Enhancing Scatter Plots Using Ellipsoid Pixel Placement and Shading

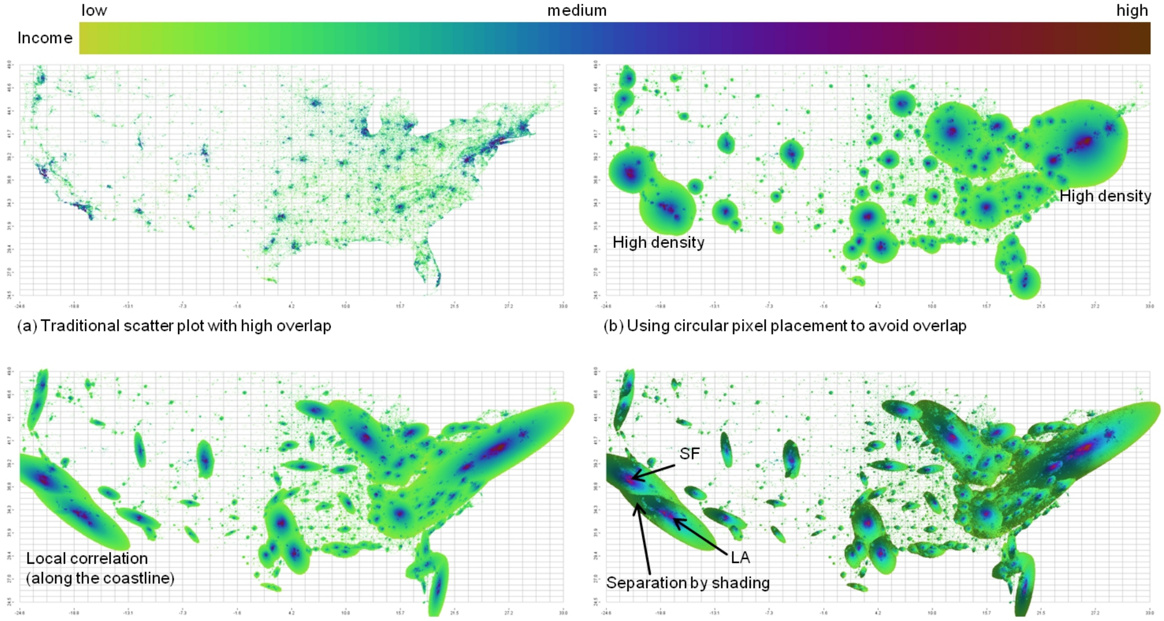
Halldór Janetzko
University of Konstanz
janetzko@dbvis.de

Ming C. Hao
Hewlett Packard Research Labs
ming.hao@hp.com

Sebastian Mittelstädt
University of Konstanz
mittelstaedt@dbvis.de

Umeshwar Dayal
Hewlett Packard Research Labs
umeshwar.dayal@hp.com

Daniel Keim
University of Konstanz
keim@dbvis.de



(a) Traditional scatter plot with high overlap

(b) Using circular pixel placement to avoid overlap

(c) Using ellipsoid pixel placement to avoid overlapping and to reveal local correlations

(d) Using shading to separate ellipsoids

Figure 1: US Census Data on a traditional scatter plot and three generalized scatter plots with different pixel placements: (b) circular pixel placement, (c) ellipsoid pixel placement without shading, and (d) ellipsoid pixel placement with shading. Note, that there are no overplotting points in figures (b), (c), and (d). (x-axis: Longitude, y-axis: Latitude, color: Income)

Abstract

Scatter plots are one of the most powerful techniques for visualizing relationships between two continuous variables. Using scatter plots, it is easy to find how one variable is affected by another. However, scatter plots may have a high degree of overlap, and therefore, important local patterns and trends may be hidden. Generalized scatter plots provide overlap-distortion optimized views, but the point repositioning algorithm used for avoiding overlap does not take the local structure of the data into account. In this paper, we enhance scatter plots using ellipsoid point placement and a cluster shading algorithm. In particular, we use local correlations to compute the rotation and aspect ratios of the ellipsoids used for the point placement, and add shading to the point clusters to visually encode the points' original locations. The effect of the shading and lighting can be controlled by the user.

1. Introduction

1.1. Motivation

To reveal correlations and trends in large multi-dimensional data, scatter plots are one of the most powerful and widely used techniques. They are intuitive and easy-to-use, but often have a high degree of overlap that may occlude a significant portion of the data values shown. For example, the traditional scatter plot in Figure 1(a) shows 333,488 income observations, but only about one thousand distinct points are visible in the scatter plot. In our previous work [1], we proposed a generalized scatter plot technique, which allows an overlap-free representation of large datasets to fit entirely into the display as shown in Figure 1(b).

