First publ. in: Information Visualization ; 9 (2010), 4. - pp. 301-311

http://dx.doi.org/10.1057/ivs.2009.34

Generalized scatter plots

Daniel A. Keim[a]
Ming C. Hao[b]
Umeshwar Dayal[b]
Halldor Janetzko[a] and
Peter Bak[a,*]

[a]University of Konstanz, Universitaetsstr. 10, Konstanz, Germany.
[b]Hewlett Packard Research Labs, 1501 Page Mill Road, Palo Alto, CA94304, USA.

*Corresponding author.

Abstract Scatter Plots are one of the most powerful and most widely used techniques for visual data exploration. A well-known problem is that scatter plots often have a high degree of overlap, which may occlude a significant portion of the data values shown. In this paper, we propose the generalized scatter plot technique, which allows an overlap-free representation of large data sets to fit entirely into the display. The basic idea is to allow the analyst to optimize the degree of overlap and distortion to generate the best-possible view. To allow an effective usage, we provide the capability to zoom smoothly between the traditional and our generalized scatter plots. We identify an optimization function that takes overlap and distortion of the visualization into acccount. We evaluate the generalized scatter plots according to this optimization function, and show that there usually exists an optimal compromise between overlap and distortion. Our generalized scatter plots have been applied successfully to a number of real-world IT services applications, such as server performance monitoring, telephone service usage analysis and financial data, demonstrating the benefits of the generalized scatter plots over traditional ones.

Keywords: scatter plot; overlapping; distortion; interpolation; smoothing; interactions

Introduction

Motivation

Large amounts of multi-dimensional data occur in many important application domains such as telephone service usage analysis, sales and server performance monitoring. Analysts want to know how much one attribute of the data set is affected by another. In 1984, William Cleveland *et al*[1] wrote:

> The scatter plot is one of our most powerful tools for data analysis.

It is still true that scatter plots (sometimes they are also called x–y diagrams) are one of the most common ways to visualize multidimensional data. Using scatter plots, we can identify the relationship between two attributes, clusters of points and outliers. However, todays scatter plots have a high degree of overlap, which obscures the true density of data values. William Cleveland *et al*[1] already noted:

> Still, we can add graphical information to scatter plots to make them considerably more powerful.

In his article, Cleveland introduced different types of enhancements including a combination with iconic techniques and a superposition of smoothing methods for enhancing the x–y axes dependency and scale-ratio. Clevelands ideas are great enhancements of scatter plots, but they do not solve the overlap problem of scatter plots showing large

data sets. In exploring large data sets, the high degree of overlap in scatter plots is one of the most severe drawbacks, which often causes correlations to be hidden or at least difficult to observe.

In this article, we address the overlap issue and propose a generalization of scatter plots where the analyst can control the degree of overlap and distortion allowing the analyst to generate many different views for revealing patterns and relations from the data. In the Section 'Applications', we will illustrate the applicability of the proposed technique. The potential of the technique in revealing previously barely visible information will be demonstrated by showing the power of combining distortion and pixel placement on real-world data set. Our examples show that the strength of the proposed methods lies first in the combination of two techniques, and second in the ability of the users to interactively guide the progress between the original and the generalized representation of the data.

Related work

To create a representation of an entire high-density scatter plot without overlap, we need a visualization technique that places a large volume of data in the limited size of the display screen. Early successful high information density displays were for example pioneered by Eicks SeeSoft system.[2] Eick allows users to analyze up to 50 000 lines of code simultaneously by mapping each line of code into a thin row for finding interesting patterns. To explore the high-density display, Eick also provided Data Visualization Slides[3] to filter the information displayed on the screen. Users can use the slider to prune the visual clutter and explore the display from several perspectives.

In the meantime, many interesting approaches for large high-density displays have been developed. In Jerding and Staskos[4] Information Mural developed a technique for displaying and navigating large information spaces without filtering. Information Murals use gray-scale shading and color along with anti-aliasing techniques to create a miniature version of the entire data set. Jerding and Stasko[4] are able to plot over 52 000 sunspot values in a small display window. Users can navigate the detailed subset of the data from the miniature overview window of the entire data set.

Later, Fish-Eye views and Degree of Interest techniques try to integrate overview and detail windows. Spence[5] and Furnas[6] for example, enable users to focus on something interesting in a large graph. Bertini and Santucci[7] use sampling techniques to localize a particular region of the display. Bertini and Santucci[7] allow the user to examine small data items in detail, avoiding a loss of information in low-density areas while reducing overplotting in high-density areas. Ellis[8] uses auto-sampling to reduce the overplotting in parallel coordinates and scatter plots. Similarly, Fua *et al*[9] uses hierarchical clustering techniques[10] to reduce clutter in several visualizations in the system.

Distortion plays an important role in avoiding overlap in scatter plots. Buering[11] provides two user interaction techniques on a small screen: a geometric-semantic zoom that smooths transitions between overview and detail; and a fish-eye distortion that displays the focus and context regions of the scatter plot in a single view. Keim[12] uses pixel-based distortion to efficiently generate cartograms for showing geography-related statistical information.

Alternatively to distortions, many researchers suggested alpha-blending, which uses the alpha-transparency of the color system to represent data points. As a result, highly overplotted areas have high opacity and sparse areas have higher transparency.[13] In the book by Antony Unwin *et al*,[13] a number of interesting visualization techniques were introduced regarding scatter plots, such as drawing overlapping points with slightly bigger sizes and reducing the x and y axes by certain factors. JMP 8 Software[14] generates scatter plots with non-parametric density contours and marginal distributions to show where the data is most dense. Each contour line in the curved shape encloses 5 per cent of the data. Carr[15] uses a hexagonal-shaped symbol whose size increases monotonically as the number of observations in the associated bin increases, and HexBin scatter plots[15] determine the brightness value of each HexBin cell depending on the number of data points in the cell. All three techniques, Unwins distortion, Carrs binning and the HexBin visualization techniques, are close to the method presented in this article. However, using these techniques analysts are not able to see and access all data points, especially if the third variable mapped to color is of high importance. In order to overcome this problem, scientists suggested to use jitter or minimal random noise, which aim at avoiding overplotting, but result in random distribution of data points, and the heterogeneous patterns of colored data points might bias users' perception.[15]

