Visualizing Data with Bounded Uncertainty

Chris Olston*
Stanford University
olston@cs.stanford.edu

Jock D. Mackinlay
Palo Alto Research Center
mackinlay@parc.com

Abstract

Visualization is a powerful way to facilitate data analysis, but it is crucial that visualization systems explicitly convey the presence, nature, and degree of uncertainty to users. Otherwise, there is a danger that data will be falsely interpreted, potentially leading to inaccurate conclusions. A common method for denoting uncertainty is to use error bars or similar techniques designed to convey the degree of statistical uncertainty. While uncertainty can often be modeled statistically, a second form of uncertainty, bounded uncertainty, can also arise that has very different properties than statistical uncertainty. Error bars should not be used for bounded uncertainty because they do not convey the correct properties, so a different technique should be used instead.

In this paper we describe a technique for conveying bounded uncertainty in visualizations and show how it can be applied systematically to common displays of abstract charts and graphs. Interestingly, it is not always possible to show the exact degree of uncertainty, and in some cases it can only be displayed approximately. We specify an algorithm that approximates the degree of uncertainty to make it displayable while minimizing the overall loss in accuracy. In addition, we consider new data delivery paradigms that offer mechanisms for interactive control over uncertainty levels, but whose use may result in hidden side effects. We propose interfaces that offer control of uncertainty levels to the user in ways that encourage careful use of these facilities.

Keywords: uncertainty visualization, bounded uncertainty, adjustable uncertainty

1 Introduction

In most data-intensive applications, uncertainty is a fact of life. For example, in scientific applications, error-prone measurements or incomplete sampling often result in uncertain data. Another example is financial analysis, where it is common for some data to represent uncertain projections about future behavior. Even when it is possible to gather precise data, there are many real-time applications, such as network monitoring, mobile object tracking, and wireless ecosystem monitoring, in which uncertainty may be introduced intentionally to conserve system resources while data is being transmitted or processed. When data is uncertain, it is critically important that analysis tools, including information visualization tools, make users aware of the presence, nature, and degree of uncertainty in the data as these factors can greatly impact decision-making. If users are misinformed about the uncertainty associated with their data, they may draw inaccurate conclusions, potentially leading to costly mistakes.

A report by the US Department of Commerce National Institute of Standards and Technology (NIST) [TK94] identifies two predominant forms of uncertainty, which we call *statistical uncertainty* and *bounded uncertainty*. Statistical and bounded uncertainty have dramatically different meanings. Statistical uncertainty is typically captured by a potentially infinite distribution of possible values with a peak indicating the most likely estimate. In contrast, with bounded uncertainty no distribution of values can be assumed, but the exact value is known to lie inside an interval defined by precise lower and upper bounds.

Pang et al. [PWL97] argue, as we do, that uncertainty should be presented along with data in visualization applications. After discussing traditional techniques for showing statistical uncertainty such as error bars, they propose an extensive suite of techniques for conveying uncertainty in scientific visualization applications. Many of these techniques can be adapted to information visualization scenarios. However, techniques for conveying statistical uncertainty tend to be misleading when used for bounded uncertainty for two reasons. First, users have been trained to interpret them as probabilistic bounds on an unbounded distribution of possible values. Second, since error bars are typically used in conjunction with an estimated exact value, the existence of a single most likely value is strongly implied.

Visualizations should clearly differentiate between the two forms of uncertainty, making it obvious whether the uncertainty is statistical or bounded in addition to conveying the degree of uncertainty. Therefore, we advocate the use of two distinct techniques for the two forms of uncertainty. To convey statistical uncertainty, it is appropriate to display the most likely value along with error bars or other glyphs as in [PWL97]. To convey bounded uncertainty, we advocate a systematic technique based on widening the boundaries and positions of graphical elements and rendering the uncertain region in fuzzy ink. We show how to apply this technique, which we call *ambiguation*, to common displays of abstract charts and graphs. Interestingly, it is not always possible to show the exact degree of uncertainty, and in some cases

*Research supported by a National Science Foundation graduate research fellowship.

it can only be displayed approximately. We specify an algorithm that approximates the degree of uncertainty to make it displayable while minimizing the overall loss in accuracy.

We also consider new issues raised by recently proposed interactive data delivery paradigms in which either statistical or bounded uncertainty is intentionally introduced to improve performance [BP93, DKP+01, HAC+99, HSW94, OW00, OW02b, YV00a, YV00b]. In these environments, data and uncertainty levels can change dynamically, and it is easy for users to falsely interpret sudden changes in the uncertainty bounds as changes in the underlying data. We propose a method for masking sudden jumps in uncertainty when they do not correspond to significant data changes to avoid drawing the user's attention.

