figure 3-4) can be computed by averaging the highest boundary position of each bar. Highlighting overlays (Figure 3 second column) require both the boundary and color of marks. In some cases mark properties can directly represent the data that encodes the mark; Figure 3-13 shows an overlay containing percentage value labels for each slice. The mark boundary provides direct access to the length of the outer slice boundary and the percentage of the pie covered by the slice.

**Axis properties.** Axis properties encompass the locations and orientations (i.e., the endpoints) of the axes in bar and line charts. The axis of a pie chart is implicitly based on a polar grid centered on the pie and is therefore extracted as part of the mark boundary property. We assume bar and line charts are two-dimensional and have no more than two axes, one vertical and one horizontal, explicitly depicted in the chart. This assumption implies that the chart area is given by the extents of the two axes. Regular gridlines (Figures 3-1 and 3-21) are an example of overlays that only require axis properties — we draw horizontal or vertical lines that span the chart area at regular intervals. Other overlays require both axis and mark properties (e.g., Figures 3-4, 3-6, 3-9, 3-24, 3-26, and 3-29).

**Data.** The data encompasses the numerical data values that the chart marks encode. Although any of the graphical overlays can be generated if the underlying data is known, the only overlays in Figure 3 that require access to the data are 3-3 and 3-23, which contain numerical data labels for a subset of the marks.

The first part of our system involves extracting the marks or data required for the author-chosen overlay (Figure 4). We use a mix of automatic and semi-automatic extraction techniques. For bar and pie charts, we use ReVision [25] to extract the relevant chart properties. ReVision applies computer vision and machine learning techniques to identify the chart type, extract the graphical marks, and infer the underlying data. For line charts we use Datathief [33] to semi-automatically extract chart properties. Our overlay system is designed to operate independently from the technique used to extract chart properties and

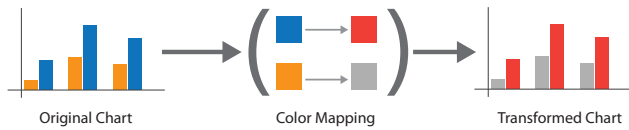


Fig. 6. One way our system highlights marks is by performing a color transformation. We apply a color mapping function to the chart; highlighted marks are colored red, while other marks are colored shades of gray.

could be combined with other extraction techniques [1, 6, 37]. Alternatively the author could provide the necessary properties directly to the system as input.

## 4.2 Overlay Implementation

The main component of our system is overlay generation (Figure 4), which we implemented in HTML and the JavaScript canvas. We render graphical overlays using a canvas element placed over the chart bitmap. We describe implementation details for each type of overlay, including the author-specified design parameters necessary to create each overlay. For brevity we describe our implementation of bar chart overlays assuming vertical bar charts, but our implementation includes the corresponding algorithms for horizontal bar charts.

### 4.2.1 Reference structures

Regular gridlines (Figures 3-1 and 3-21) require the author to provide a gridline direction (horizontal or vertical) and the number of divisions. Our system computes the chart area bounding box from the extracted axis properties and draws equally spaced lines in the author-specified direction to divide the chart into the author-specified number of divisions. For example, Figure 3-1 is generated using horizontal gridlines and four divisions. For a pie chart (Figure 3-11) the gridlines are arranged radially and the author only has to specify the number of divisions. Some overlays draw gridlines that emanate from or target individual marks (Figures 3-6, 3-16, and 3-26). In these cases the author must specify the target marks by clicking on them. Our system uses hit testing based on the mark boundaries to identify the clicked marks.

### 4.2.2 Highlights

All overlays containing highlights require authors to first select the marks or data points they wish to highlight. Our system generates two variations of overlays containing highlights. Figures 3-2, 3-12, and 3-22 highlight marks by desaturating non-highlighted marks. We use the mark boundaries to produce this effect. First, we trace the boundaries of all the non-highlighted marks and fill the boundaries with a semi-transparent white overlay. By only desaturating the areas of the chart within the boundaries of non-highlighted marks, we preserve the color of any background elements. We then draw a black line on the boundary of the highlighted marks for bar and pie charts. For line charts, we increase the width of the highlighted line.