Also different interaction techniques are suggested to solve the overplotting problem. Different zooming techniques are commonly used. With these methods, users are able to request more details by increasing the magnification of representation. The question of determining the optimal level of the resolution still remains a challenge.[13] Additionally, scaling of the axis has been applied in many fields to endow higher density areas with a larger space and sparse areas with a lower space. Square root and Logarithmic scaling to x- and/or y-axis is certainly very popular. This method, however, is only applicable to the visualization when it represents an explicit feature of the data. The applicability of such scaling techniques is therefore very limited. Figure 1 illustrates some examples of the methods mentioned. The data used for the representations is the Telephone Conference Data (N = 37 788), described in the Application section in more detail. The data set shows the duration of the conference call on the x-axis, the charges for the call on the y-axis and the number of participants are mapped to color. First the (a) original data set with a linear scaling, (b) logarithmic scaling on the x- and

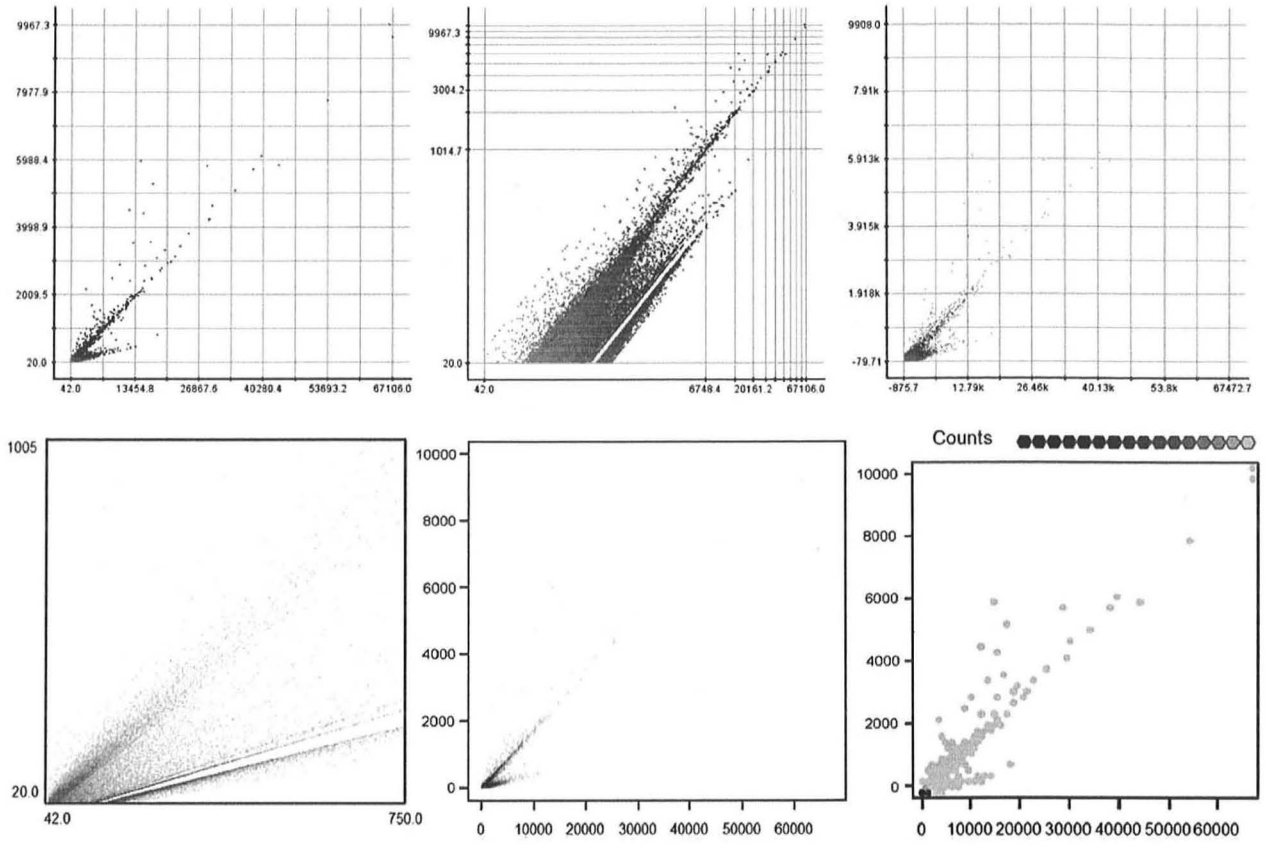

Figure 1: Traditional techniques aiming to overcome the overplotting problem applied on a Telephone Service Usage data set. X-axis represents the duration, y-axis represents the charges for the call and number of participants are mapped to color.

y-axis, (c) jitter enhanced, (d) zooming, (e) alpha-blending and finally the (f) HexBin approach are shown in Figure 1.

Generalized scatter plots

Common to the related approaches mentioned above is that they provide various ways to reduce clutter (for example, color-shading, sampling, clustering and distortion) to visualize a large data set. In this article, we propose generalized scatter plots with a variable degree of distortion and a variable degree of overlap. The traditional scatter plots are just a special case of our generalized scatter plots with no distortion and the data induced overlap. As in normal scatter plots, each data point is presented as one color pixel or small color icon, and users can move the pointer to see the content. In the distorted and/or overlap-optimized generalized scatter plots, data values are placed as close as possible to their original positions. Our method allows the user to dynamically move the slider to optimize the degree of overlap and distortion to generate the best possible view. It allows a seamless path from traditional scatter plots and our generalized scatter plots.

In addition, we incorporate the intelligent visual analytics queries[16] to perform root-cause visual analyzes. The real application examples in the areas of telephone service usage analysis, computer server performance monitoring and financial data analysis demonstrate the benefits of our generalized scatter plots over the traditional ones. The remainder of this article is organized as follows: The section Generalized scatter factors, further describes the method by general optimization goals and layout algorithms of the generalized scatter plots. The efficiency of our new approach and applications are described in the sections 'Applications' and 'Computation of optimization goals'.