The central feature of these new data delivery paradigms is that applications can control the degree of uncertainty. While it can be beneficial to offer control over uncertainty levels to users, naive interfaces may not effectively expose the tradeoffs involved. Specifically, decreasing the uncertainty for some of the data typically results in either increased uncertainty for other data or increased system resource utilization. We propose ways to offer users control over uncertainty levels that encourage judicious use of the control mechanism.

The remainder of this paper is structured as follows. We begin by discussing related work in Section 2. We then formally define the two forms of uncertainty and describe some new interactive data delivery contexts in which they occur in Section 3. Then, in Section 4 we describe our systematic approach to conveying the presence, form, and degree of uncertainty. We address misleading sudden jumps in Section 5. We then discuss user-controlled uncertainty tuning for interactive data delivery in Section 6. Finally, we summarize the paper in Section 7.

2 Related Work

In certain visualization scenarios, data may be unavailable for display or even purposefully omitted for a variety of possible reasons, giving rise to uncertainty. The importance of visually informing the user of the absence of data has been identified [WO98] and techniques for doing so have been proposed in, *e.g.*, Clouds [AHL+98, HAC+99] and Restorer [TCS94]. We focus on a different type of uncertainty where all the data is present but precise values are not known.

Numerous ways to convey the degree of uncertainty in data using overlayed annotations and glyphs have been proposed, as in, *e.g.*, [PWL97]. Another approach is to make the positions of grid lines used for positional reference ambiguous [CR00]. Uncertainty can also be indicated by adjusting the color, hue, transparency, etc. of graphical features as in, *e.g.*, [DK97, Mac92, vdWvdGG98]. Some techniques for conveying uncertainty by widening the boundaries of graphical elements have also been proposed. For example, in [WSF+96], the degree of uncertainty in the angle of rotation of vectors is encoded in the width of the vector arrows. Also, [PWL97] proposes varying the thickness of three-dimensional surfaces to indicate the degree of uncertainty.

To our knowledge, however, none have focused on accurately and unambiguously conveying not only the presence and degree but also the form of uncertainty in data, as we do. We also believe that our work is the first to establish systematic methods for conveying bounded uncertainty by widening the boundaries and positions of graphical elements in abstract charts and graphs. The approach in [FWR99] for displaying cluster densities gives a visual appearance similar to our ambiguated line charts (discussed later) but serves a different purpose.

Our work also addresses control over uncertainty levels. Interfaces for controlling uncertainty levels were proposed in [HAC+99], but that work does not address ways to make the user cognizant of tradeoffs between decreased uncertainty and increased resource utilization.

3 Forms and Sources of Uncertainty

In this section we first characterize the two common forms of uncertainty and then provide a brief overview of some emerging data delivery paradigms in which uncertainty in one of these two forms is intentionally introduced for performance reasons.

3.1 Uncertainty Representations

In this paper we consider two commonplace forms of uncertainty, as described in [TK94], [PWL97], and elsewhere. Consider a numeric data object O whose exact value V is not known with certainty. There are two predominant ways in which partial knowledge about the possible values of V can be represented: *statistical uncertainty* and *bounded uncertainty*. Under statistical uncertainty, the uncertain value of a data object can be represented in a number of ways, depending on the statistical model. In one common case, when errors follow a normal distribution, the uncertain value of a data object can be represented by a three-tuple $\langle E, D, P \rangle$ of real numbers, where $D \geq 0$ and $P \in (0, 1]$. Here, E is an estimate that represents the most likely candidate for the unknown value V, and P is the probability that V lies in the confidence interval $[E - D, E + D]$. Typically, P is fixed at, say, $P = 0.95$, and D is chosen so that the value V lies inside the confidence interval $[E - D, E + D]$ with probability P. Under bounded uncertainty, there is some numeric interval $[L, H]$ that is guaranteed to contain the exact value V, *i.e.*, $L \leq V \leq H$. Under bounded uncertainty, the probability that V is outside the interval is zero, but, unlike with statistical uncertainty, no assumptions can be made about the probability distribution of possible values inside the interval.

Both forms of uncertainty commonly occur in scientific and other applications [TK94]. For example, bounded uncertainty can occur when measurements are taken using a device having an unknown degree of imprecision that lies within known bounds. Statistical uncertainty can occur, for example, when single or repeated measurements are taken in conditions exhibiting experimental variability, often resulting in an unbounded probability distribution over possible values featuring a central peak. Both bounded and statistical uncertainty can also occur in emerging data delivery paradigms that intentionally introduce uncertainty for performance reasons. In these paradigms there is often the opportunity to adjust the uncertainty levels interactively, unlike with traditional sources of uncertainty. Next, we discuss two interactive data delivery techniques that exhibit these properties: *progressive sampling* and *approximate cache synchronization*.