The overlays shown in Figures 3-7, 3-17, and 3-27 highlight marks by performing a color transformation (Figure 6). Given a set of author-selected marks to highlight, our system finds all pixels in the chart bitmap that match the original colors of the target marks. It then modifies the color of these pixels to the author-specified highlight color (such as red, in these examples). Finally, it converts all the remaining pixels to their grayscale values by averaging their RGB components. This approach ensures that the highlighted marks are perceptually distinct from the non-highlighted marks. Moreover, because it transforms pixel colors across the entire chart it ensures that color-based legends correctly match the graphical marks even after the transformation. However this approach may also unintentionally re-color important chart elements such as text or background graphics.

### 4.2.3 Redundant encodings

Overlay authors can choose to add numerical data labels or an alternative visual encoding for all of the marks in the chart (the default)

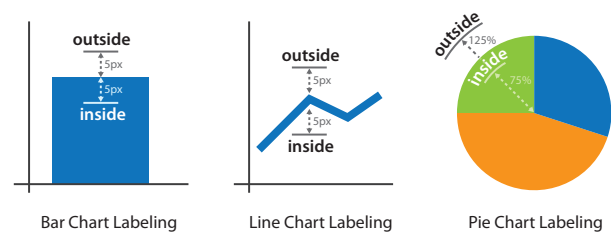


Fig. 7. Our system allows authors to specify whether labels are placed inside or outside marks. For line charts, this choice translates to above or below a mark.

or to a author-selected subset of marks. For data label overlays (Figures 3-3, 3-13, and 3-23) authors can optionally specify where each label should be placed relative to a mark boundary (Figure 7). For example, in a bar chart the author could specify that the label should be placed inside the bar, which translates to centering the label 5 pixels inside the top of the bar. Our system also provides default label layouts; for bar and line charts, it centers labels 5 pixels above each mark, while for pie charts it centers data labels radially within each pie slice and puts them at 75% of the radius from the center of the pie.

To generate alternative visual encodings (Figures 3-8, 3-18, and 3-28) our system relies on information about the mark boundaries. For the overlay shown in Figure 3-8, the author must select each group of marks to connect with a line. Our system then generates a circular point at the center of the top of each mark and draws lines connecting the circular points. For the overlay shown in Figure 3-18, our system uses the radius of the pie to draw circular arcs outside of the pie chart. For each slice our system draws a circular arc that spans the outer boundary of the slice and whose radius alternates between 3 and 5 pixels greater than the radius of the pie. Finally, for the overlay shown in Figure 3-28, our system draws rectangles whose fill color and height are the same as the author specified points on the line.

### 4.2.4 Summary statistics

Our system can create overlays depicting the mean, median, global maximum, global minimum of the data using just the mark and axis properties. For bar and line charts (Figures 3-4,3-9, 3-24, and 3-29), our system computes these statistics in image space using the heights in pixels of each mark, as given by the mark boundaries. The statistic is computed in pixel units and our system simply draws a line on the chart at the computed height and places the statistic label just above the line, centered horizontally in the chart area. For pie charts (Figures 3-14 and 3-19), our system computes the statistics in angular extent using the mark boundaries of each slice. It then draws a thick circular arc outside the pie which subtends the angular extent of the computed statistic and places the statistic label just outside the center of the arc.

### 4.2.5 Annotations

Our system implements two types of annotation layers: text annotations that are automatically associated with marks (Figures 3-5, 3-15, and 3-25) and freehand annotations (Figures 3-10, 3-20, and 3-30). For text annotations, our system allows authors to click on the chart and enter text. It then computes the nearest mark to the bounding box of the text and draws an arrow from the bounding box to the mark. For freehand annotations, our system gives authors a black, pen-like tool to create arbitrary annotations.