Method

The overlap optimized scatter plot allows a variable degree of distortion and a variable degree of overlap. The distortion is based on a linear distortion in x and y direction similar to the HistoScale approach,[17] which ensures an equal distribution of the data in x and y dimension. The distortion grants more space to areas with high density and less space to areas with low density while retaining

neighborhood relationships of the data points. More space in high density areas allows us to decrease the necessity of pixel displacements, which improves the effectiveness and efficiency of the generalized scatter plots.

In contrast to the simple linear scaling of HistoScale, we use a variable distortion level, which can be interactively adjusted by the user. For a given distortion level (between 0 per cent and 100 per cent), the overlap optimization can also be interactively adjusted between 0 per cent and 100 per cent, where 0 per cent corresponds to the data-induced overlap level and 100 per cent corresponds to complete overlap avoidance if possible for in the given display space. The overlap optimization is done by a circular arrangement around the original location regarding to a given ordering of the elements. The ordering usually corresponds to the coloring attribute with a default ordering starting colors that occur least frequently. With this arrangement, we generate a natural-looking visualization without artifacts. The ordering of elements prevents randomly arranged points that would not benefit the user. The details of the implementation and user involvement in creating overlap-optimized scatter plots is discussed in the following sections.

Implementation

Because our generalized scatter plot system is designed as an interactive tool, our algorithm has to be as efficient as possible. The overlap-optimized pixel placement algorithm is the most time-consuming part of our generalized scatter plot and therefore has been carefully implemented.

```
Algorithm: doPixelArrangement(OrderedList DataObjects)
int [ ] [ ] overlapCnt := new int[width][height];
for each o of DataObjects do
  Point p := o.getPixelPos();
  if (overlapCnt[p.x][p.y] < maxOverlap) then
    o.setPaintPos(p);
    overlapCnt[p.x][p.y] + +;
  else
    rearrangeDataObject(o, p, overlapCnt);
  end
end
```

As depicted in 'doPixelArrangement', our algorithm displaces the points in order of their priority (for example, the value of the point) to avoid random patterns in the resulting visualization. Because we have to remember how many data objects are already located at a specific pixel location, we need a two-dimensional integer array representing each pixel of the display area. For each data point, the program has to look up the number of data objects already placed at the preferred position of the data object and compare this to the maximum allowable number of overlapping points, which depends on the interactively chosen overlap level (slider). If the current data object can be placed at its preferred location, we have to store this information in the two-dimensional integer arrays. Otherwise, we have to look for the next free pixel position in order to place the current data object as illustrated in 'rearrangeDataObject'.

The procedure rearrangeDataObject does the real pixel placement: In order to have a fast algorithm for each pixel we store the radius, which was used for the last displacement (The initial value is 1). We can calculate the pixels of the circle around point p with this radius. The calculation of the pixels is done by using a modified version of the Bresenham Midpoint algorithm.[18] The Bresenham algorithm was modified in such way that it calculates the pixels of a circle with a line width of two. The modification was necessary, as with the standard Bresenham algorithm not every pixel is touched when we increase the radius by one, which means that a significant number of pixels are not used, thereby creating artifacts in the resulting visualization.

The calcCirclePoints method returns the pixels of the circle ordered by their distance from the original pixel position. As we have a choice of candidate pixels we have to check each of them until we can either place the data object or there arent any pixels left on the circle with the current radius. In the second case, we have to increase the radius and calculate the new pixel positions and then go on as described above. When we have found a suitable pixel position we can store the radius we had to use to accelerate the next displacement operations.

```
Algorithm: rearrangeDataObject(o, p, overlapCount)
int r := getLastUsedRadius(p);
Point[ ] circlePoints := calcCirclePoints(p, r);
while new place not found do
  if any circlePoints left then
    Point p := next circlePoint;
    if (overlapCount[p.x][p.y] < maxOverlap) then
      o.setPaintPos(p);
      overlapCount[p.x][p.y] + +;
    else
      r + +;
      circlePoints := calcCirclePoints(p, r);
    end

  end
updateLastUsedRadius(p, r);
```

Smooth interpolation

As our generalized scatter plot provides plots anywhere in between the traditional scatter plot and the overlap-optimized visualization, we had to implement a smooth interpolation between these extremes. The interpolations of distortion and overlap-optimized visualization are calculated differently and are therefore independent of each other. For the interpolation between the distorted and non-distorted positions, we use a weighted average. This weight can be adjusted interactively using a slider and it directly influences the linear interpolation. The system automatically determines the distortion level that

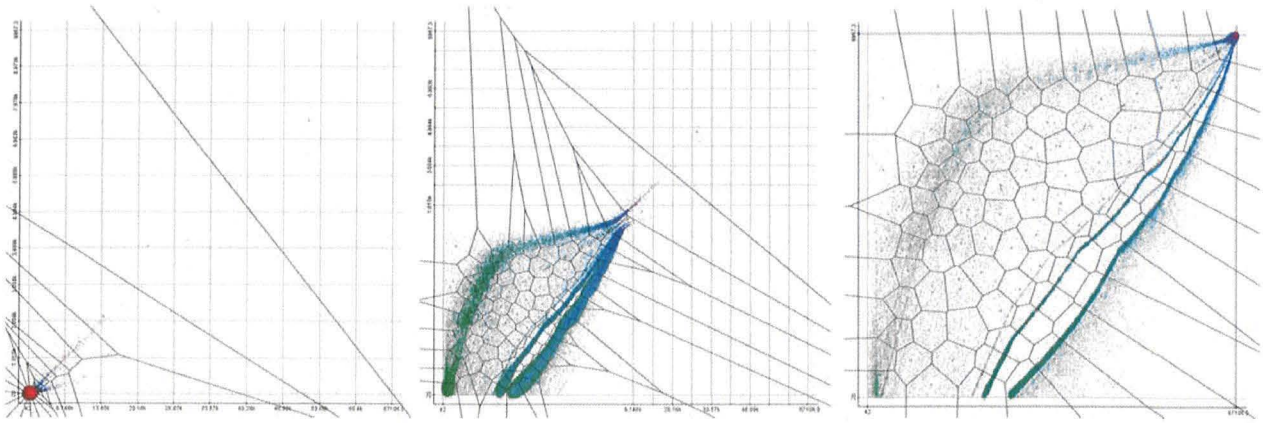

Figure 2: Intelligent Data Analysis using k-Means (=100) and Veronoi cells for their representation.

best fits to the data (see the section 'Computation of optimization goals' for details). Using the slider, the user can easily adapt the distortion level according to users preference.