3.2 Sampling

Some data represents aggregate quantities such as an average or a sum over a large population of source data. Aggregation with grouping is a common operation performed in the analysis of large data sets [GCB+97]. For example, consider an academic database consisting of student grade reports. A query might request average grades, grouped by major department. Sampling techniques can be used to reduce the time required to compute the aggregated quantities, giving results that carry statistical uncertainty. If there has not been time to sample the data in advance, estimates can be generated on the fly by scanning the data in random order to generate a stream of estimates that grow more accurate over time. As time progresses, the estimate E changes and the confidence interval $[E - D, E + D]$ gradually shrinks in size (assuming P is fixed). Eventually, after all the data has been processed, the width of the confidence interval becomes zero indicating that the exact value V is known, *i.e.*, $D = 0$ and $E = V$.

3.2.1 Progressive Sampling

[HH97] describes an approach in which a large data set is partitioned into several groups, and *progressive sampling* is performed simultaneously on all the groups to generate one stream of estimates per group. Since the data for all the groups (*e.g.*, grade records across all departments) is often stored in the same database, computing all the aggregate values (*e.g.*, average grades) usually requires sharing the resources of the database. The simultaneous computation of several aggregate quantities from the same database presents an opportunity to tune the amount of resources dedicated to refining each group's estimate. Dedicating more resources to one group will cause the corresponding confidence interval to shrink more rapidly. However, since the total amount of resources is fixed, the confidence intervals of other groups will shrink more slowly as a consequence. By specifying how to allocate resources among groups, the client application can control the relative rate at which estimates for different groups improve [HH97].

3.3 Approximate Cache Synchronization

Often, information analysis and visualization tasks are performed at a distinct location from where data is generated or collected. For example, in scientific applications, remote sensor readings taken at different locations might be fed over a network to a central monitoring station for real-time visualization. Typically, in these applications and others, the monitoring station caches the remotely generated data and uses it for visualization. Ideally, the cached data could be kept consistent with the remote data as it changes, but exact consistency would require refreshing the cache every time the data changes at any of the sources. Doing so could be prohibitively expensive in terms of the network and computational resources required if the amount of remote data is large or frequently updated. Thankfully, in many applications exact consistency is unnecessary because some degree of uncertainty can be tolerated as long as the user is made aware of it [YV00b]. Recently, alternatives to exact cache consistency have been proposed in which cached data is only kept approximately consistent with respect to source data to reduce the overhead of refreshing, *e.g.*, [BP93, DKP+01, HSW94, OW00, OW02b, YV00a, YV00b].

One simple and flexible *approximate cache synchronization* technique for numeric data, initially proposed in [OW00], works as follows. For each remote data object O being cached at a central location, the cache stores a numeric interval $[L, H]$ that is guaranteed to contain the exact source value V, *i.e.*, $L \leq V \leq H$. The source and cache cooperate to ensure that this containment guarantee always holds, thereby providing bounded uncertainty. The positions of the interval endpoints are determined based on systems considerations, so the end application cannot assume any probability distribution for V within the interval. The width of the interval, *i.e.*, $H - L$, determines the degree of uncertainty, and also the overhead required to maintain the containment guarantee. Wide intervals carry high uncertainty but tend to incur less refreshing overhead than narrower intervals with lower uncertainty.

There are two different scenarios in which caching intervals is useful. First, in *constrained uncertainty* scenarios, resources are flexible but usage incurs a cost, so uncertainty should be introduced as much as is tolerable to the end application to minimize resource utilization. In a system described in [OW02a], applications can assign and adjust constraints on uncertainty levels by specifying the maximum interval width for individual objects, or for an aggregate over a set of objects. The system reduces resource utilization as much as possible while still meeting the constraints. By contrast, in *constrained resource* scenarios, the computational and network resources available for refreshing data are severely limited. In this scenario, it may not be possible to meet fixed uncertainty goals, but it is still desirable to minimize the overall level of uncertainty within the limitations on resource utilization. In a system described in [OW02b], applications can assign and adjust priorities for cached objects, and the system refreshes higher priority objects more frequently than lower priority objects. In this way, low uncertainty can be achieved for objects assigned high priorities by the application, in exchange for increased uncertainty for the other objects.

4 Representing Uncertain Data Visually

Having described the two common forms of uncertainty and some ways they can occur, we are now ready to discuss ways to represent uncertain data visually. In most abstract charts and graphs, data values are graphically encoded either in the positions of graphical elements, as in a scatterplot, or in the extent (size) of elements along one or more dimensions, as in a bar chart. When the underlying data is uncertain, we believe it is appropriate to clearly indicate not only the presence and degree but also the form of uncertainty. As described in Section 3.1, statistical and bounded uncertainty encode two dramatically different distributions of potential values. Due to this key difference, using the same display technique to represent both forms of uncertainty could mislead the user. Instead, we advocate two alternative methods for conveying uncertainty in the positions or extents of graphical representations of data: *error bars* for statistical uncertainty and *ambiguation* for bounded uncertainty. We begin by describing these general techniques and then show how they can be applied to some common types of charts and graphs.

