Stacking Graphic Elements to Avoid Over-Plotting

Tuan Nhon Dang, Leland Wilkinson, and Anushka Anand

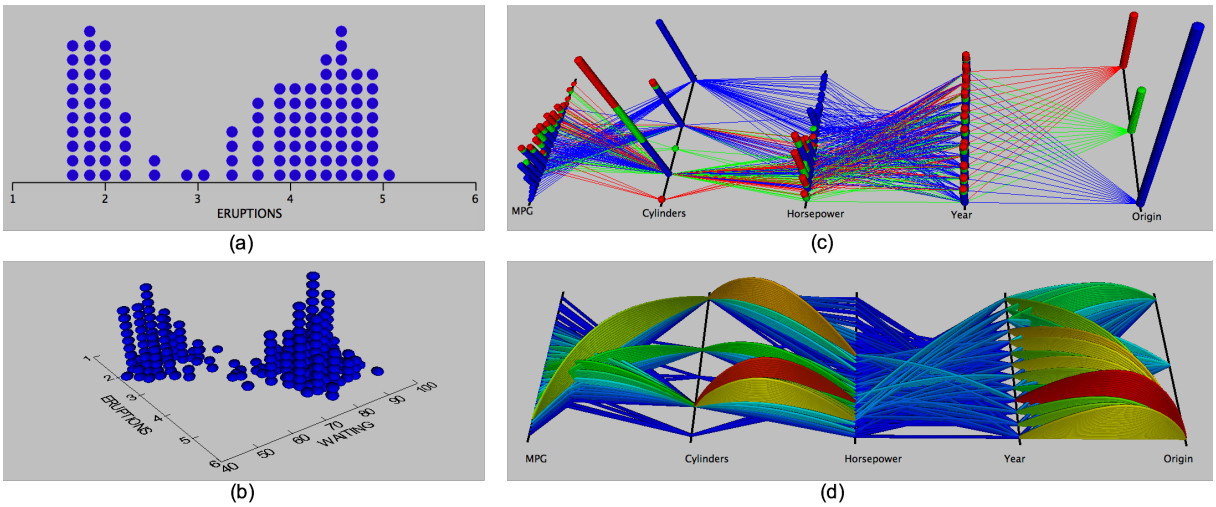


Fig. 1. Visualization of Old Faithful data: (a) Single-variable dot plot; (b) Two-variable dot plot; Visualization of Cars data: (c) Parallel Coordinate Dot Plot; (d) Stacked Parallel Coordinate Plot.

Abstract—An ongoing challenge for information visualization is how to deal with over-plotting forced by ties or the relatively limited visual field of display devices. A popular solution is to represent local data density with area (bubble plots, treemaps), color (heatmaps), or aggregation (histograms, kernel densities, pixel displays). All of these methods have at least one of three deficiencies: 1) magnitude judgments are biased because area and color have convex downward perceptual functions, 2) area, hue, and brightness have relatively restricted ranges of perceptual intensity compared to length representations, and/or 3) it is difficult to brush or link to individual cases when viewing aggregations. In this paper, we introduce a new technique for visualizing and interacting with datasets that preserves density information by stacking overlapping cases. The overlapping data can be points or lines or other geometric elements, depending on the type of plot. We show real-dataset applications of this stacking paradigm and compare them to other techniques that deal with over-plotting in high-dimensional displays.

Index Terms—Dot plots, Parallel coordinate plots, Multidimensional data, Density-based visualization.

1 Introduction

Overplotting instances in scatterplots, parallel coordinates, and other displays has been a longstanding problem in information visualization. Data points with similar or identical values partially or completely overlap in a scatterplot. The same data points overlap in a parallel coordinate plot and in many other multivariate displays. We tackle this problem with a strategy that stacks representation elements (symbols, lines, etc.) in an additional dimension based on density or counts. The height of the stack shows the density in the neighborhood of a data value.

Several other solutions to this problem have been been devised over several centuries of charts and graphics. One approach is to aggregate cases in local neighborhoods and to use area to represent density. For example, we can bin cases inside political, geographic, or abstract regions and represent counts by the size of circles or length of bars superimposed on a map of those regions [7]. The so-called "bubble plot" was invented by Playfair [25] and has been used for over two centuries. Unfortunately, sizing symbols to represent aggregates ameliorates but does not eliminate overlapping; bubbles or other representation elements can still cover each other when their sizes exceed nearest-neighbor distances. Furthermore, using area to represent magnitude incurs nonlinear distortions in perception [8, 38, 32].

Another approach is to partition a space, aggregate within partitions, and adjust the size of partitions in order to represent counts [20, 10]. This approach prevents overlaps but risks displacing or distorting density locations. Furthermore, using polygon size to represent the count of points within a partition has the same nonlinear perceptual problems that arise with bubble plots.

Another approach is to aggregate and use color to represent the magnitude of the aggregation [5, 21, 14]. The most common instance of this approach is the choropleth map. Another approach is to use kernel density estimation or other forms of smoothing and then represent the smoothed density with a color or intensity gradient. These approaches risk distortion from using color to represent a continuum. Lightness/darkness has a concave downward psychometric function [38] and representing a linear scale with hue is problematic [4].

Another approach is to dispense with aggregation and to use random local displacement to prevent overlap, commonly called *jittering* [6]. This is a useful method for representing bootstrap estimates in a

- *Tuan Nhon Dang is with Department of Computer Science, University of Illinois at Chicago, E-mail: tdang@cs.uic.edu.*
- *Leland Wilkinson is with SYSTAT Inc., Department of Computer Science, University of Illinois at Chicago, and Department of Statistics, Northwestern University, E-mail: leland.wilkinson@systat.com.*
- *Anushka Anand is with Department of Computer Science, University of Illinois at Chicago, E-mail: aanand2@lac.uic.edu.*

Manuscript received 31 March 2010; accepted 1 August 2010; posted online 24 October 2010; mailed on 16 October 2010.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

common space [38]. For smaller datasets, this approach works rather well. Plotting area is quickly overwhelmed as N increases, consequently, jittering has only limited application.

Another approach is to use transparency/opacity (alpha channel) to represent density. This method is attractive because alternative density estimators, such as kernel density estimation or histograms take longer to compute. A scatterplot of semi-transparent point symbols whose diameters are roughly a tenth of the frame width resembles a kernel density plot with saturation used to represent the height field for the density.

Many of these approaches have been applied to parallel coordinates, where over-plotting lines can create the same problems as over-plotting points. Solutions have involved kernel smoothing methods [12, 23] and 3D parallel coordinates[29][18, 16].

A serious drawback with all of these methods is that they break the mapping between individual points and locations on the screen or display area. The consequence is that brushing and linking are problematic [2] (although see [36] for some other remedies for this drawback).

By contrast, we propose using stacking to provide a view of dense data that reveals a range of magnitude considerably larger than what is available in color and area representations. We have also developed new stacking methods for extending these representations to 3D. We are not enthusiastic advocates of the use of 3D, but we show examples of appropriate use of stacking where 2D representations of magnitude fail.

The main contributions of this paper are:

- We revise the single-variable dot plot algorithm [37] and extend it to multi-dimensional space.
- We extend stacking dots to stacking lines and show how this helps solve a problem with parallel coordinates and other case-wise line plots.
- We show how stacking can be used to represent larger datasets without disabling brushing and linking.

This paper is organized as follows: Section 2 reviews related work in the field of density estimation. Section 3 describes and discusses our stacking algorithms for two and three dimensions. Section 4 extends stacking dots to stacking lines. We present our conclusions in Section 5.