In some cases, it is not useful to draw a scatter plot without any overlapping points. For very large data sets or highly skewed data distributions, for example, the properties of the data are difficult to see if too many pixels are displaced by the overlap-optimized pixel placement algorithm. For this reason we provide a slider to enable the user to control the degree of overlap. For a smooth interpolation between the data-induced maximum overlap and the overlap-free scatter plot, we determine the maximum number of overlapping points and use $x \cdot maxOverlapPoints$ (where x is in [0, 1] and is set by the user with a slider) as the maximum overlap degree in the generated visualization. A short video showing the transition from an original scatter plot to an overlap-optimized scatter plot, based on the Telephone Service Usage data set, is available under: www.informatik.uni-konstanz.de/fileadmin/dataMining/GSP_video.avi.

Intelligent visual queries

Interactivity is an important aspect of our visual data exploration technique. To make large volumes of multidimensional data sets easy to explore and interpret, in addition to the layered drill-down and zoom, we allow intelligent visual analytics queries to be performed by the user. Analysts can rubber band an interesting area and invoke data mining algorithms to determine attributes that are closely related to the selected attribute. A number of data mining methods including correlation analysis, similarity functions and cluster analysis are used to analyze relations between attributes. The mining results are again presented as generalized scatter plots, allowing the user to seamlessly continue the exploration.

We choose the clustering task to demonstrate the added value of the overlap optimization. Clustering is often used to represent significant groups in the representation and to extract patterns of interest by partitioning the data set. The system currently includes the k-Means clustering algorithm and the creation of Voronoi tessellations, which are modules that can be easily extended by techniques most suitable for the underlying data set. The k-Means algorithm partitions the data into k clusters, in which each point belongs to the cluster with the smallest distance. Resulting in k cluster centers that correlate with the distribution of the data. The Voronoi tessellation is applied to the resulting cluster centers, in order to create a visual feedback to which clusters the data points belong. Red marks are used to show cluster centers, and the borders of the Voronoi cells are drawn in black.

Figure 2 shows the results of an analysis using the described methods for the original, medium-distorted and maximum-distorted scatter plot. In all cases the degree of overlap is set to 50 per cent. From the representations, it is clearly evident that overlap optimization improves the clustering results. The Original data set (left) is partitioned inefficiently and results in a crowd of cluster centers with no ability to segment the data. The medium distortion (middle) reveals already more details and more separable segments appear. Finally, the maximum distortion (right) reveals fully the patterns previously hidden in the data. The cluster centers follow the high-density lines and enhance the visibility of these patterns, and the Voronoi cells clearly segment the data points along these patterns.

Optimization goals

One optimization goal is that the displacement of pixels with respect to their original position should be minimal, which is important in order to understand the generated scatter plots. For a given data set of n points p1, pn, let O(pi) denote the original location and N(pi) denote the calculated position in the generated scatter plot, and d(O,N) is a distance function in the scatter plot measuring

the Euclidean distance of O and N. The displacement error is calculated as follows:

$$e_{disp} = \sum_{i=1}^{n} \frac{d(O(p_i), N(p_i))}{n}$$

The displacement error measures the amount of positional changes of all data points between the original scatter plot and the generalized scatter plot. The second optimization goal is that the overlap of points should be as minimal as possible. The overlap of points can be measured by the following function:

$$e_{overlap} = \frac{|\{p_i | \exists j : N(p_i) = N(p_j) \wedge i \neq j\}|}{n}$$

Note that there is a trade-off between the two functions: If we increase the distortion we usually get a lower overlap error but always have a higher displacement error. To calculate a combined optimization function we suggest a weighted sum of the error functions:

$$c * e_{dist} + (1 - c) * e_{overlap} \rightarrow \text{MIN}$$

with c being proportionality constant. Increasing c would allow a lower level of distortion and a higher degree of overlap. Correspondingly, decreasing c would show inverse results. For a balanced weight of displacement and overlap errors, c should be set to 0.5. In the section 'Computation of optimization goals', we evaluate the two-error functions and show that we can use the optimization function to determine the optimal distortion and overlap values.

Applications

Real-world data sets can best show the contribution of the proposed overlap-optimized scatter plot technique. We selected three domains, in which scatter plots are commonly used. First, we show the already introduced Telephone Conference Call data set with 37 788 entries. Second, we show Server Performance Evaluation with 69 056 measurements in five performance measures. Third, we show a Financial data set containing bond market prices of 1 month in two consecutive years (March 2004 and 2005) having over 8000 entries. The insights gained through our technique are the result of consulting experts of the particular field in informal interviews.

A telephone service usage analysis

Telephone service usage analyses include the following tasks:

- exploring the distribution of the call amounts,
- determining the call duration time and the most common charges, and
- investigating the correlation of the conference call charge with the length of the call and with the number of participants.

Overlap-optimized scatter plots can help revealing the answers to these questions. Overlap-optimized scatter plots have the advantage that they are more similar to traditional scatter plots – in the case of no distortion and data-induced full overlap they are identical to traditional scatter plots. There is no need to an artificial binning, which also helps to retain a more traditional view of the data and especially displays neighborhood relationships better. In this example, we also show different versions of our Generalized Scatter Plots with different distortion and overlap levels. Upper left in Figure 3 displays a normal scatter plot without distortion and maximum data-induced overlap. From left to right, we reduce the overlap first to 50 per cent, then to 100 per cent. From up to down, we increase the distortion level first to 50 per cent, then to 100 per cent. Highest degree of distortion with no overlap is presented in the lower right corner.

Note that overlap and distortion can be controlled independently, which means that arbitrary other configurations can easily be interactively explored. In comparing the different variants, it is clear that the normal scatter plot (upper left in Figure 3) reveals only a very limited amount of information. Two lines are barely visible and therefore it is difficult to see how many calls are made. Also, the very high duration and charge calls are very sparse and are therefore barely visible. Also, the dependency between participants and charge/duration is not visible due to the high level of overplotting.