4.1 Error Bars

Error bars and their variants have been well studied as a suitable means to convey statistical uncertainty [Cle85, Tuf01, Tuk77].

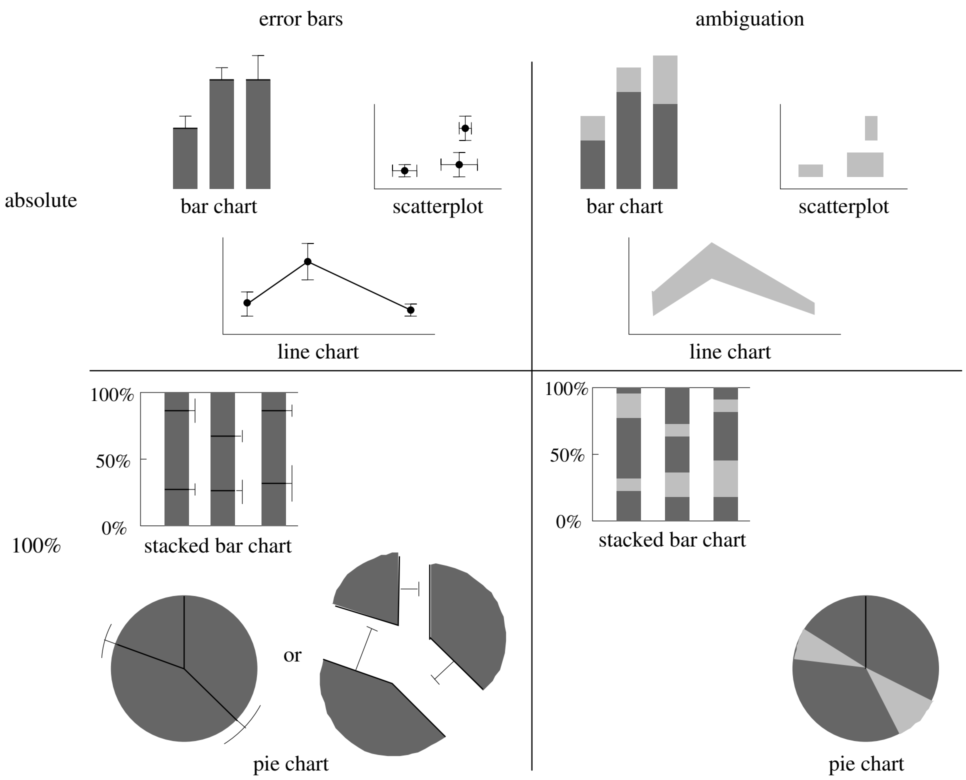

Figure 1: Error bars and ambiguation applied to some common chart types.

For each uncertain data value to be represented visually, the idea is to use the normal display technique to render the estimate E in place of the unknown exact value V. Error bars are then added to indicate uncertainty in the position or boundary location in proportion to the size of the confidence interval $[E - D, E + D]$. Some standard uses of error bars are illustrated in the upper left quadrant of Figure 1. When uncertainty occurs in bounded rather than statistical form, it is important to avoid the use of error bars since the accepted interpretation implies a potentially unbounded distribution extending beyond the error bars. Even worse, rendering an exact estimate using the normal display technique strongly implies the existence of a most likely value E, but in bounded uncertainty no most likely value can be assumed.

4.2 Ambiguation

To convey the presence and degree of bounded uncertainty, we propose the use of a technique we call ambiguation. The main idea behind ambiguation when uncertain data is encoded in the extent of a graphical element is to widen the boundary to suggest a range of possible boundary locations and therefore a range of possible extents. The ambiguous region between possible boundaries can be drawn as graphical fuzz, giving an effect that resembles ink smearing. A straightforward application of this technique is illustrated in the ambiguated bar chart in the upper right quadrant of Figure 1. To indicate positional uncertainty, rather than drawing a crisp representation of the graphical element at a particular position, the representation is elongated in one or more directions and drawn using fuzz. A simple application of this technique is illustrated in the ambiguated scatterplot in the upper right quadrant of Figure 1.

Other variations of boundary or position ambiguation may be possible, but the necessary feature is that no particular estimate or most likely value should be indicated. Rather, the entire range of possible values for the boundary or position of the graphical element should be presented with equal weight. This key characteristic is in contrast with error bars and other approaches such as fuzzygrams and gradient range symbols [Har99] that emphasize a known probability distribution over data values.