### 4.2.6 Interaction and Layering

Interactive overlays allow end-users to directly manipulate and move the information displayed in the overlay. Our system supports such viewer interaction for most of the overlays shown in Figure 3. For example, the interactive version of the polar gridline overlay Figure 3-11 rotates based on the viewer's mouse cursor position relative to the center of the pie (Figure 8-1). Our system also allows viewers to interactively select target marks for the highlighting overlays (Figures 3, second column) and the redundant encoding overlays (Figures 3, third

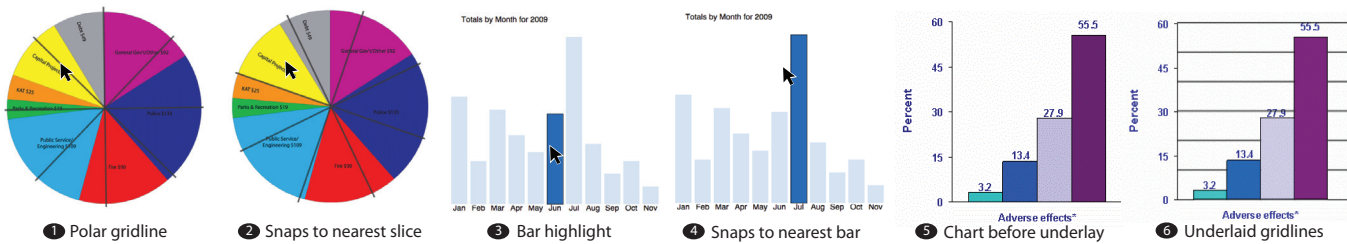


Fig. 8. Interactive overlays (1-4) and underlays (5-6) generated by our system. (1) The user can reposition a polar gridline by moving the cursor. (2) When the user holds the shift key, the gridline snaps to the slice nearest to the cursor. (3) The user can highlight a mark by mousing over it. (4) When the user holds the shift key, the overlay highlights the nearest mark to the cursor. (5) A bar chart without any overlays. (6) Our system creates an underlay effect by re-rendering the areas of the chart that lie within mark boundaries after rendering the gridlines.

column). Similarly viewers can select one or more marks as in (Figures 3-6, 3-16, and 3-26) and our system will generate gridlines emanating from just the selected marks. Note that our system also implements snapping to marks in interactive overlays. For example, holding the shift key in an interactive overlay containing highlights will highlight the mark nearest to the mouse cursor (Figure 8-4).

Finally our system can layer the information designed for an overlay, under the marks to produce an underlay. Our system first renders the graphical overlay as we have described above and then draws the sections of the image bitmap that lie within each mark boundary on top of the overlay layer (Figure 8-6). While this approach works well for elements such as gridlines that extend beyond the boundaries of the marks, it may not produce the desired effect when the elements lie within a mark as for some highlight overlays. However, we believe that alternative underlay templates could be designed to handle such cases.

## 5 RESULTS

We have used our system to generate each type of overlay shown in Figure 3 for 32 bar, 53 pie, and 7 line charts drawn from the Web. Figure 9 presents a subset of these results for 15 different charts (original chart is shown on top and overlaid chart is shown on the bottom). We include a larger set of both static and interactive overlays generated by our system on the web at <http://vis.berkeley.edu/papers/grover/supplemental>.

**Reference structures.** Figures 9-1, 9-6, and 9-11 show example reference structures. The original chart in Figure 9-1 did not contain gridlines, so we overlay a regular gridline with five divisions to aid projection and extraction of the data values. In Figure 9-6, we overlay ticks at regular 22.5 degree intervals for a selected pie slice to facilitate anchoring and extraction of the slice angle. In Figure 9-11, we overlay horizontal gridlines from a few selected data points of interest to also aid projection.

**Highlights.** Figures 9-2, 9-7, and 9-12 show examples of highlights. In Figures 9-2 and 9-12 we use the color transformation approach to facilitate visual grouping and draw the viewer’s attention to the red highlighted elements. While this approach converts most pixels outside the selected marks to shades of gray, it properly preserves the mapping between the legend and the color of both the highlighted and non-highlighted marks. In Figure 9-7 we apply a highlight overlay that desaturates non-highlighted marks. This approach preserves the hue and texture of the pie slices.