2 Related Work

While the oldest form of stacking is probably the simple tally, the approaches we introduce in this paper are related to representations of density such as histograms and kernels. Histograms are the classic method for displaying densities [25, 24]. Their virtues are ease of computation (one or two passes through the data suffice) and interpretability (histogram bars allow visual estimates of counts or density within intervals). Their drawbacks are statistical efficiency (histograms do not closely approximate population distributions in smaller samples), scale sensitivity (histogram shapes change when cutpoints vary), and location bias (histogram bars are not necessarily centered on regions of high conditional density). Kernel density estimates overcome these deficiencies [27, 28]. Rather than grouping observations together in bins in the way a histogram does, the kernel density estimator centers a kernel probability function at each observation. The estimator consists of the sum of these functions; if the functions are smooth (the usual case) then the kernel density is smooth, unlike the histogram. The virtues of kernel density estimators compared to histograms are their unbiasedness and efficiency. Their drawbacks are computational complexity (particularly in higher-dimensional spaces) and their dependence on a bandwidth parameter. Choice of bandwidth is not a serious drawback; however, adaptive methods have proven effective in practice [31].

Dot plots offer a third approach to density representation. Similar to tallies, dot plots stack small dots on top of each other to represent conditional counts. Their advantages include a natural framework for brushing and linking (each dot represents an individual case or instance), low location bias (dots are centered on data values, unlike histograms), flexiblility (dot plots work for both continuous or categorical variables), and sensitivity (dot plots reveal granularity in integer or rounded data, unlike histograms or kernels).

Dot plots have been used for over a century in the medical, economic, and scientific literature (e.g., [15, 22]). They were originally hand-drawn by medical and scientific illustrators. After Wilkinson [37] introduced an algorithm and computer program to render dot plots automatically, these plots began to appear in a few statistical packages, such as, SYSTAT, SPSS, Stata, and R. Other packages (such as BMDP and Minitab) have offered graphics that appear to be dot plots but are instead simply histograms with stacks of dots replacing the bars. It is easy to distinguish "fake" dot plots from the real thing. Simply input irregularly spaced data and observe whether the stacks are evenly spaced. If they are, the graphic is not a true dot plot.

3 Stacking Dots

The remainder of this paper covers several different types of stacking, inspired by the original formulation for dot plots. We begin with a technical introduction to the dot plot algorithm and then extend this algorithm to multiple dimensions. We provide several examples on real datasets to illustrate the usefulness of these extensions. In the next section, we introduce new methods for stacking lines and show how these can enhance the usefulness of parallel coordinate plots and other displays.

3.1 One Variable

This section covers the original dot plot algorithm introduced in [37] and shows a modification that reduces directional bias in the placement of dots. This modification also enables the extension of dot plots to higher dimensions.

3.1.1 Single-variable Dot Plot Algorithm

Figure 2 shows a dot plot for a bimodal dataset. The observations are the duration of eruptions in minutes for the Old Faithful Geyser in Yellowstone National Park, Wyoming, USA [27].

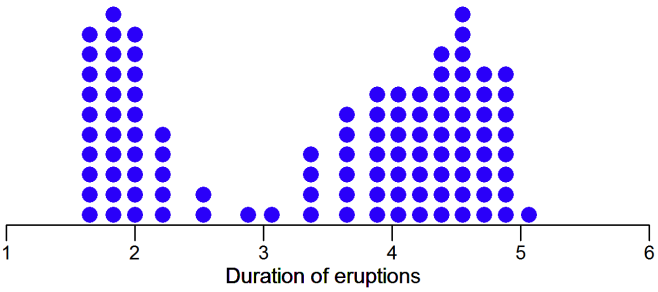


Fig. 2. Dot plot of the duration in minutes of Old Faithful eruptions.

In its simplest form, the algorithm in [37] for producing this plot orders a set of observations $\{X_1, X_2, ..., X_n\}$ and starts with the smallest data value X_1. For any X_i, we look below it within a dot-size neighborhood and stack the dots representing the observations belonging to this neighborhood. The algorithm moves to the next observation not included in the current dot stack until there is no observation left.

The size of the dots is crucial to the visual appearance of stacked dot plots. Large dots can over-smooth the data, while small dots may highlight too many random details. Choice of dot size is discussed in Wilkinson's original paper [37], which relates dot size to the bar-width of histograms and bandwidth of kernel density estimators. Stacker picks a default dot size based on the guidelines in Wilkinson's paper. A slider is available for adjusting the dot size in real time to view more or less detail.

3.1.2 Improved Version of Single-variable Dot Plot Algorithm

There are three limitations of the original algorithm:

- This algorithm produces dot plots by moving from the smallest data point X_1 to the largest data point X_n (from left to right). In like manner, dot plots can be produced by moving in reverse order (from right to left). These different moving directions produced different dot plots on the same batch of data. This is not a desirable consequence.
- Symmetrically distributed data may result in an asymmetric dotplot.
- Multidimensional dot plots cannot be extended from this algorithm because it is intrinsically unidimensional along a single numerical axis.

We have devised an undirected algorithm that overcomes the drawbacks of the original dot plot algorithm. Instead of starting with the smallest data value (moving from left to right) or the largest data value (moving from right to left), we begin with a data point that has the maximum number of neighbors in a dot-radius neighborhood. Here is a summary of the algorithm:

1. For each data point X_i, create a set of its neighbors C_i in a dot radius neighborhood $h/2$ including itself.
2. Instantiate a set of data points that have not been considered, $D = \{X_1, X_2, ..., X_n\}$.
3. Start with a data point $X_l \in D$ that has the maximum number of neighbors.
4. Place $|C_l|$ dots above the center of this set c_l, where $c_l = (max(C_l) + min(C_l))/2$.
5. Update D: $D = D - C_l$.
6. Remove all elements of C_l from the set of neighbors of other data points in D.
7. Move to the next data point $X_l \in D$ that has a maximum number of neighbors.
8. Repeat steps 4-7, until there are no data values left to plot (D is empty).

Figure 3 compares this algorithm to the original for a symmetric dataset with 5 observations. The first panel shows dot plots produced by a left-to-right implementation of the original algorithm, the second panel shows dot plots produced by a right-to-left implementation of the original algorithm, and the third panel shows dot plots produced by the undirected algorithm. The symmetry of the original data is revealed by the undirected algorithm.

3.1.3 Comparison of Dot Plots with Histograms and Kernel Density Estimates

To understand the strengths and weaknesses of dot plots, it helps to compare and contrast them with histograms and kernel density estimates. First of all, dot plots are better for revealing granularity in data. This can be important when examining integer-valued rating scales or rounded physical measurements for miscodings or unusual gaps. Specifically, a dot plot places the dots where the data values actually occur, so that they leave gaps between values empty for granular data [38]. In contrast, kernel densities smooth over gaps in the data, and histogram bins tend to cover gaps.

Second, histograms and kernel densities obscure local features because they aggregate. By contrast, the dots in dot plots can be labelled or colored with group values. Moreover, since dot plots represent individual observations by single symbols, they are suitable for brushing and linking.