4.3 Discussion

The complementary use of error bars and ambiguation makes the presence, degree, and form of uncertainty clear. First, these techniques make it easy to identify the specific data values that are uncertain by suggesting imprecision in the graphical property (position or boundary location) in which the values are encoded. For bounded uncertainty, the position or boundary is made ambiguous using fuzzy ink, and for statistical uncertainty, error bars are added to visually suggest the possibility of a shift in position or boundary location. Second, these techniques allow the degree of uncertainty to be read in a straightforward manner using the same scale used to interpret the data itself. Finally, the use of two visually distinct techniques makes it clear which of the two forms of uncertainty is present, and each technique conveys the properties of the form of uncertainty it represents.

Ambiguation and error bars work well when data is encoded as the position or extent of graphical elements. Coping with displays that use other graphical attributes such as color and texture to encode data is left as a topic for future work. In the absence of analogous techniques for other graphical attributes, when uncertainty is present it is desirable to only use charts and graphs that encode data using position and extent alone so the presence, degree, and form of uncertainty can be clearly and unambiguously depicted.

4.4 Application to Common Chart Types

Figure 1 illustrates how error bars and ambiguation can be applied to some common chart types (exhaustive illustration on all known chart types is omitted for brevity). While these techniques are general and can be applied to a broad range of displays that use position and extent to encode data, we focus on abstract charts and graphs, which can be classified into two categories: *absolute displays* and *100% displays*. In absolute displays, each data value is given a graphical representation whose extent or position is plotted on an absolute scale. Examples of absolute displays include simple bar charts (which encode data in the upper boundaries of bars), scatterplots (which encode data in the positions of points), and line graphs (which encode data in the positions of points and lines). It is generally straightforward to add error bars or apply ambiguation to boundaries and positions in absolute displays such as those displayed in the top half of Figure 1.[1]

In 100% displays, the scale ranges from 0% to 100%, and n values $V_1, V_2, \ldots, V_n$ are plotted on this relative scale. Each value V_i is plotted as a graphical element whose size is proportional to the fraction $\frac{V_i}{\sum_{1 \leq j \leq n} V_j}$ of the total over all n values. Examples of 100% displays include stacked bar charts and pie charts. Indicating uncertainty in 100% displays is more challenging than doing so in absolute displays. In 100% displays, the graphical elements usually contact each other directly, so the boundary between two elements indicates the difference between them in terms of relative contribution to the total. To inform the user of statistical uncertainty in the locations of these boundaries, error bars can be drawn adjacent to the boundaries. Alternatively, for pie charts the wedges can be separated, leaving space for error bars extending directly from the boundaries between wedges. The lower left quadrant of Figure 1 illustrates these techniques. Bounded uncertainty can be indicated by inserting an ambiguous region of fuzzy ink between each pair of elements whose shared boundary is uncertain, as illustrated in the lower right quadrant of Figure 1. It turns out that determining the sizes to use for the fuzzy and solid regions in an ambiguated 100% display is not trivial because each region of fuzz shares a border with two solid data regions. We address this challenge next.

4.4.1 Ambiguation in 100% Displays

Here we consider how to draw an ambiguated pie chart (the formulation for a stacked bar chart is similar). Suppose the data for the pie chart consists of an ordered list of n objects $O_0, O_1, \ldots, O_{n-1}$ whose values are known to lie inside the intervals $[L_0, H_0], [L_1, H_1], \ldots, [L_{n-1}, H_{n-1}]$, respectively. As a first step, the absolute uncertainty intervals need to be converted into relative ones that indicate the smallest and largest possible fraction of the chart covered by each data object:

$$L_i^r = \frac{2\pi \cdot L_i}{\sum_{j=0}^{n-1} H_j - H_i + L_i} \qquad H_i^r = \frac{2\pi \cdot H_i}{\sum_{j=0}^{n-1} L_j - L_i + H_i}$$

The smallest possible fraction L_i^r occurs when the value of O_i is as low as possible, *i.e.*, equal to L_i, and the values of all other objects $O_j \neq O_i$ are as high as possible, *i.e.*, equal to H_j. The rationale for H_i^r is symmetric.

Ideally, an allocation of fuzzy and solid ink that conveys the uncertainty exactly could be found, so that each data object O_i has a corresponding solid pie wedge of arc length L_i^r (in radians) and two adjacent fuzzy wedges of total arc length $H_i^r - L_i^r$. For example, suppose we wish to draw a pie chart for two data objects, each with values in the interval $[1, 2]$, and thus relative contributions of between $\frac{1}{3}$ and $\frac{2}{3}$ each. A simple chart with two solid wedges of arc length $\frac{2\pi}{3}$ each plus a fuzzy wedge also of arc length $\frac{2\pi}{3}$ achieves the ideal of conveying exactly the uncertainty intervals present in the data.