**Redundant encodings.** Figures 9-3, 9-8, and 9-13 show examples of redundant encodings. In Figure 9-3, the overlaid red line connecting the blue bars illustrates the trend in the data over years better than the bars alone. In Figure 9-8, the overlaid arcs facilitate extracting the value encoded by the pie slice based on arc length rather than slice angle or area. The arcs are especially useful for the very small slices. In Figure 9-13, overlaid bars emphasize the values of selected points along the x-axis, which is especially useful here because the x-axis encodes a categorical variable.

**Summary statistics.** Figures 9-4, 9-9, and 9-14 show overlays that contain visual representations of the max, mean, and median respec-

tively. In each of these cases, the statistic was computed using all of the marks. However, the overlay author can choose to compute these statistics using a subset of the marks, for example a single line in a chart.

**Annotations.** Figures 9-5 and 9-10 show freehand annotations. In Figure 9-5, the annotation author has marked the large jump between the three lowest bars in the chart and the other bars. In Figure 9-10, the author has drawn attention to the text in one of the slices. Figure 9-15 shows a text annotation that for the sharp peak in the line chart.

Our system also implements interactive versions of many of these overlays. Figure 8-1 depicts an interactive gridline overlay for a pie chart that is positioned based on the viewer’s cursor position. Figure 8-3 depicts an overlay that highlights the mark underneath the cursor by desaturating the other marks. Our interactive overlays also implement snapping. When the viewer holds the shift key, the gridline snaps to the nearest pie slice (Figure 8-2), or the system highlights the nearest bar to the cursor (Figure 8-4).

### 5.1 Limitations

While our overlay generation system can produce high-quality overlays for many existing charts, it also has a few limitations. Our system does not consider visual elements that fall outside of the marks and axes. Such elements include legends, axis labels, background graphics, background colors, etc. In some cases our overlays may interfere with these elements. In Figure 10-1, for example, the original chart bitmap contains two textboxes describing the underlying data. Our highlight overlay desaturates the parts of both textboxes that lie within the non-selected pie slices, making it difficult to read parts of the text.

Although our system does not require the underlying data to generate most overlays, it does require accurate knowledge of mark and axis properties. Inaccuracies in the location of mark boundaries can produce visual artifacts in some overlays. Figure 10-2 shows a highlight overlay that contains artifacts at the bar boundaries due to slight inaccuracies in boundary extraction from the ReVision [25] system. While overlay authors could manually fix such inaccuracies today, we also expect that automated mark extraction techniques will improve and produce pixel-accurate boundaries.

Some of our overlay designs have specific limitations. The color transformation overlay (Figures 9-7 and 9-12) assumes a constant fill color within a mark. Antialiasing or compression artifacts violate this assumption, and can cause artifacts at the edges of marks (Figure 10-3). Finally, we use a simple layout algorithm for placing data labels, which can result in label-label overlaps.

## 6 FUTURE WORK AND CONCLUSION

When reading a chart to extract, compare, or aggregate values, viewers must complete a sequence of perceptual and cognitive processes. However, many existing visualizations do not support these processes. We have identified five types of graphical overlays that are designed to support the perceptual and cognitive processes and facilitate chart reading. We have shown that most of these overlays can be generated based only on mark and axis properties and do not require access to the underlying data. Based on this insight we have presented a system that