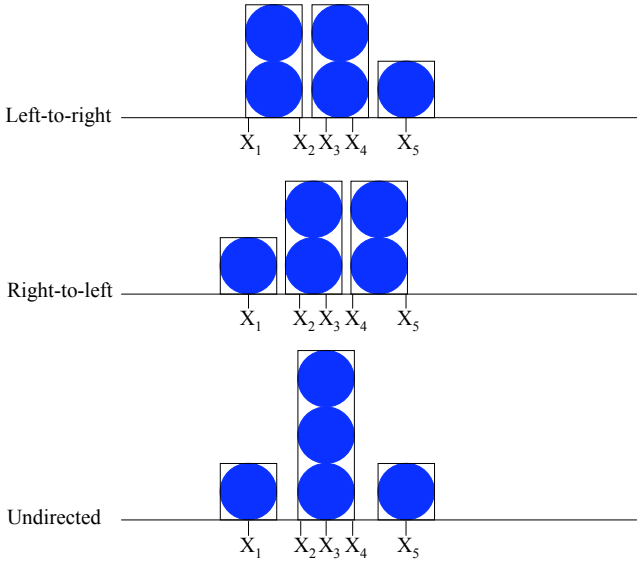


Fig. 3. Dot plots produced by different algorithms.

Third, dot plots lend themselves to highlighting outliers in a display because singletons can be labeled, colored, resized or reshaped to stand out. Histograms and kernels tend to conceal singleton outliers because they do not have the mass to affect the overall shape.

Finally, dot plots can be devised to handle large datasets by imposing an aggregation on each dot. This condition allows us to define a continuum between N dots (the traditional dot plot) and 1 dot (the whole batch aggregated into a single large dot). From a mathematical perspective, a dot is a ball in a metric (usually Euclidean) space whose radius determines the set of points within its boundary. In our software, we have implemented a slider that determines the dot radius and allows us to drill-down to different levels of detail.

3.2 Two Variables

The undirected dot plot algorithm is easily extended to the two-variable case. We simply compute Euclidean distance between points. Data point $P(p_1, p_2)$ belongs to the neighborhood of data point $Q(q_1, q_2)$ if

$$(p_1 - q_1)^2 + (p_2 - q_2)^2 \leq r^2 \quad (1)$$

where r is dot radius.

3.3 Software

The visualization testbed implemented in this paper, called Stacker, provides a number of controls to create and manipulate stacked plots. Figure 4 shows the console. The open data set contains utility usage by a Chicago family from 2002 to 2008 [19]. The variables plotted are electricity and water usage. We have distinguished summer and winter months by selecting red and blue from a color scale. Stacker offers a color slider for each category on a variable, so a user can pick contrasting colors. For a continuous variable, Stacker offers a number of linearized scales. The sliders below this panel control dot size, opacity, aspect ratio of the geometric representation elements (usually dots), and overlap of the elements. A button directly below these sliders allows the user to expend additional time to minimize overlaps iteratively. The bottom panel offers options for brushing and displaying axes and data points.

Figure 5 illustrates the use of these controls on the utility dataset. Figure 5(a) shows the result of using the slider to overlap dots. Figure 5(b) shows the same plot with ellipsoidal dots. Figure 5(c) shows transparent dots; the opacity was set to 0.3. This device helps to reveal embedded sub-densities; most dots are in the area of low water

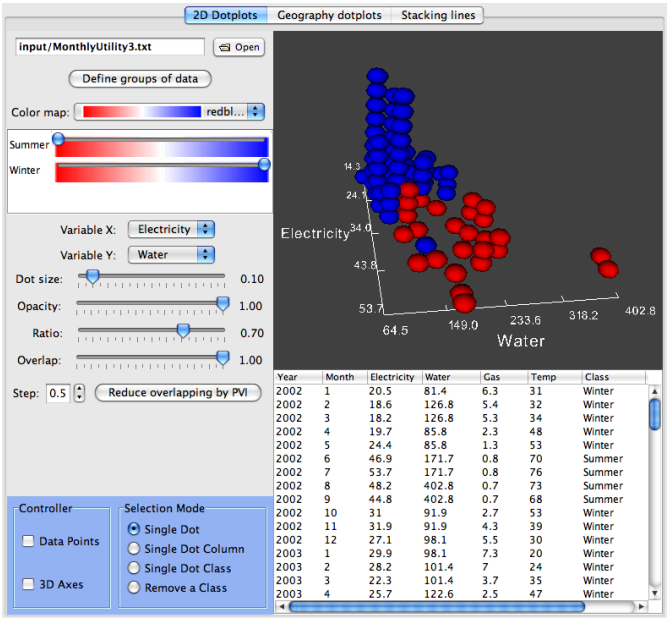


Fig. 4. Stacker console.

and low electricity usage. Figure 5(d) shows the plot colored to distinguish summer (red) and winter (blue) months. This plot shows that high water and high electricity usage occur in summers.

3.4 Geographic Dot Plots

This section describes an application in which 3D dot plots are used to show the distribution of Lyme disease in the United States. Dot plots are particularly suited to geographic applications because they stack where the data are measured (as opposed to choropleth maps, which are aggregations).

Figure 6(a) shows the distribution of Lyme disease cases reported in the U.S. in 2005, mapped by the CDC. As depicted, the Lyme disease distribution is dense in the Midwest and the East. Each case is represented by a black dot in the graph. The map is saturated; however, there is little density discrimination in the concentrated areas. Consequently, it is hard to compare the densities in the Midwest and East, and it is hard to see variations within the East distribution.

Dot plots overcome these limitations by representing density through dot stacks. Dots are stacked exactly over the counties where the Lyme disease cases are reported. Thus, one can visualize not only Lyme disease cases by region, but also the density distribution within a region.

Figure 6(b) shows the dot plot of Lyme disease distribution in 2005. One might argue, of course, that a county-level choropleth map would be a suitable alternative to the map in Figure 6(a). Nevertheless, it would be hard to argue that a county-level choropleth map using hue or brightness could reveal the threefold increase in Lyme disease cases centered around New England vs. the Midwest.

Dots can be colored by year as depicted in Figure 7. In particular, red dots are Lyme disease cases reported in 2005, green dots as reported in 2006, and blue dots as reported in 2007. Moreover, Stacker also supports geodata brushing as depicted in Figure 7. Users can browse over the dots; the region containing the dots is highlighted in the table. Similarly, users can select regions from the table and see them highlighted in the display. With brushing, users can easily compare Lyme disease densities among different regions. Furthermore, with Stacker, a user can select several regions for further analysis, while fading others from the view. In this way, one can reduce the effects of occlusion. In this display, data brushing can be done at the state or county level.