Unfortunately, due to the nature of 100% displays, this ideal is not always achievable. In some cases it is not possible to convey the exact uncertainty intervals. For example, suppose we wish to draw a pie chart for three data objects with identical intervals of $[1, 2]$, or relative contributions of between $\frac{1}{5}$ and $\frac{1}{2}$ each. To indicate the smallest possible relative contribution of each data object, we need three solid wedges of arc length $\frac{2\pi}{5}$ each. The three gaps between the wedges will be filled with fuzz. We denote the arc lengths of the three fuzzy regions as F_0, F_1, and F_2. No matter how we arrange the solid wedges, the combined arc length of the fuzzy regions is $F_0 + F_1 + F_2 = 2\pi - 3 \cdot \frac{2\pi}{5} = \frac{4\pi}{5}$. To convey that the maximum possible relative contribution of each data object is $\frac{1}{2}$, we need to arrange the three solid wedges, with fuzz in between, such that each solid wedge taken together with the two adjacent fuzzy regions spans a total arc length of π (half circle). Therefore, we require that $F_0 + \frac{2\pi}{5} + F_1 = F_1 + \frac{2\pi}{5} + F_2 = F_2 + \frac{2\pi}{5} + F_0 = \pi$. We have defined a system of four equations with three unknowns (F_0, F_1, and F_2) that is overconstrained and has no solution. Therefore, an arrangement of solid wedges that conveys the exact uncertainty intervals that occur in the data does not exist in this case.

The only way to draw an ambiguated pie chart in such cases is to approximate the level of uncertainty that occurs in the data, which is undesirable but necessary. In general, consider the task of creating a pie chart with ambiguation for a data set consisting of n objects $O_0, O_1, \ldots, O_{n-1}$. Let $S_i \geq 0$ be the new arc length of the solid portion of the display for O_i, and let $F_i \geq 0$ be the arc length of the fuzzy region between the solid region for the two objects O_i and $O_{(i+1) \bmod n}$. An optimization problem arises with the goal of minimizing the total amount of fuzz without giving false information, *i.e.*, without drawing a chart indicating uncertainty intervals that do not contain the actual uncertainty intervals intrinsic in the data: Minimize $\sum_{i=0}^{n-1} F_i$ such that $\sum_{i=0}^{n-1} S_i + \sum_{i=0}^{n-1} F_i = 2\pi$ and for all i: $S_i \leq L_i^r$ and $F_{(i-1) \bmod n} + S_i + F_i \geq H_i^r$.

The objective function minimizes the total amount of fuzz, which in turn minimizes the overall loss in accuracy due to the use

[1] Displays such as stacked bar charts that are not normalized to sum to 100% are problematic when used in conjunction with these uncertainty indicators because the interpretation can be ambiguous. For example, in an absolute stacked bar chart, an error bar or a fuzzy region appearing at the top of the stack can be interpreted either as uncertainty in the topmost element or as uncertainty in the overall height of the stack (the sum over all elements).

of approximation. The first constraint requires that all of the solid and fuzzy wedges placed together create a full circle. The second constraint ensures that, for each object, the arc length of the solid region is no larger than the minimum relative contribution L_i^r of the object's value to the total. The third constraint ensures that, for each object, the combined arc length of the solid region taken together with the two adjacent fuzzy regions is no smaller than the maximum relative contribution H_i^r of the object's value to the total. The second and third constraints together ensure that the uncertainty intervals implied by the chart contain the actual uncertainty intervals they approximate.

We have implemented an ambiguated pie chart renderer that invokes a publicly available linear program solver to solve this optimization problem and determine the best possible pie chart layout to minimize the loss in accuracy due to displaying approximate uncertainty intervals. It runs in under 10 milliseconds on a modest workstation for data sets of 25 objects, which is large for a 100% chart. It is therefore suitable for execution as a part of an interactive rendering cycle. In certain extreme cases where a data set exhibits a great deal of uncertainty, the linear program has no solution, and it is impossible to generate an ambiguated 100% chart, even by approximating the uncertainty intervals. This problem can sometimes be resolved by reordering the wedges, but this method may cause a disconcerting effect in dynamic displays of changing data. Instead, we advocate using a special icon when no pie chart can be drawn to indicate extreme uncertainty. The user can react by requesting a decrease in uncertainty to make the chart displayable, using the interface proposed below in Section 6.

5 Avoiding Misleading Sudden Jumps

When the graphical display is refreshed due to a change in the underlying data, sudden jumps in the displayed data will tend to draw the user's attention. This characteristic is usually appropriate when the jump corresponds to a drastic change in the data. However, consider the case where the source of uncertainty is approximate caching, as discussed in Section 3.3. In approximate caching, systems considerations, rather than semantic events, trigger the source to refresh the cached interval, so sudden jumps in the graphical display of data and uncertainty level may not correspond to significant changes in the underlying data. Moreover, the absence of jumps may not rule out changes. Therefore, in visualization applications that receive data feeds from approximate caching protocols, it is desirable to mask sudden jumps in the data or uncertainty level that would inappropriately draw the user's attention. Sudden jumps can be masked by smoothly animating the transitions between old and new data and uncertainty values at a slow enough rate to not be overly distracting.