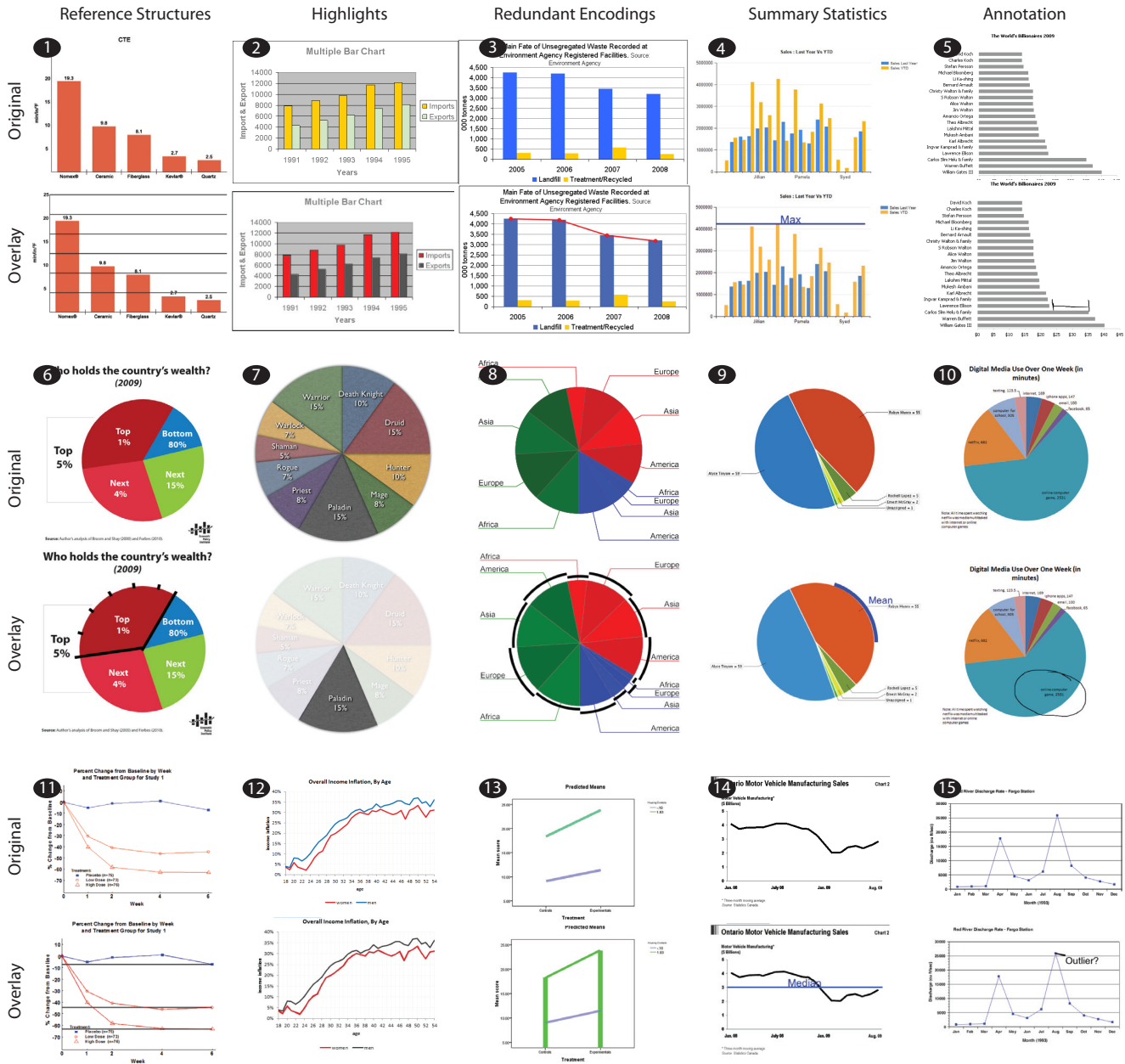


Fig. 9. Example static overlays generated by our system for 15 different charts drawn from the Web. The original chart is shown on top and the overlay is shown on the bottom.

can make it easy to add graphical overlays of all five types to existing charts.

We plan to address the limitations of our system in our future work. Our system only considers the marks and axes of the chart and may produce overlays that interfere with background graphics, descriptive text, legends, etc. Automated techniques for finding background graphics and text would allow our system produce overlays that do not obscure or overlap with these graphic and text elements.

Our label placement algorithms can generate overlaps that make it difficult to read the label text or the underlying chart elements. Such overlap is especially problematic if the overlay author chooses to label lots of closely spaced marks. More sophisticated label layout algorithms, such as approaches based on simulated annealing [35], could mitigate these issues.