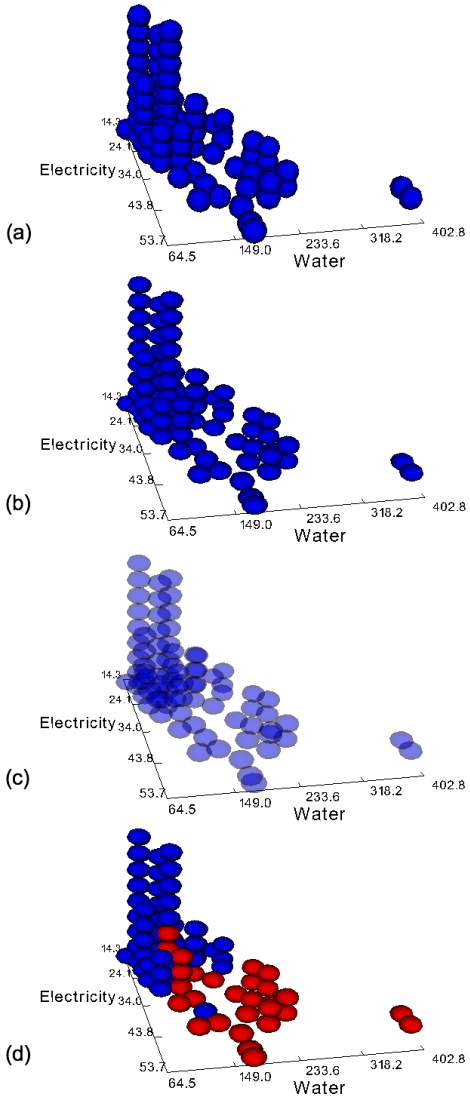


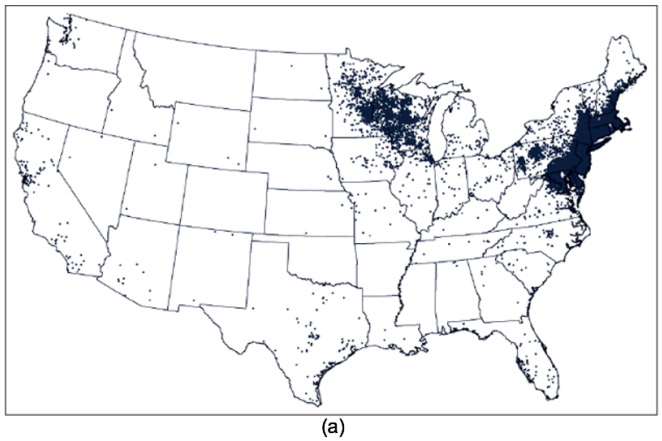
Fig. 5. Two dimensional dot plots of the data set: (a) Vertically overlapped dots; (b) Ellipsoidal dots; (c) Dot plot with opacity set to 0.3; (d) Colored dots distinguishing different groups.

3.5 Parallel Coordinate Dot Plots

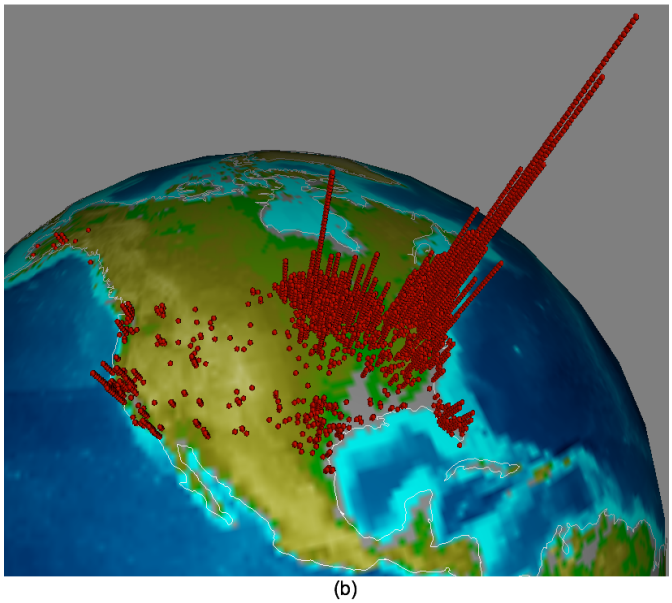
Parallel coordinate plots [13] have become a commonly used technique for providing insight into multivariate data. In parallel coordinates, each N-dimensional data point is transformed into a polyline that intersects N parallel axes at the respective value on that dimension. Parallel coordinate plots allow displaying high-dimensional data in one plot because the axes are visually separated from each other. For larger data sets, however, it is difficult to discern relative frequencies of subsets of points.

A number of extensions have addressed this problems. For the visualization of categorical variables, Parallel Sets [3] substitutes the individual data points at each axis by a frequency-based area element. However, this method works only with categorical variables. Another strategy is to replace regions of similar lines by a density gradient. Wegman [34] was the first to do this and many have followed [1, 17, 26, 12].

We have devised two alternatives. The first, *parallel dot plots*, is illustrated in Figure 8. This figure contains parallel coordinate plots of the Utility data used in Section 3.3. Polylines are colored by the Season categorical variable. Figure 8(a) shows a conventional parallel coordinate plot rendered by Stacker. Figure 8(b) depicts the parallel coordinate plot with a dot plot at each axis.



(a)



(b)

Fig. 6. Lyme disease distribution in 2005: (a) Scatterplot overlaid on map (b) Geographic dot plot.

States	2005	2006	2007
Alabama	20	6	0
Alaska	13	3	1
Arizona	11	11	0
Arkansas	1	0	0
California	93	93	69
Connecticut	2354	1756	1753
Delaware	688	466	596
Dist of Columbia	116	62	10
Florida	55	35	21
Georgia	18	7	0
Idaho	14	2	1
Illinois	161	112	113
Indiana	73	24	17
Iowa	144	97	68
Kansas	11	3	1
Kentucky	13	5	0
Louisiana	6	0	0
Maine	529	338	245
Maryland	2561	1250	1232
Massachusetts	2749	1404	2318
Michigan	80	54	34
Minnesota	1261	915	893
Missouri	26	2	2
Montana	5	0	0
Nebraska	13	5	2
Nevada	17	3	1
New Hampshire	896	617	265
New Jersey	3127	2430	3362
New Mexico	9	2	0
New York	4171	4469	5550
North Carolina	78	24	31
North Dakota	17	3	2
Ohio	59	44	31
Oklahoma	1	0	0
Oregon	12	3	1
Pennsylvania	3999	3253	4271
Rhode Island	177	308	35
South Carolina	41	13	7
South Dakota	3	0	0
Tennessee	38	10	4
Texas	111	36	38
Utah	9	4	1

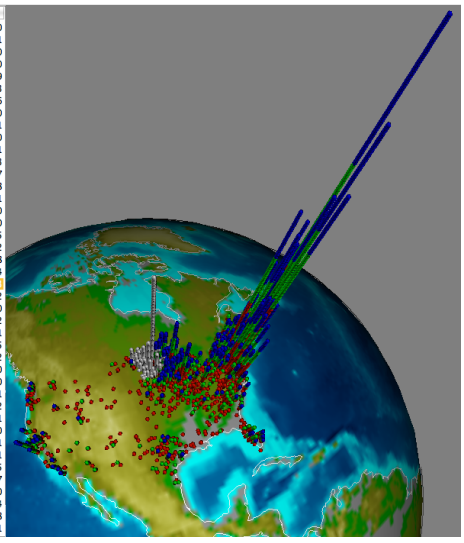
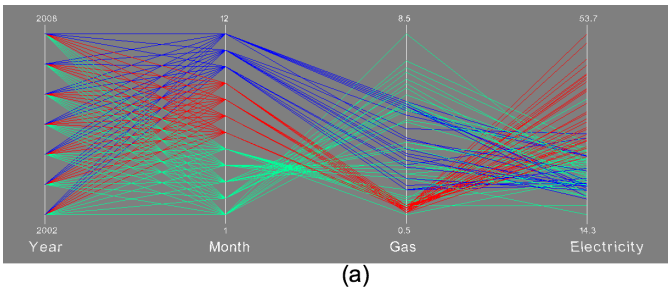
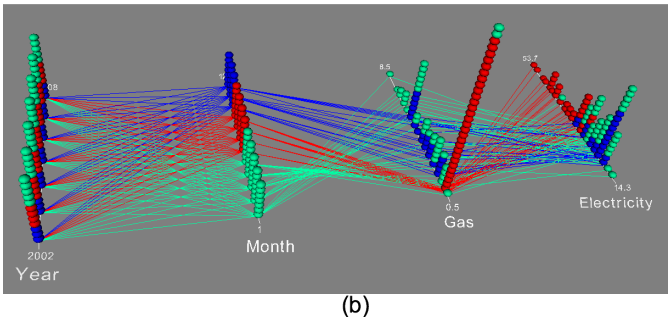


Fig. 7. Data brushing in dot plot presenting Lyme disease distribution.