6 Controlling the Degree of Uncertainty

In environments where the data being visualized is obtained from an approximate caching or progressive sampling source, there is often an opportunity to exert control over the uncertainty levels or the rate at which the uncertainty improves at a per-object granularity, as discussed in Section 3. In these paradigms, a decrease in the uncertainty of some data objects is offset by either a corresponding increase in the uncertainty of other data objects or an increase in system resource utilization. Therefore, it may be desirable for the visualization system to mediate control over the uncertainty levels. One way to perform this mediation is for the visualization system to maintain default uncertainty levels[2] that are either uniform, lower for graphical elements near the center of the screen, or lower for elements that have remained on the screen for a long time. In some situations where bounded uncertainty is displayed, it may also be appropriate to require that uncertain regions do not overlap in the dimension(s) of interest, making the relative order of data values always discernible. At any point, the user can override the default uncertainty levels by clicking on an area of interest, causing that area to "come into focus" via a decrease in uncertainty. Then, the visualization system should gradually return the uncertainty levels to their default state.

In progressive sampling or constrained resource approximate caching scenarios (see Sections 3.2.1 and 3.3, respectively), the result of bringing one region of data "into focus" will be increased uncertainty (or slower rates of uncertainty improvement) for the rest of the display. However, under constrained uncertainty caching (see Section 3.3), there is no limit to how much the overall uncertainty can be reduced, but lower uncertainty incurs a higher communication cost. It may be important to convey the cost to the user via a network traffic status indicator, for example, so that the increase in traffic resulting from requesting lower uncertainty levels is indicated visually. Noticing the increase in traffic, the user could click on the network indicator to reduce traffic again by affecting a mild increase in uncertainty across the board. If the network traffic indicator does not provide adequate incentive for the user to be sparing when lowering uncertainty levels, it may be appropriate to have uncertainty levels continually increase by default. This property would force the user to click periodically to maintain data in sharp focus, requiring effort commensurate with the amount of work required of the network infrastructure thereby communicating the cost to the user. Furthermore, if the user abandons the visualization for, say, a coffee break without closing the application, the display would eventually become entirely out of focus, incurring no network costs while the visualization is not being used.

7 Summary

We have identified and treated three issues that arise in visualizing data with uncertainty. The first issue stems from the fact that uncertainty comes in two predominant forms: statistical uncertainty and bounded uncertainty. Since the two forms of uncertainty have quite different meanings, it is important to avoid the use of visual indicators that can be misinterpreted as representing the wrong form of uncertainty. To address this issue, we advocated the use of two distinct techniques for indicating the degree of uncertainty associated with each graphical data element: error bars and ambiguation. Each technique is well suited to the form of uncertainty it is intended to represent. We showed how to apply these techniques systematically to common charts and graphs, which in some cases requires displaying approximations to the uncertainty

[2] Users of approximate caching systems might wish to modify the default uncertainty level of a particular data object by specifying the lower and upper endpoints of the uncertainty interval, causing the display of that object to remain static until the data value moves beyond one of the endpoints. This feature can serve to alert users when a data value exceeds a critical threshold.

levels. To handle those cases, we specified an algorithm that finds displayable approximations while minimizing the overall loss in accuracy.

The second issue we addressed is that sudden jumps in the graphical display of data and uncertainty level can be misleading in some cases. We proposed a remedy that involves smoothly animating data transitions to mask sudden jumps. Finally, the third issue arises in the context of emerging interactive data delivery paradigms that exhibit the convenient property that uncertainty can be controlled at a fine granularity by the visualization application. Although uncertainty can be controlled, decreasing the uncertainty is not free, and results in either increased uncertainty elsewhere or increased computational and communication costs. This property motivated us to propose interfaces that offer control of uncertainty levels to the user in ways that encourage careful use of these facilities.

References

[AHL+98] R. Avnur, J. M. Hellerstein, B. Lo, C. Olston, B. Raman, V. Raman, T. Roth, and K. Wylie. CONTROL: Continuous output and navigation technology with refinement on-line. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 567–569, Seattle, Washington, June 1998.

[BP93] R. S. Barga and C. Pu. Accessing imprecise data: An approach based on intervals. *IEEE Data Engineering Bulletin*, 16(2):12–15, June 1993.

[Cle85] W. S. Cleveland. *The Elements of Graphing Data*. Wadsworth, 1985.