While increasing the distortion level (from up to down) and decrease the overplotting (from left to right), the highly clustered data is partitioned and more details about the data become visible. At least two curves are visible for maximum overlap and medium distortion level (central left), which split into at least four separate curves by maximum overlap and maximum distortion level (lower left). Finally, minimum overlap and maximum distortion (lower right corner) clearly shows interesting details that are neither visible in the traditional scatter plot nor in the binned scatter plot. In this final representation, up to nine different curves can be discerned, each corresponding to a particular rate. In addition, analysts are able to learn additional facts form the data, demonstrating the additional value of our generalized scatter plots. The following correlations between the charges, duration and the number of participants can be observed:

1. The left curve illustrates that the most expensive calls have high volumes (many data points) and correlate with the time and number of participants. However, there is a wide distribution in charges. Interestingly, the most expensive calls are the national calls.

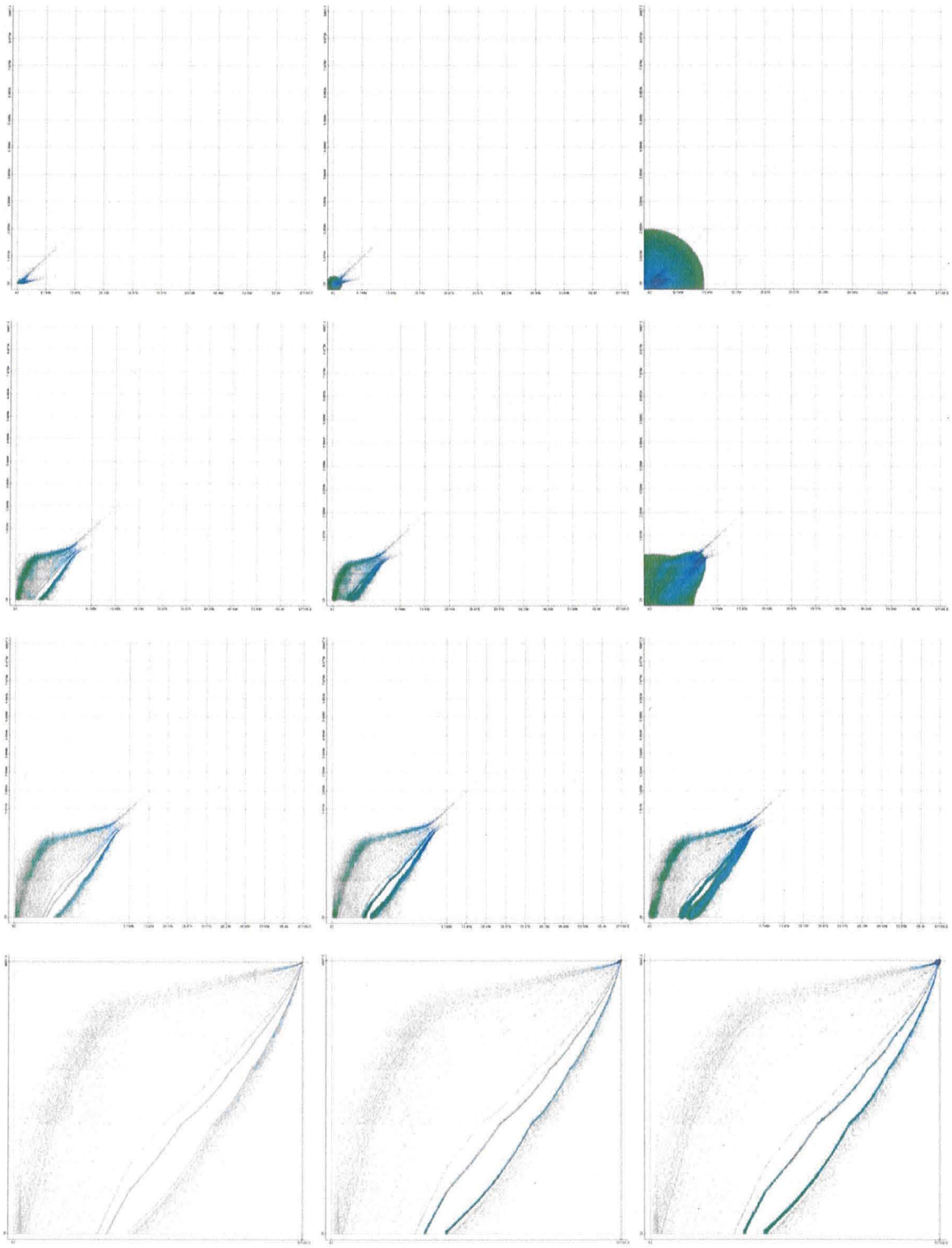

Figure 3: Telephone Service Scatter Plot without Overlap. Call duration on the *x*-axis, charges on the *y*-axis, the color is mapped to number of participants. Original data represented in upper left corner, overlap is stepwise reduced (left-right) and distortion is stepwise increased (up-down).

2. Then, there is a middle curve that is significantly less expensive but also shows a clear correlation between charge and seconds. This curve is due to a special rate to Canada, which can only be used for a small number of participants (only green points).
3. The right section contains the international calls. There are three blue curves representing three different service providers (AT&T, Sprint and ConCall). The rightmost curve has the highest number of calls (AT&T, the thickest curve), which with a high degree of overlap could not be shown in Figure 3 (left upper plots).
4. The thickness of the curves reveals the number of national and internal calls. From the comparison of the thickness of the curves we learn that the international calls have a clear charge structure for each provider (solid lines) whereas the charges of national calls are more variable and depend on other parameters not shown in the visualization (for example time of the day).

Server performance generalized scatter plot matrix

As a second application example, we use server performance data with measurements such as queue length, dispatched jobs, disc, swap and message queue. To answer the question Which system resource causes the server to be busy? The analyst can apply the generalized scatter plot visualization technique to find correlations and closely related attributes. Figure 4 shows a generalized scatter plots matrix of the data with a medium distortion level and no overlap. The lower left half of the scatter plot matrix shows the traditional scatter plots and the upper right half shows our generalized scatter plots. The color of the pixels shows the overall server performance measurement server busy per cent. The following facts can be observed:

- There are many jobs with a low value for queue length, most of which also have a small value for server busy per cent (green color).
- Server busy per cent has a high correlation with all attributes except message queue and swap (all top right corners of dispatched jobs and disk usage have blue and burgundy colors).
- Server busy per cent is highly correlated to the number of dispatched jobs and the disk usage. This fact is indicated by the burgundy color in the top right corner.