(a)



(b)

Fig. 8. Views of the Utility data set: a) Conventional parallel coordinate plot; (b) Parallel coordinate dot plot.

Why do this? Does it add to the visual clutter of the display? Does 3D impair decoding? First of all, we are not suggesting to replace parallel coordinate plots with 3D parallel dot displays. We are instead advocating a pairing of the two types of display in the same application. Stacker has simple controls to switch back and forth between these representations. Second, we have arranged a mild perspective view and narrow dot towers to facilitate length-based magnitude comparisons and to minimize occlusion, both of which are significant problems for 3D bar graphs and other 3D displays. As can be seen in the figure, the dot towers highlight major differences in frequency that are not evident in the parallel coordinates display. Notice, for example, that low gas consumption in the summer months is immediately apparent in the dot display but not conspicuous in the ordinary parallel coordinate display.

Figure 9 shows data brushing on parallel dot plots. Data brushing can be done in several ways: one polyline, one dot, one dot stack, or one class of data. Specifically, Figure 9(a) depicts single *dot* brushing in which the polyline and dots associated with that record are highlighted (the record selected is March 2008). Figure 9(b) depicts single *stack* brushing (the dot stack selected contains months with very low gas usage). Figure 9(c) depicts single *class* brushing (the class selected is winter of every year from 2002 to 2008).

4 Stacking Lines

A second approach we have devised involves stacking lines instead of dots. For parallel coordinate plots and other types of line plots, we have implemented a method that extends the dot-stacking algorithm to line elements. As with dots, we can stack either single lines per case or stack lines aggregating subsets of cases. In this section, we will show two different stacking strategies for parallel coordinate plots and an example of stacking regression lines.

4.1 Stacked Parallel Coordinate Plots

Figure 10 shows several parallel coordinate displays of the Pollen dataset (`http://lib.stat.cmu.edu/datasets/pollen.data`). This artificial dataset, used in a competition at the 1986 Joint Meetings of the American Statistical Association, was given its name as well as fake variable names in order to mislead analysts into thinking it had to do with biology. Instead, the dataset contains a cluster of points spelling the word EUREKA surrounded by random points in five dimensions. Nowadays, the Pollen data are used to illustrate the

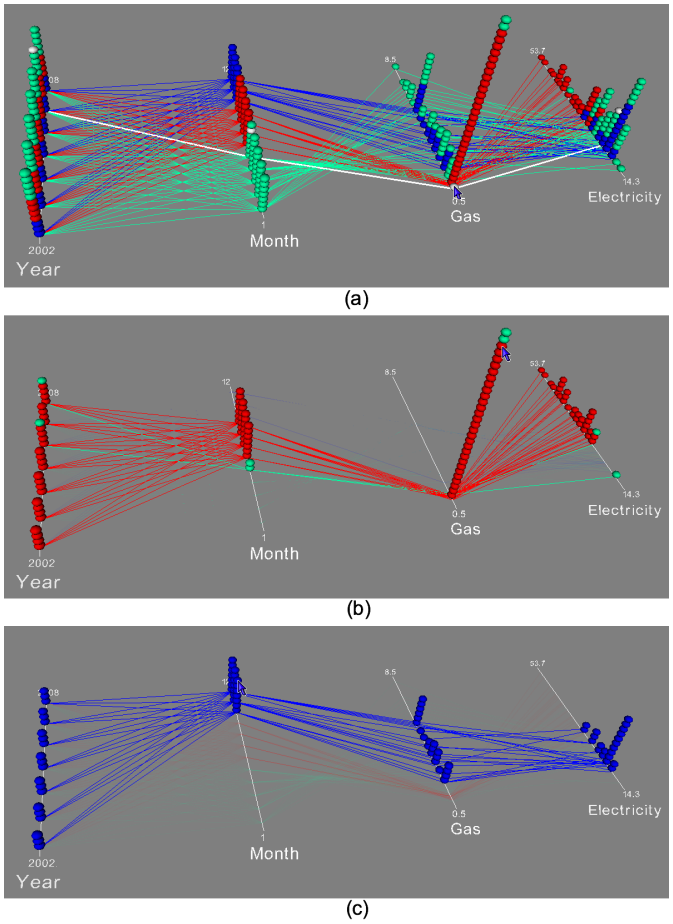


Fig. 9. Data brushing on parallel coordinate dot plot of the Utility data set: (a) One dot brushing; (b) One dot stack brushing; (c) One class of data brushing.

effectiveness of panning and zooming in 3D visualization programs. This effectiveness depends on knowing that EUREKA is spelled at the centroid and knowing that zooming and rotating around this position can reveal the hidden word. The real challenge involves discovering the structure without knowing this information.

Figure 10(a) shows an arbitrary 2D view of the data in the form of a scatterplot. Figure 10(b) is an ordinary parallel coordinate plot of the data. In both of these graphics, it is impossible to discern the core word. Figure 10(c) shows the same display with a density gradient devised by Wegman[35]. A faint line in the center gives us a clue that there may be a subset of points with its own distribution. Figure 10(d) shows a more recent version of Wegman's display [1]. The density gradient has been modified to make the middle line more salient.

Figure 10(e) shows a stacked parallel coordinate plot of the same data where stacks are colored based on their heights. One easily recognizes the central cluster comprising EUREKA. This is a clear illustration of the fact that stacking enables a larger range of magnitude representation than intensity. As we have seen with the Lyme dataset, it is difficult to construct a linear brightness gradient that has a resolution throughout its range sufficient to detect extreme values [8].

From Figure 10(e), we selected the central cluster in the stacked parallel coordinate display and requested plots of these data in new windows: Figure 10(f) zooms in on the central cluster; the polylines are colored by stack height. One can detect six smaller clusters in the central cluster. Figure 10(g) shows a scatterplot of data in the six clusters. Zooming in on this collection of points, we see the word EUREKA spelled in the scatterplot. Additionally, the data points are colored by the stacks which they belong to. In just three gestures, we have identified the code at the center of the scatterplot.