[CR00] A. Cedilnik and P. Rheingans. Procedural annotation of uncertain information. In *Proceedings of the IEEE Visualization Conference*, pages 77–84, Salt Lake City, Utah, October 2000.

[DK97] T. J. Davis and C. P. Keller. Modelling and visualizing multiple spatial uncertainties. *Computers and Geosciences*, 23(4):397–408, 1997.

[DKP+01] P. Deolasee, A. Katkar, A. Panchbudhe, K. Ramamritham, and P. Shenoy. Adaptive push-pull: Disseminating dynamic Web data. In *Proceedings of the Tenth International World Wide Web Conference*, Hong Kong, China, May 2001.

[FWR99] Y. Fua, M. O. Ward, and E. A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 58–64, San Francisco, California, October 1999.

[GCB+97] J. Gray, S. Chaudhuri, A. Bosworth, A. Layman, D. Reichart, M. Venkatrao, F. Pellow, and H. Pirahesh. Data cube: A relational aggregation operator generalizing group-by, cross-tab, and subtotals. *Data Mining and Knowledge Discovery*, 1(1):29–53, 1997.

[HAC+99] J. M. Hellerstein, R. Avnur, A. Chou, C. Hidber, C. Olston, V. Raman, T. Roth, and P. Haas. Interactive data analysis with CONTROL. *IEEE Computer*, August 1999.

[Har99] R. L. Harris. *Information Graphics*. Oxford University Press, 1999.

[HH97] J. M. Hellerstein and P. J. Haas. Online aggregation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 171–182, Tucson, Arizona, May 1997.

[HSW94] Y. Huang, R. Sloan, and O. Wolfson. Divergence caching in client-server architectures. In *Proceedings of the Third International Conference on Parallel and Distributed Information Systems*, pages 131–139, Austin, Texas, September 1994.

[Mac92] A. M. MacEachren. Visualizing uncertain information. *Cartographic Perspective*, (13):10–19, 1992.

[OW00] C. Olston and J. Widom. Offering a precision-performance tradeoff for aggregation queries over replicated data. In *Proceedings of the Twenty-Sixth International Conference on Very Large Data Bases*, pages 144–155, Cairo, Egypt, September 2000. (Extended version available at http://www-db.stanford.edu/pub/papers/trapp-ag.ps.).

[OW02a] C. Olston and J. Widom. Approximate caching for continuous queries over distributed data sources. Technical report, Stanford University Computer Science Department, 2002. http://dbpubs.stanford.edu/pub/2002-8.

[OW02b] C. Olston and J. Widom. Best-effort cache synchronization with source cooperation. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 73–84, Madison, Wisconsin, June 2002. (Extended version available at http://www.db.stanford.edu/ olston/publications/bes.pdf.).

[PWL97] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, pages 370–390, November 1997.

[TCS94] R. Twiddy, J. Cavallo, and S. M. Shiri. Restorer: A visualization technique for handling missing data. In *Proceedings of the IEEE Visualization Conference*, pages 212–216, Washington, D.C., October 1994.

[TK94] B. N. Taylor and C. E. Kuyatt. Guidelines for evaluating and expressing the uncertainty of nist measurement results. Technical report, National Institute of Standards and Technology Note 1297, Gaithersburg, Maryland, 1994.

[Tuf01] E. R. Tufte. *The Visual Display of Quantitative Information, Second Edition*. Graphics Press, 2001.

[Tuk77] J. W. Tukey. *Exploratory Data Analysis*. Addison Wesley, 1977.

[vdWvdGG98] F. J. M. van der Wel, L. C. van der Gaag, and B. G. H. Gorte. Visual exploration of uncertainty in remote-sensing classification. *Computers and Geosciences*, 24(4):335–343, 1998.

[WO98] A. Woodruff and C. Olston. Iconification and omission in information exploration. In *SIGCHI '98 Workshop on Innovation and Evaluation in Information Exploration Interfaces*, Los Angeles, CA, April 1998. http://www.fxpal.com/ConferencesWorkshops/CHI98IE/submissions/woodruff.htm.

[WSF+96] C. M. Wittenbrink, E. Saxon, J. J. Furman, A. T. Pang, and S. K. Lodha. Glyphs for visualizing uncertainty in environmental vector fields. *IEEE Transactions on Visualization and Computer Graphics*, pages 266–279, September 1996.

[YV00a] H. Yu and A. Vahdat. Design and evaluation of a continuous consistency model for replicated services. In *Proceedings of the Fourth Symposium on Operating Systems Design and Implementation*, San Diego, California, October 2000.

[YV00b] H. Yu and A. Vahdat. Efficient numerical error bounding for replicated network services. In *Proceedings of the Twenty-Sixth International Conference on Very Large Data Bases*, pages 123–133, Cairo, Egypt, September 2000.