The generalized scatter plots help to avoid the overlap problem by repositioning points to the nearest unoccupied screen position using a circular pixel placement. However, the overlapping points are not placed according to the relationship between two variables of the scatter plot. The generated dense area always has a circular shape, as shown by the east and west coasts in Figure 1(b) which is an artifact of the technique and may mislead users.

In this paper, we derive a new ellipsoid pixel placement technique to arrange overlapping points based on the local correlation of the two variables as shown in Figure 1(c). Users can quickly distinguish different orientations of the areas with a high local point density. The different orientations of the ellipsoids reveal local structures, e.g., on the east and west coasts and in the Chicago area, which are not visible in Figure 1(b). In Figure 1(d), we add shading to encode the original locations of the repositioned points, which is important in a geographical application, such as US census data.

1.2 Related work

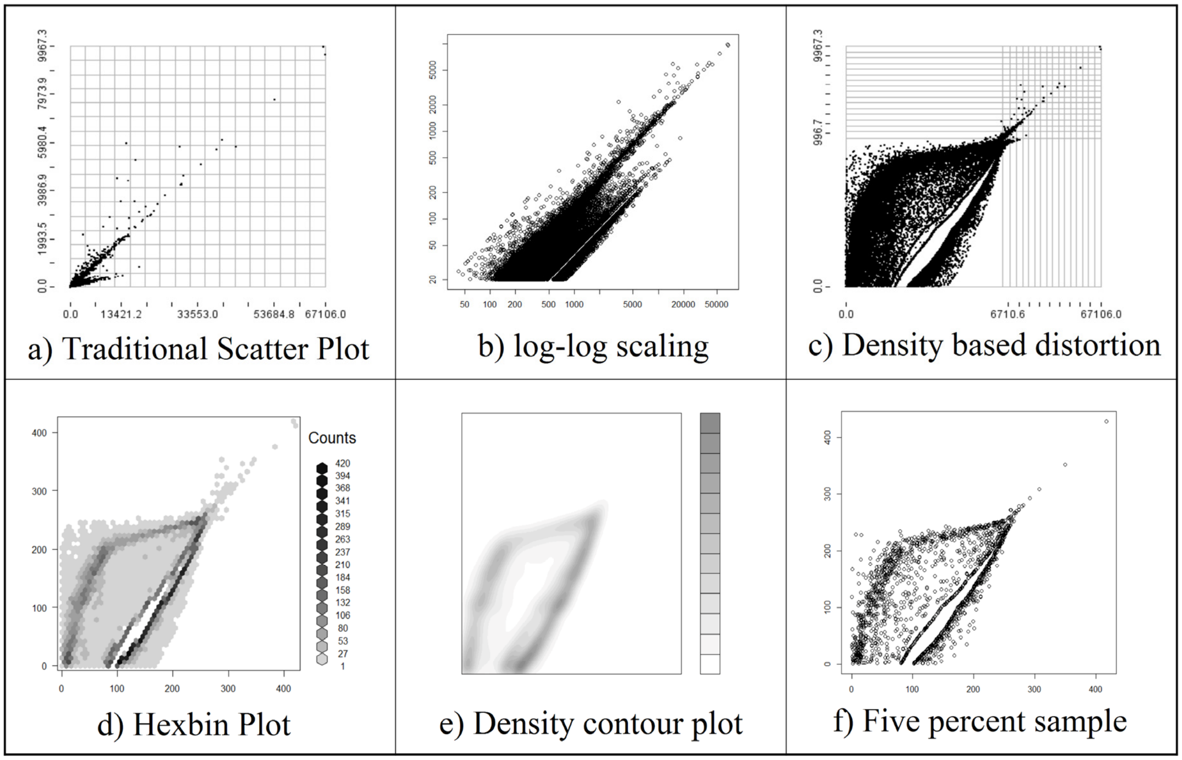
There are many extensions of scatter plots trying to solve the overlap problem of traditional scatter plots. In 1984, Cleveland [5] introduced sunflowers to draw overlapping points and used different glyphs to show the data density. Cleveland's approach is an improvement, but it does not consider the direction of local correlations. There are a number of interesting approaches to solve the overlap problem based on density, distortion, and animation as described below.