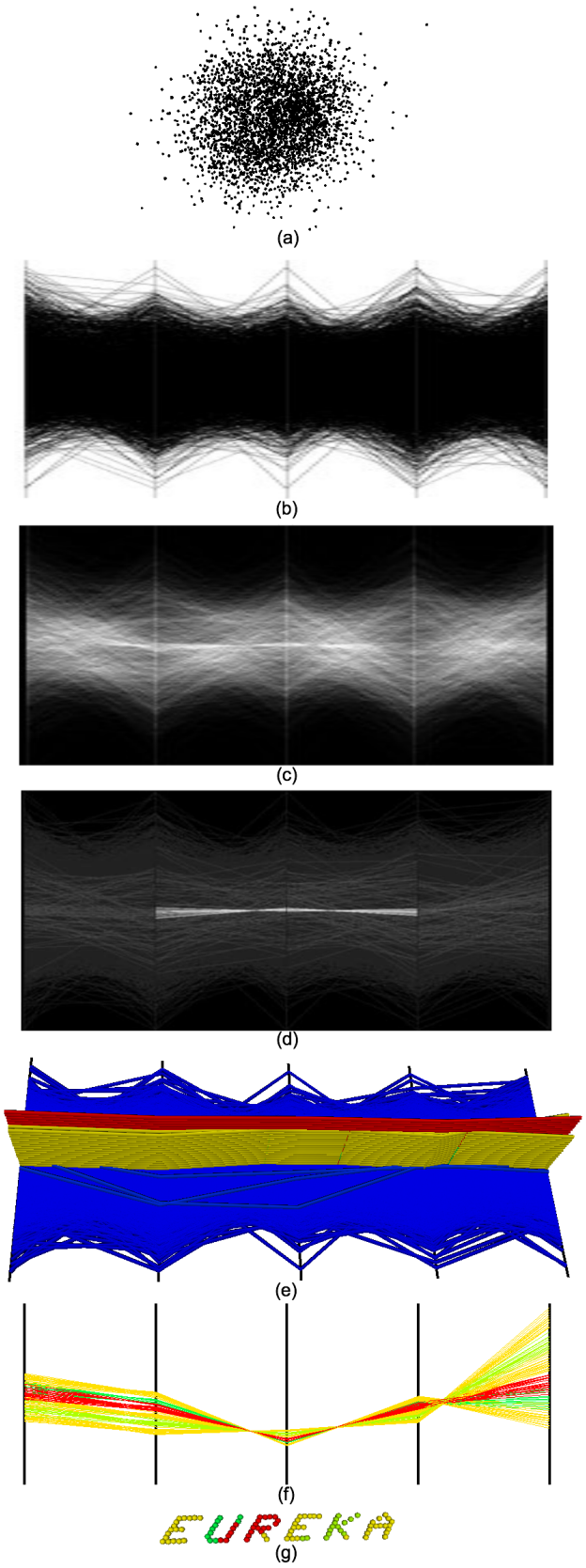


Fig. 10. Visualization of the Pollen data set: (a) Scatterplot; (b) Conventional parallel coordinates; (c) Brightness used to represent density of profiles; (d) Interactive Parallel Coordinates Frequency Plots with a brightness gradient; (e) Parallel coordinate plot with line stacks; (f) The central stacks colored by height; (g) Scatterplot of the subset of points in (f).

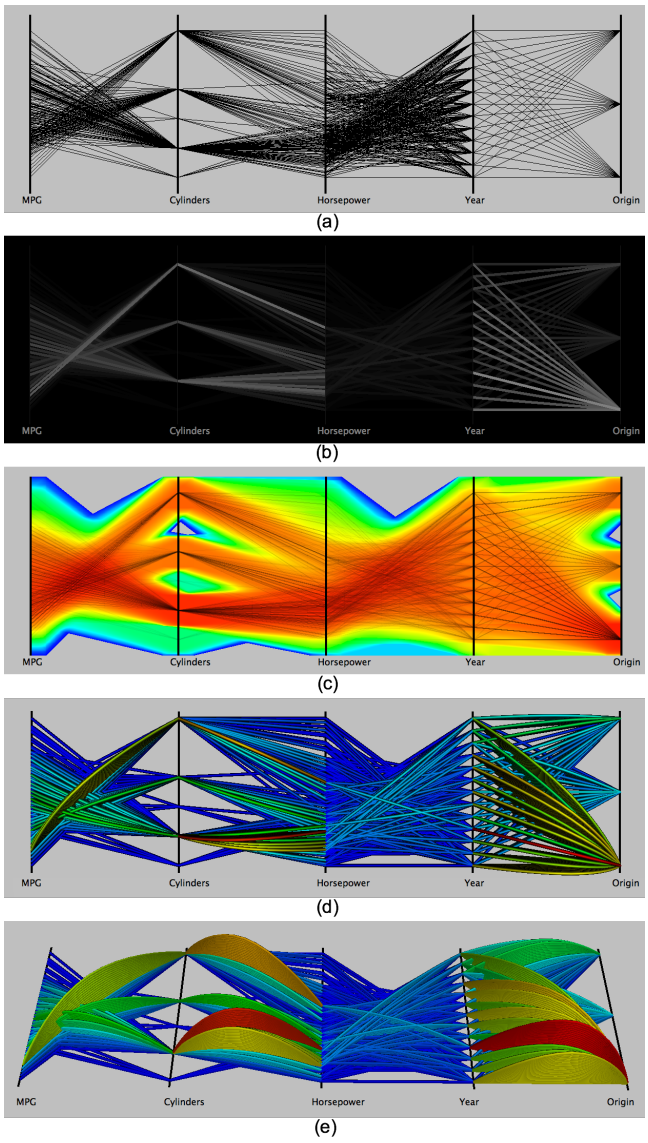


Fig. 11. Several techniques to visualize a portion of the Cars dataset: (a) Interactive Parallel Coordinates Frequency Plots when a scaling factor s = 0.5 ; (c) Density Plots using logarithmic color transfer function; (d) Parallel coordinate plot with line stacks; (e) Another view of Parallel coordinate plot with line stacks; .

Another example is shown in Figure 11. In this example, we use the Cars dataset. The data contain 392 instances and several different kinds of variables: categorical, integer, continuous. Figure 11(a) is an ordinary parallel coordinate plot of the data. Figure 11(b) shows the Interactive Parallel Coordinates Frequency Plot [1] for the same data. This program assigns pixel intensity proportional to the frequency of the adjacent pairs of variables.

Figure 11(c) shows a Density Plot [23] for the same data. This technique smooths the polylines (similar to a kernel density estimate) and generates a color gradient in order to highlight clusters. A nonlinear transfer function is used to accentuate the contrast.

Figure 11(d) and (e) show a stacked parallel coordinate plots in two different views. As in the previous example, stacks are colored based on their heights. Looking at Figure 11(d) (e), one notices that the last pair of variables (Year and Origin) shows more cars coming from the first origin than from the others. In all cars produced by the first origin, one can easily detect the year during which the first origin produced the most (the red stack). This important detail is not evident in the other plots.

4.2 Categorical Stacked Parallel Coordinate Plots

As with ordinary dot plots, stacked parallel coordinate plots work well for both continuous and categorical data; no special adjustments are required. Figure 12 shows how this works. The data are from the Titanic survivors data set [9]. The data involve four categorical attributes for each of the 2201 people on board the Titanic when it struck an iceberg and sank. The attributes are cabin class (first class, second class, third class, crew member), age (adult or child), sex (male or female), and whether or not the person survived.

Figure 12(a) shows a parallel coordinate plot of these data using an algorithm in [12]. Figure 12(b) shows the stacked parallel coordinate plot. The figure shows another variation of our stacking algorithm. We anchor the lines at the vertices in order to reduce occlusions and we permute the values to place large densities at the rear. The stacks are shaded with a linearized hue scale to help distinguish density magnitudes. The stacked parallel coordinate plot clearly reveals the major discrepancies in survival rates. A disproportionate number of males and adult passengers did not survive. Notice that Figure 12(a) fails to reveal this dramatic contrast because there is not sufficient dynamic range in the color palette to represent the magnitude of the count range.