One interesting effect occurs in queue length and disc: Most of the jobs have low queue length (green), except for two exceptional clusters (blue) that have high disk usage. The service manager can perform intelligent visual queries to find the root cause of the problems and take preventive actions. The advantage of using generalized scatter plots is to allow analysts to visually compare the correlations across many different attributes in one single display without losing any details due to overplotting even for very large data sets.

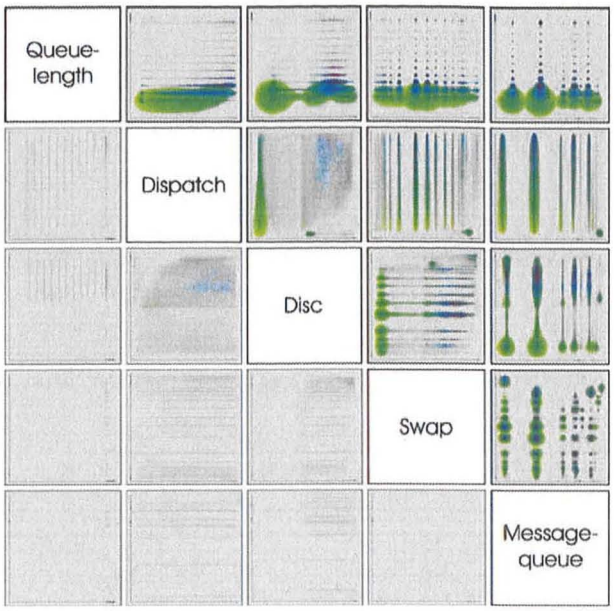

Figure 4: Overlap-Optimized Scatter Plots of Server Performance Data with medium Distortion and no Overlap (color: Server Busy in per cent). Below the diagonal are the original scatter plots and above the diagonal are the overlap-optimized scatter plots.

Financial data analysis

As a further application example, we have chosen to apply the described methods on financial data, which is traditionally interested in large number of observations in a high resolution. The current example aims to compare the prices of a large number of bonds in a yearly development. For this purpose, the prices of March 2004 (x-axis) and prices of March 2005 (y-axis) are compared with the additional information about the number of month a particular stock was on the market (represented by color), as shown in Figure 5. Darker colors indicate that the bond is a longer period on the market and brighter colors that it is fairly new on the market.

The combination of the distortion and pixel placement can reveal the information hidden in the representation. The expected linear correlation between the prices at the two different points in time is seen through the distortions technique. Pixel placement makes the relation between bond prices and time on the market visible. The first representation (upper left scatter plot) shows the traditional scatter plot. High distortion, using HistoScale, is applied on the right upper representation. Overlap-free techniques are shown in the two lower representations, without distortion (left) and with distortion (right). As a result of the overlap free and high distortion technique,