In 1999, Lee Wilkinson [8] suggested the usage of semi-transparency to make overlapping data points partially visible. JMP 8 Software [9] generated scatter plots with nonparametric density contours and marginal distributions to show where the data is most dense. Each contour line in a curved shape encloses 5% of the data. Another aggregation based visualization approach to detect data anomalies is presented by Maciejewski et al. in [26]. Carr et al. [4] used a hexagonal-shaped symbol with its size increasing monotonically as the number of observations in the associated bin increased, and HexBin scatter plots [12] determined the brightness value of each HexBin cell depending on the number of data points. Later, Bowman and Azzalini's smooth contour scatter plot [2, 3] applied smoothing techniques to show linearly increasing overlaps with different shades, and Bachthaler and Weiskopf's continuous scatter plots [13] built a continuous model out of the discrete data input. The Information Mural by Jerding and Stasko [23] uses anti-aliasing and greyscale to deal with overplotting when the number of data points exceeds the number of available pixels.

Distortion plays an essential role in avoiding overlap in scatter plots. Büring et al. [10] provided two 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 on small screens. Other well-known distortion techniques are cartograms which can be seen in [11] for example. In the book by Antony Unwin et al. [6], overlapping points are drawn in bright white on a darker background with slightly larger sizes. Unwin et al. also suggested using the alpha-transparency to represent data points. As a result, highly over plotted areas have high opacity and sparse areas have higher transparency. Later, Keim et al. [1] introduced generalized scatter plots to allow the analyst to optimize the degree of overlap and distortion to generate the best possible visualization as described in section 2 with more detail. Another technique dealing with replacing data points avoiding overplotting is presented by Trutschl et al. in [24]. The basic idea is to use neural networks (SOMs) in order to prevent overplotting while visually grouping similar objects. Using this grid-based approach comes with the disadvantage that there might be gaps as not every neuron may represent a data point. A simple kind of pixel placement is furthermore proposed in [25] by Aris et al. where overlapping points are replaced in a column- and row- based manner.

Robertson et al. [7] used animated scatter plots and small multiples to show trends in multi-dimensional data. Yu-Hsuan Chan et al. [14] used flow-based scatter plot techniques that extend 2D scatter plots with sensitivity coefficients to highlight local variations of one variable with respect to the other. Both Chan et al.'s streamline clustering and our ellipsoid pixel placement is able to show local trends. The overlap problem still remains in Chan et al.'s scatter plots but not in our new ellipsoid scatter plots.

Figure 2 compares six existing methods that are related to the method presented in this paper. The data used in Figure 2 for the representations is a telephone service usage data set with 37,787 records. The dataset shows the duration of the call on the x-axis and the total charge on the y-axis. The traditional scatter plot in Figure 2(a) shows some interesting linear patterns being unfortunately obscured by a high amount of overplotting. Figure 2(b) shows the same dataset with a logarithmic scaling of both axis. A density-equalizing distortion is applied in Figure 2(c) enlarging the high density region in the lower left. For a fair comparison, the same distortion is used for all subsequent visualizations as our technique is applied to the distorted data set in Figure 10 in section 5.2. Aggregation based methods like HexBin and the smooth contour scatter plots resulting from 2D kernel density estimations [2, 3] in Figures 2(d) and 2(e) do not suffer from overplotting as they display density and not the original data



a) Traditional Scatter Plot

b) log-log scaling

c) Density based distortion

d) Hexbin Plot

e) Density contour plot

f) Five percent sample

Figure 2: A comparison of related work dealing with the overplotting problem for telephone conference usage data (x-axis: duration of call, y-axis: charge for call).

points. Smoothed contour scatter plots show linearly increasing overlap with different shades. Another well known technique tackling overplotting is to apply sampling (see for example [22]) in order to reduce the number of points. In Figure 2(f) we used a five percent sampling with the result that dense regions will still remain visible while low-density patterns may disappear. Note that the techniques (a), and (b) do not tackle overplotting while (c) may reduce the amount of overplotting.

Concerning the applied shading approach there are several methods that apply a lighting model to enhance the visual representation. For instance Van Wijk et al. use lighting in [20] to make treemaps better readable and understandable. Furthermore, an approach similar to ours is followed by Willems et al. [21] to light aggregated trajectories with a technique similar to bump mapping. Note that our technique is also related to a three dimensional representation of kernel-density estimation [16], which only shows an aggregated view of the data instead of showing each individual data point. Our technique enables us to show an additional third dimension by color and access and interact with each data item being crucial while accessing meta-information stored for each data point.

1.3 Contributions

In this paper, we combine the best properties of the above methods. We enhance scatter plots with the following three novel techniques:

Ellipsoids replace circular pixel placement to show the local correlations between two variables. The direction and aspect ratio of the ellipsoid is the result of applying a local Principal Component Analysis (local PCA). Shading is added to the pixel placement to visually encode the original location before repositioning. For example, in the US census data set, if the points are repositioned without applying shading and lighting, it is difficult to determine the original location of these points. Each data point (pixel) is accessible and can be queried for detailed information.