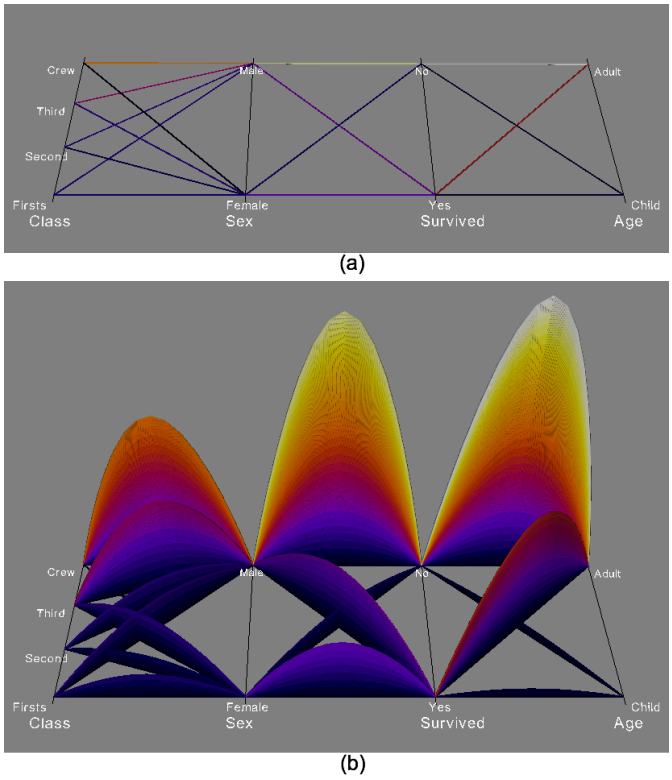


Fig. 12. Visualization of Titanic dataset: (a) Parallel coordinates of counts with profiles colored to represent frequency; (b) Parallel coordinates of counts with line stacks to represent frequency.

An analytic tool should allow us to drill down easily to subcategories. Figure 13 shows how this works with Stacker. We color the stacks by cabin class, so we can examine the conditional distributions of the other variables. Figure 13(a) shows this breakdown; it is evident that the largest category of passengers was the crew (red). Figure 13(b) highlights the crew with the help of single class filtering. Figure 13(c) highlights all surviving passengers. A surprising number of the male crew survived relative to the number of female first or second class passengers who survived. This outcome did not reflect well on the company that owned the Titanic (the White Star Line). In Figure 13(d), we highlight male surviving passengers by clicking on the middle line stack. In each of the selections in the figure, Stacker highlights cases in its data editor, so we can examine them more closely.

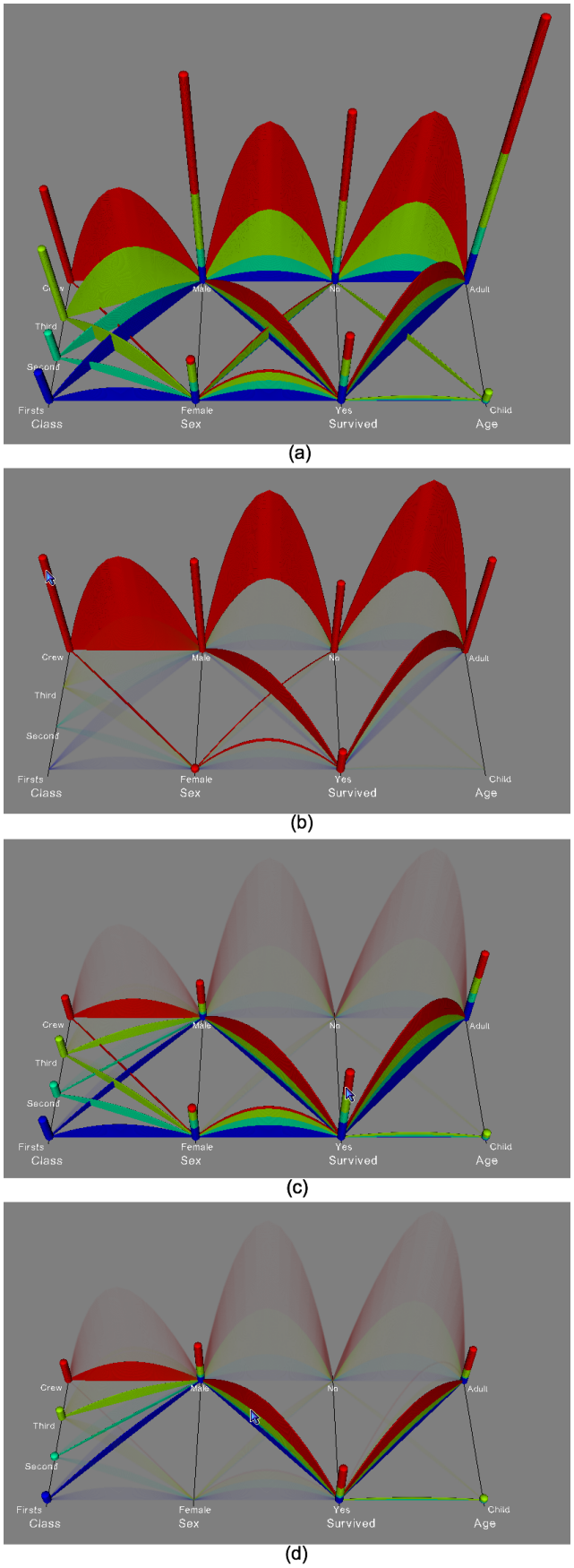


Fig. 13. (a) Parallel coordinate with line stacks of the Titanic dataset; (b) Class filtering: Crew in the boat; (c) Dot stack brushing: Survived passengers; (d) Line stack brushing: Male survived passengers.

4.3 Stacked Regression Lines

Our stacking algorithm can be applied to other geometric elements used in visualizations. Figure 14 shows an example. The upper panel shows a plot of the level curves of the theoretical distribution of sample estimates of a population regression function. Statistical packages compute this distribution in order to derive a confidence interval on a regression line. Statisticians know that the cross-section of this distribution is a normal distribution under the conditional normality assumption for ordinary least squares, but no statistics book or Web site that we could find plots this distribution in such a way that would reveal this conditional distribution. Students would be hard pressed to discern the real shape of the distribution from Figure 14(a).

Accordingly, we decided to simulate the distribution by computing 5,000 multiple regressions. Each regression was computed on 25 randomly generated bivariate normals with a population correlation of .7. Figure 14(b) shows the result of our simulation. We have oriented the 3D plot a little differently in order to show clearly the resulting dot plot comprised of the ends of the cylinders representing the regression lines. The envelope of these lines clearly reveals the underlying normal distribution. Having this figure in an introductory statistics book would give students an understanding of the distribution of linear regression functions with normally distributed error. The two panels of Figure 14 would make an excellent side-by-side plot for illustrating the geometry of the ordinary least squares regression model and help students get beyond formulas.

5 Conclusions

Tallies are one of the oldest forms of visualization. The term tally comes from the practice of cutting notches into bones or sticks. More than 30 years ago, Tukey reformulated tallies as stem-and-leaf plots [30], which are numerical tallies that closely resemble single-variable dot plots. In the computer era, tallies appear to have fallen out of favor (although there is at least one iPhone app for them). Perhaps this is because they are thought to be primitive, to be replaced by "modern" kernel densities, mosaic plots, and other displays. We argue otherwise. Stacking counts deserves a new look and a formal evaluation. We plan to do that now that we have a testbed for generating them. We believe stacking, in appropriate applications, can outperform mosaics, treemaps, parallel coordinates, and other count displays in a perceptual experiment involving judging count magnitudes.

In conclusion, we note the following:

- Stacking in 3D should be regarded as an adjunct rather than a substitute for 2D displays. Just as a 3D plot of a manifold can reveal mathematical structure not discernable in a 2D contour plot, so can stacking reveal a data-based structure not evident in 2D panels or contours, as we have seen in Figure 14.

- Stacking is especially appropriate when color, area, or other aesthetic variables do not have the dynamic range to permit extreme magnitude comparisons. Figure 10 is a cogent example of this point. While brightness perception admits to several orders of magnitude in ideal conditions, printed and computer displays greatly reduce the perceptual range of a stimulus. The third panel of Figure 10 shows a faint white horizontal area, while the fifth panel shows a huge ridge running through the display. We would need custom filtering on the black-and-white image to expose this large a contrast.