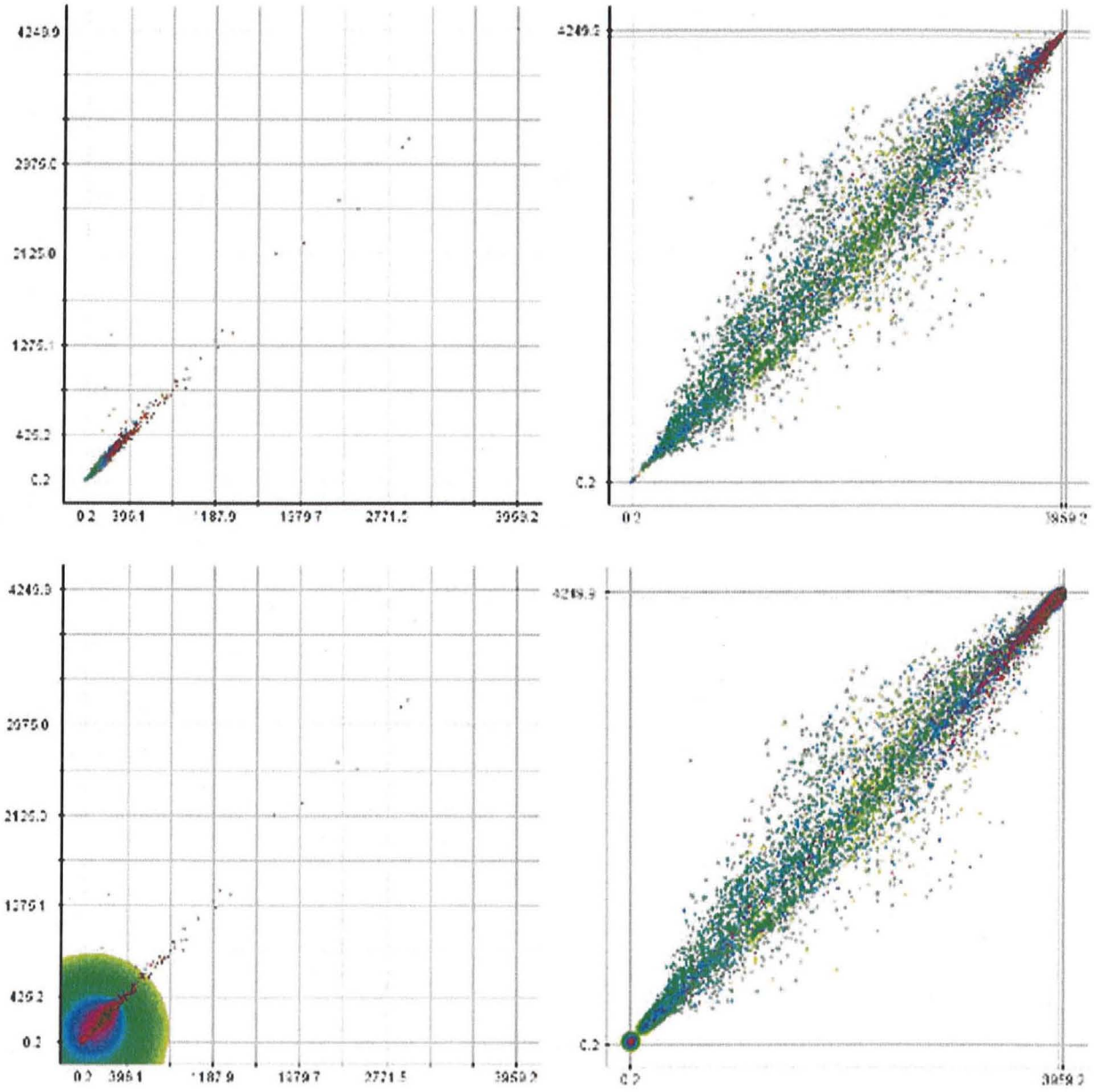

Figure 5: Financial data set showing bond market prices of two consecutive years. Color represents the number of months on the market. The original data (left upper) reveals its information only when distortion technique (right upper) and overlap free techniques without (left lower) and with distortion (right lower) are applied.

analysts are able to explore, interpret and make previously unknown information visible:

- A large number of bonds, close to the diagonal, indicate consolidate and secure development.
- Some of the bonds are more sensitive to changes then others, and are therefore less risk aversive (deviation from regression line). Above the diagonal are bonds with an increase in price; below are those with a decrease in price.
- The time the bonds are on the market plays a central role in their price, though not in their risk averseness.
- Many of the high-risk bonds are ones that are only shortly on the market. Many of these were positive, but also an equal amount negative during the investigated time period.

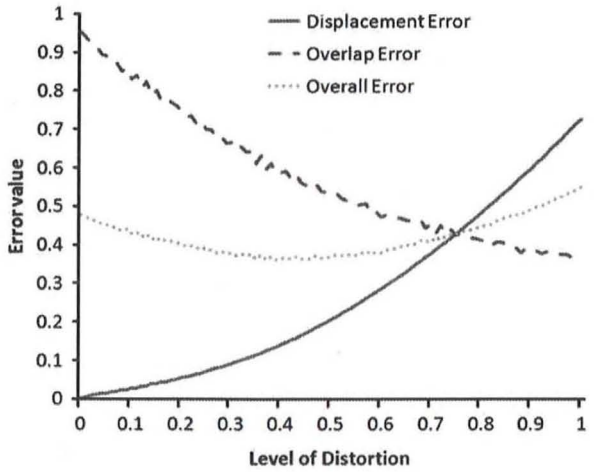

Figure 6: Overlap, displacement and overall-error as a function of distortion level.

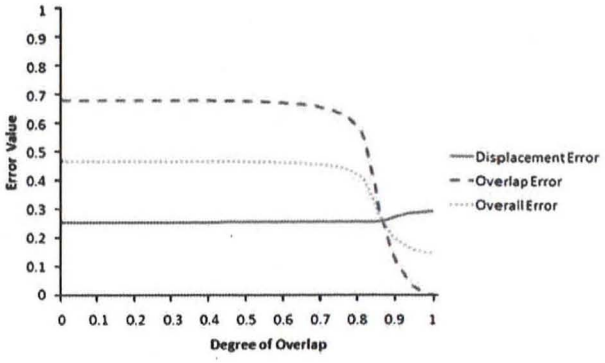

Figure 7: Overlap, displacement and overall-error as a function or degree of overlap.

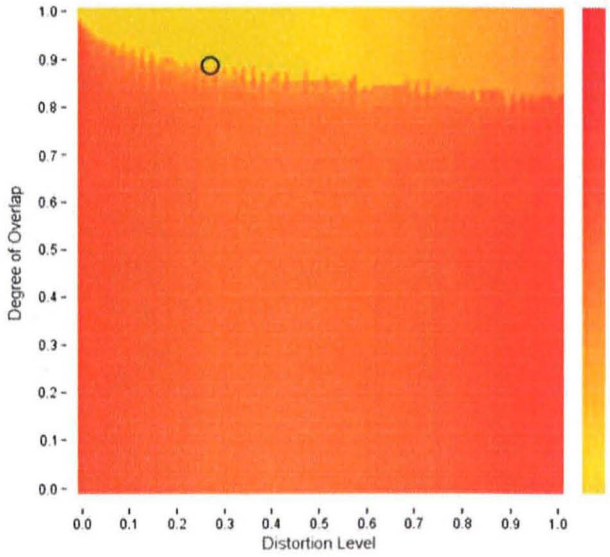

Figure 8: Overall-error as a function of distortion level and degree of overlap. Yellow color represents lower and red color higher error values. The optimal combination of the parameters is circled (d=0.28, o=0.89) for balanced weight ($c = 0.5$). This optimal value (lowest value colored yellow) might change as a function of the weighting (c).

Computation of optimization goals

In this section, we discuss the trade-off between displacement and overlap error that have been introduced in the subsection 'Implementation'. On the one hand, if we increase the distortion using the distortion slider, consequently the displacement error measuring the displacement of points with respect to their non-distorted location increases. On the other hand, the overlap error measuring the percentage of overlap points decreases with increasing distortion. We measured both error functions while increasing the distortion level first, as shown in Figure 6. This evaluation was carried out using the telephone conference call data set discussed in the subsection 'A telephone service usage analysis'. As the displacement error function is increasing while the overlap error function is decreasing, it is obvious that the combined (overall) optimization function has a well-defined minimum. Currently, the minimum occurs at a 73 per cent distortion level (for example, Figure 3 (lower center) for a corresponding visualization).

Similarly, the variations of the degree of overlap have an impact on the computed error functions. As shown in Figure 7, the overlap error function is highly sensitive to higher degrees of overlap. The impact of degrees of overlap on the displacement error is marginal. These representations and computations are based on a balanced weight for displacement and overlap errors ($c = 0.5$). The optimal degree of overlap is 89 per cent for the current data set.