Our system currently provides default design parameters (e.g. highlight color, font type, etc.) for each overlay and chart type. We chose these defaults via ad-hoc experimentation; we tested a few different parameter values across a small corpus of charts and selected those

that we thought worked best. Automatic selection of parameter values based on additional analysis of the chart image could improve the default settings. For example, we could use the size of the chart to suggest the number of gridlines based on graphical perception results (e.g., [14]), or we could select a highlight hue that differs strongly from the rest of the chart.

Our overlay taxonomy is focused on bar, pie and line charts. We believe that other types of visualizations such as treemaps or parallel coordinate displays could benefit from the types of overlays we propose in our taxonomy. However the specific designs are likely to differ depending on the chart type.

Finally, our overlay designs are inspired by perceptual and cognitive models of graph comprehension. However, these models have not been fully validated through empirical studies. We plan to use our overlay generation system to run user studies investigating how overlay design parameters facilitate the perceptual and cognitive processes involved in chart reading tasks.

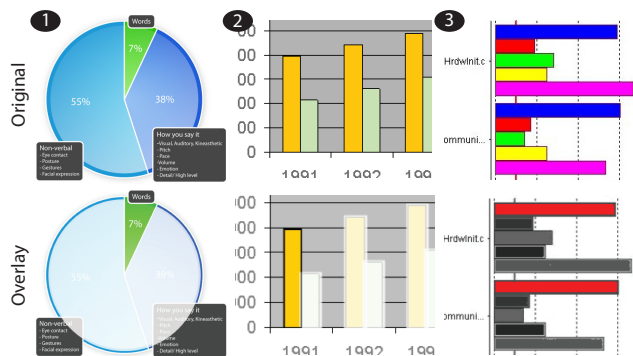


Fig. 10. Limitations of our overlay system. (1) Desaturation affects the text boxes. (2) Mark boundary errors cause desaturation to extend beyond the true mark boundary. (3) Color transformation does not capture all the pixels inside marks due to antialiasing or compression noise. Note that (2-3) are cropped versions of the original overlays.

## ACKNOWLEDGMENTS

The authors would like to thank Wesley Willett for figure assistance. This work was supported by NSF grant IIS-1016920 and an NSERC PGS D scholarship.

## REFERENCES

- [1] Arizona Software. GraphClick. <http://www.arizona-software.ch/graphclick>. Retrieved Jun 11, 2012.
- [2] D. Barger and T. Moscovich. Reflowing digital ink annotations. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 385–393, 2003.
- [3] L. Bartram, B. Cheung, and M. Stone. The effect of colour and transparency on the perception of overlaid grids. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1942–1948, Dec. 2011.
- [4] P. Baudisch and R. Rosenholtz. Halo: a technique for visualizing off-screen objects. In *CHI '03: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 481–488, 2003.
- [5] J. Bertin. *Sémiologie Graphique*. Gauthier-Villars, 1967.
- [6] I. Bormann. DigitizeIt. <http://www.digitizeit.de>. Retrieved Jun 11, 2012.
- [7] British Broadcasting Corporation. EU budget plans for 2011. <http://www.bbc.co.uk/news/uk-politics-11645975>. Retrieved Mar 31, 2012.
- [8] P. A. Carpenter and P. Shah. A model of the perceptual and conceptual processes in graph comprehension. *Journal of Experimental Psychology: Applied*, 4(2):75–100, 1998.
- [9] 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(387):531–554, 1984.
- [10] C. Collins, G. Penn, and S. Carpendale. Bubble sets: Revealing set relations with isocontours over existing visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1009–1016, Nov.-Dec. 2009.
- [11] J.-D. Fekete, D. Wang, N. Dang, A. Aris, and C. Plaisant. Interactive poster: Overlaying graph links on treemaps. *IEEE Symposium on Information Visualization*, 2003.
- [12] Google Finance. <http://www.google.com/finance/>. Retrieved Mar 30, 2012.
- [13] S. Gustafson, P. Baudisch, C. Gutwin, and P. Irani. Wedge: clutter-free visualization of off-screen locations. In *CHI '08: Proceedings of the 26th annual SIGCHI conference on Human factors in computing systems*, pages 787–796, 2008.
- [14] J. Heer and M. Bostock. Crowdsourcing graphical perception: Using mechanical turk to assess visualization design. In *CHI '10: Proceedings of the 28th international conference on Human factors in computing systems*, pages 203–212, 2010.
- [15] J. Heer, F. B. Viégas, and M. Wattenberg. Voyagers and voyeurs: supporting asynchronous collaborative information visualization. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 1029–1038, 2007.