- Stacking is not limited to points and lines. Theme River [11] and Baby Name Voyager [33] stack areas to represent time series. Our algorithm implements stacking at the individual case level. We advocate extending our algorithm to other geometric representation elements.

Acknowledgments

This work was supported by NSF/DHS grant DMS-FODAVA-0808860.

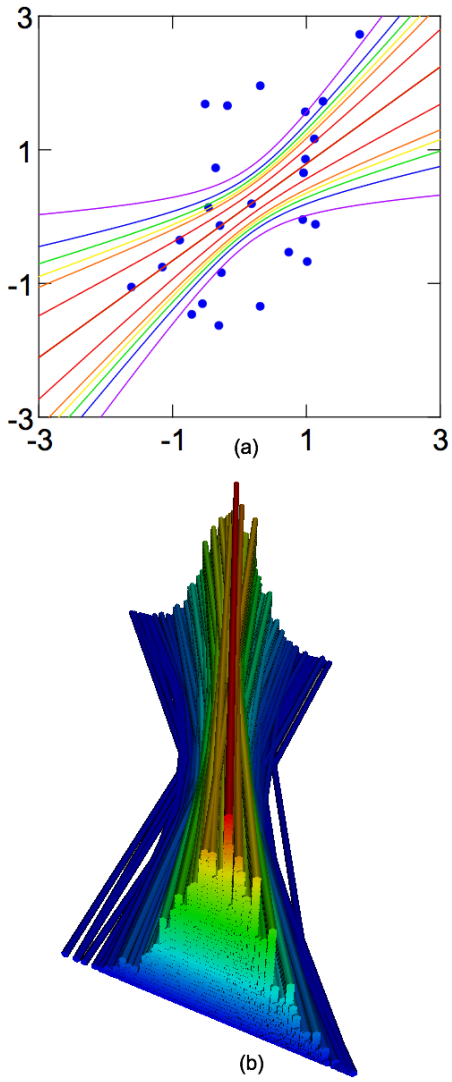


Fig. 14. Regression lines: (a) contours of conditional distribution computed by SYSTAT; (b) 5,000 regression lines stacked and colored by height of stack.

REFERENCES

[1] A. O. Artero, M. C. F. de Oliveira, and H. Levkowitz. Uncovering clusters in crowded parallel coordinates visualizations. In *IEEE Symposium on Information Visualization*, pages 81–88, 2004.

[2] R. A. Becker and W. S. Cleveland. Brushing Scatterplots. *Technometrics*, 29:127–142, 1987.

[3] F. Bendix, R. Kosara, and H. Hauser. Parallel sets: Visual analysis of categorical data. In *IEEE Symposium on Information Visualization*, pages 133–140, 2005.

[4] C. A. Brewer. Guidelines for selecting colors for diverging schemes on maps. *The Cartographic Journal*, 33:79–86, 1996.

[5] D. B. Carr. Looking at large data sets using binned data plots. In A. Buja and P. Tukey, editors, *Computing and Graphics in Statistics*, pages 7–39. Springer-Verlag, New York, 1993.

[6] J. M. Chambers, W. S. Cleveland, B. Kleiner, and P. A. Tukey. *Graphical Methods for Data Analysis*. Chapman and Hall, New York, 1983.

[7] W. S. Cleveland. *The Elements of Graphing Data*. Hobart Press, Summit, NJ, 1985.

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

[9] R. J. M. Dawson. The "unusual episode" data revisited. *Journal of Statistics Education*, 3:3, 1995.

[10] M. T. Gastner and M. E. J. Newman. Diffusion-based method for producing density equalizing maps. In *Proceedings of the National Academy of Sciences*, volume 101, pages 7499–7504, Washington, DC, USA, 2004. National Academy of Sciences.

[11] S. Havre, B. Hetzler, and L. Nowell. Themeriver: Visualizing theme changes over time. In *Proceedings of the 2000 IEEE Symposium on Information Visualization*, pages 115–123, Washington, DC, USA, 2000. IEEE Computer Society.

[12] J. Heinrich and D. Weiskopf. Continuous parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1531–1538, 2009.

[13] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1:69–91, 1985.

[14] D. F. Jerding and J. T. Stasko. The information mural: A technique for displaying and navigating large information spaces. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):257–271, /1998.

[15] W. S. Jevons. *On the condition of the gold coinage of the United Kingdom, with reference to the question of international currency*. Macmillan, London, 1884.

[16] J. Johansson, M. Cooper, and M. Jern. 3-dimensional display for clustered multi-relational parallel coordinates. In *In Proceedings 9th IEEE International Conference on Information Visualization (2005*, page 188, 2005.

[17] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *IEEE Symposium on Information Visualization*, pages 125–132, 2005.

[18] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure in visualizations of dense 2d and 3d parallel coordinates. *Information Visualization*, 5(2):125–136, 2006.

[19] A. Johnson. Utilities dataset. Personal communication, 2010.

[20] B. Johnson and B. Shneiderman. Treemaps: A space-filling approach to the visualization of hierarchical information structures. In *Proceedings of the IEEE Information Visualization '91*, pages 275–282, 1991.

[21] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 8(1):1–8, 2002.

[22] A. F. Krieg, J. R. Beck, and M. Bongiovanni. The dot plot: A starting point for evaluating test performance. *Journal of the American Medical Association*, 260:3309–3312, 1978.

[23] K. T. Mcdonnell and K. Mueller. Illustrative parallel coordinates. *Computer Graphics Forum*, 2008.

[24] K. Pearson. Maps and chartograms. Gresham Lecture, 1891.

[25] W. Playfair. *The commercial and political atlas and statistical breviary. Original edition (1786) edited and republished by H. Wainer and I. Spence*. Cambridge University Press, Cambridge, 1786.

[26] J. F. Rodrigues, A. J. M. Traina, and C. Traina. Frequency plot and relevance plot to enhance visual data exploration. In *Computer Graphics and Image Processing*, pages 117–124, 2003.

[27] D. W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. John Wiley & Sons, New York, 1992.

[28] B. Silverman. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, New York, 1986.

[29] M. Streit, R. C. Ecker, K. sterreicher, G. E. Steiner, H. Bischof, C. Bangert, T. Kopp, and R. Rogojanu. 3D parallel coordinate systems-a new data visualization method in the context of microscopy-based multicolor tissue cytometry. *Cytometry Part A*, 69A(7):601–611, 2006.

[30] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley Publishing Company, Reading, MA, 1977.

[31] B. A. Turlach. Bandwidth selection in kernel density estimation: A review. In *CORE and Institut de Statistique*, 1993.

[32] C. Ware. *Visual thinking for design*. Morgan Kaufman, Burlington, MA, 2008.

[33] M. Wattenberg. Baby names, visualization, and social data analysis. In *Proceedings of the 2005 IEEE Symposium on Information Visualization*, page 1, Washington, DC, USA, 2005. IEEE Computer Society.

[34] E. J. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85:664–675, 1985.

[35] E. J. Wegman and Q. Luo. High dimensional clustering using parallel coordinates and the grand tour. *Computing Science and Statistics*, 28:361–368, 1997.

[36] A. Wilhelm. User interaction at various levels of data displays. *Computational Statistics and Data Analysis*, 43(4):471–494, 2003.

[37] L. Wilkinson. Dot plots. *The American Statistician*, 53:276–281, 1999.

[38] L. Wilkinson. *The Grammar of Graphics*. Springer-Verlag, New York, 2nd edition, 2005.