More important is the case, in which we vary both the distortion level and the overlap at the same time. Figure 8 shows the overall error value as a combination of distortion levels and degrees of overlap. For this particular data set, the weight (c value) was set to 0.5, because experiments showed only marginal differences for c values between 0.25 and 0.75. However, in other data sets, an increase of the c value could result in a significant decrease of the optimal distortion level and an increase optimal degree of overlap (inversely for a decreasing c). Distortion levels are mapped to the x-axis and degrees of overlap are mapped to the y-axis. The overall error values are mapped to the different colors of a heat map reaching from yellow to red. More red areas refer to higher error values and yellow areas to lower error values, in order to achieve highest highlighting of areas of interest. The results show that there is a complex interaction between these two parameters. The optimal combination of these

parameters is where both values are the lowest for a minimal overall error value ($d = 0.28, o = 0.89$). While the user can interactively choose any distortion and overlap level, we use the minimum of the combined optimization function for a good initialization of distortion and overlap level.

Conclusion

In this article, we present generalized scatter plots, a new technique for visualizing large amounts of multi-attribute data. The approach is a generalization of traditional scatter plots and solves the overlap problem. Our technique maps each data point to one pixel of the display. We implemented overlap-optimized pixel placement technique to place identical data points in a neighborhood around the already plotted pixels. In our system, we enable the users to smoothly vary the degree of overlap and level of distortion, in order to generate the best view for their applications. The generalized scatter plots system is an efficient and effective solution to the overlap problem, which allows scatter plots to be used for the exploration of very large data sets. We apply our technique to real data sets dealing with telephone service usage analysis, server performance monitoring applications and financial data to demonstrate the wide applicability of our technique. The results show that our technique provides significantly more information than regular scatter plots.

An additional advantage of the overlap optimization technique is that it allows users to conduct and apply automatic analysis techniques. We demonstrate that k-Means algorithm and Voronoi tessellation are highly effective, reveal more information and allow the extraction of hidden patterns when using moderate degree of overlap and high degree of distortion. A considerable aspect in applying clustering algorithms in overlap optimized scatter plots is when decreasing the degree of overlap, some artificial clusters may appear and bias users' perception. Therefore, we suggest to extend the computation of optimal degree of overlap as a function of clustering ability and error. It is also to further research to find other techniques and make them applicable to overlap-optimized scatter plots. Especially the computation of correlation is a challenge and will be addressed in future research.

We also evaluate our generalized scatter plots according to an automatic optimization function, and show that an optimal compromise between overlap and distortion exists. Our future work is to automate the interpolation process between distortion and overlap.

References

1 Cleveland, W.S. (1984) The many faces of a scatterplot. *Journal of the American Statistical Association* 79(388): 807–822.
2 Eick, S.G., Steffen, J.L. and Summer, E.E. (1992) Seesoft-A tool for visualizing line oriented software statistics. *IEEE Transactions on Software Engineering* 18(11): 957–968.
3 Eick, S.G. (1994) *Data Visualization Sliders*. New York: ACM, pp. 119–120.
4 Jerding, D.F. and Stasko, J.T. (1995) The information mural: A technique for displaying and navigating large information spaces. In: N.D. Gershon and S.G. Eick (eds.) *INFOVIS*. InfoVis' 95: Proceedings of the 1995 IEEE Symposium on Information Visualization; 30–31 October, Atlanta, GA, USA. Infovis, pp. 43–50.
5 Spence, R. and Apperley, M. (1999) *Data Base Navigation: An Office Environment for the Professional*. San Francisco, CA, USA: Morgan Kaufmann Publishers.
6 Furnas, G.W. (1986) Generalized fisheye views. *SIGCHI Bull* 17(4): 16–23.
7 Bertini, E. and Santucci, G. (2006) Give chance a chance: Modeling density to enhance scatter plot quality through random data sampling. *Information Visualization* 5(2): 95–110.
8 Ellis, G. and Dix, A. (2007) A taxonomy of clutter reduction for information visualisation. *IEEE Transactions on Visualization and Computer Graphics* 13(6): 1216–1223.
9 Fua, Y.H., Ward, M.O. and Rundensteiner, E.A. (1999) *Hierarchical Parallel Coordinates for Exploration of Large Datasets*. IEEE Computer Society Press, pp. 43–50.
10 Zhang, T., Ramakrishnan, R. and Livny, M. (1996) BIRCH: An efficient data clustering method for very large databases. In: H.V. Jagadish and I.S. Mumick (eds.) *SIGMOD conference*. Proceedings of the 1996 ACM SIGMOD International COnference on Mangagement of Data; 4–6 June, Montreal, Quebec, Canada. ACM Press, pp. 103–114.
11 Buering, T., Gerken, J. and Reiterer, H. (2006) User interaction with scatterplots on small screens – A comparative evaluation of geometric-semantic zoom and fisheye distortion. *IEEE Transactions on Visualization and Computer Graphics* 12(5): 829–836.
12 Keim, D., North, S., Panse, C. and Schneidewind, J. (2002) Efficient cartogram generation: A comparison. *INFOVIS*. 2002 IEEE Symposium on Information Visualization (InfoVis 2002); 27 October – 1 November, Boston, MA, USA. IEE Computer Society, pp. 33–36.
13 Unwin, A., Theus, M. and Hofmann, H. (2006) *Graphics of Large Datasets: Visualizing a Million (Statistics and Computing)*. Secaucus, NJ: Springer-Verlag, New York.
14 SAS Institute Inc, JMP Software, http://www.jmp.com/software.
15 Carr, D.B., Littlefield, R.J., Nicholson, W.L. and Littlefield, J.S. (1987) Scatterplot matrix techniques for large N. *Journal of the American Statistical Association* 424–436.
16 Hao, M.C., Dayal, U., Keim, D.A., Morent, D. and Schneidewind, J. (2007) Intelligent visual analytics queries. VAST '07: Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology. Washington DC, USA: IEE Computer Socitey, pp. 91–98.
17 Keim, D.A., Panse, C., Schafer, M., Sips, M. and North, S.C. (2003) HistoScale: An efficient approach for computing pseudo-cartograms. VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS '03); Washington DC, USA: IEE Computer Society, p. 93.
18 Bresenham, J. (1977) A linear algorithm for incremental digital display of circular arcs. *Commun ACM* 20(2): 100–106.