- [16] W. C. Hill and J. D. Hollan. Deixis and the future of visualization excellence. In *Proc. of IEEE Visualization*, pages 314–320, 1991.
- [17] N. Kong and M. Agrawala. Perceptual interpretation of ink annotations on line charts. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 233–236, 2009.
- [18] S. M. Kosslyn. Understanding charts and graphs. *Applied Cognitive Psychology*, 3(3):185–225, 1989.
- [19] S. Kriz and M. Hegarty. Top-down and bottom-up influences on learning from animations. *International Journal of Human-Computer Studies*, 65(11):911–930, 2007.
- [20] G. L. Lohse. A cognitive model for understanding graphical perception. *Human-Computer Interaction*, 8(4):353–388, 1993.
- [21] M. Migurski, T. Carden, and E. Rodenback. Oakland Crimespotting. <http://oakland.crimespotting.org/>. Retrieved Mar 30, 2012.
- [22] S. Pinker. *A theory of graph comprehension*, pages 73–126. Lawrence Erlbaum Associates, 1990.
- [23] W. Playfair. *Playfair's Commercial and Political Atlas and Statistical Breviary*. Cambridge University Press, 2005.
- [24] R. M. Ratwani, J. G. Trafton, and D. A. Boehm-Davis. Thinking graphically: Connecting vision and cognition during graph comprehension. *Journal of Experimental Psychology: Applied*, 14(1):36–49, 2008.
- [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 *UIST '11: 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, dec. 2011.
- [28] M. Stone and L. Bartram. Alpha, contrast and the perception of visual metadata. *Color Imaging Conference*, 2009.
- [29] A. Sun, J. Valentino-DeVries, and Z. Seward. A Week on Foursquare. <http://graphicsweb.wsj.com/documents/FOURSQUAREWEEK1104>. Retrieved Mar 30, 2012.
- [30] S. Trickett and J. Trafton. Toward a comprehensive model of graph comprehension: Making the case for spatial cognition. In *Diagrammatic Representation and Inference*, volume 4045 of *Lecture Notes in Computer Science*, pages 286–300. Springer Berlin, 2006.
- [31] E. R. Tufte. *The Visual Display of Quantitative Information*. Graphics Press, 1983.
- [32] E. R. Tufte. *Envisioning Information*. Graphics Press, 1990.
- [33] B. Tummers. DataThief III. <http://www.datathief.org>. Retrieved Mar 31, 2012.
- [34] UK Department of Energy and Climate Change. National Heat Map. <http://ceo.decc.gov.uk/nationalheatmap/>. Retrieved Mar 30, 2012.
- [35] I. Vollick, D. Vogel, M. Agrawala, and A. Hertzmann. Specifying label layout style by example. In *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 221–230, 2007.
- [36] M. Wertheimer. Laws of organization in perceptual forms. *A source book of Gestalt psychology*, pages 71–88, 1938.
- [37] L. Yang, W. Huang, and C. Tan. Semi-automatic ground truth generation for chart image recognition. In H. Bunke and A. Spitz, editors, *Document Analysis Systems VII*, volume 3872 of *Lecture Notes in Computer Science*, pages 324–335, 2006.
- [38] J. Zacks and B. Tversky. Bars and lines: A study of graphic communication. *Memory & Cognition*, 27(6):1073–1079, 1999.