2. Generalized Scatter Plots

Basic idea

Generalized scatter plots with circular pixel placement as shown in Figure 1(b) introduced in [1] are an overlap-optimized representation of large datasets that fit entirely into the display. In order to avoid overplotting a density-equalizing distortion is applied and a circular pixel placement algorithm is proposed to move data points still overplotting to the next free pixel position. The main contribution is to allow the analyst to optimize the degree of overlap and distortion to generate the best possible view. The authors assumed providing only the final result of distortion or pixel result may not result in the best possible visual representation. An intermediate state should be the optimal visualization according to some optimality criteria. Therefore, interaction possibilities were provided to the user allowing any intermediate state between a traditional scatter plot and a fully distorted view or a scatter plot without any overplotting.

Technique

To avoid overplotting the authors implemented methods changing the data point's position. Basically, two ways of dealing with any overplotting occurring in the scatter plot visualization were provided. One method is a density equalizing distortion which may reduce the occurring overplotting, as it enlarges regions with a high density and shrinks regions with low densities. The data space is partitioned and for each bin the density is determined and this information is used to resize each bin (shown in Figure 3). After applying a distortion technique there might still exist overplotting and the authors therefore propose a pixel placement approach replacing overplotting data points to the next free position in a circular fashion.

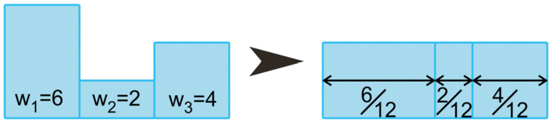


Figure 3: Distortion technique calculating relative density of regular grid cells and resizing the cells accordingly.

In Figure 4 we show a schematic explanation of the pixel placement algorithm. The basic idea is to iterate through all data points in an ordered manner (e.g., ordered by a third dimension color value) and check whether their original position is still unoccupied. If it is unoccupied, the data point can be placed there, and in the other case the next free pixel position has to be found. Therefore, a circle around the original position is calculated (green area in Figure 4) and all the pixels of the green area are checked, whether they are unoccupied and the first free position found is chosen. Note that while iterating through all data points processing them one-by-one and not in parallel, the final result can not contain any overplotting data points. Even when two circles may visually merge together (in the middle of Figure 4), there is no overplotting in the intersection area. The intersection area contains only single data points, namely the ones that were processed first. In the description above, we have omitted some details of the pixel placement algorithm and assumed for easiness of explanation that there is only a possibility to turn pixel placement on or off and there exists no intermediate state. But the algorithm is parameterized allowing the user to adjust the degree of overplotting to his needs. This allows also to cope with cases when the amount of data points exceeds the number of screen pixels.

Merits

Generalized scatter plots have variable degrees of distortion and variable degrees of overlaps. Traditional scatter plots are just special cases of generalized scatter plots with no distortion and a data induced overlap, as shown in Figure 2(a). As in normal scatter plots, each data point is presented as one color pixel, and users can move a pointer to see the content. In the distorted and/or overlap-optimized generalized scatter plots, data values are placed as overlap-free as possible and as closely as possible to their original positions.

Generalized scatter plots enable analysts to use color to visualize the third attribute for identifying patterns, while the previous approaches use color to represent density. For example, color can be mapped to "income", as shown in Figure 5. In Figure 5, analysts can quickly identify three different types of income (low: blue; medium: green; and high: red) by three different colors for the west coast. Analysts are able to quickly observe the income distribution and relationships in the dense area, such as the largest cluster, formed by low income families (blue). The smallest circles represent high income families (red).

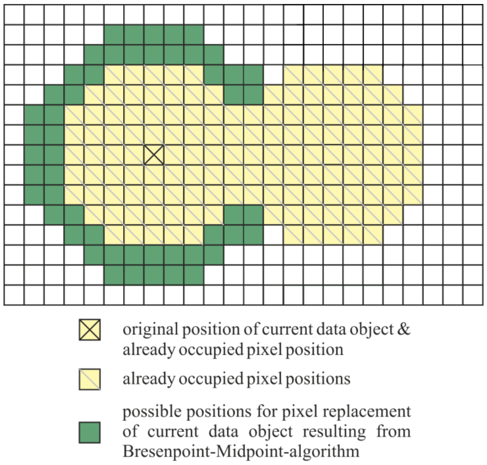

original position of current data object & already occupied pixel position
already occupied pixel positions
possible positions for pixel replacement of current data object resulting from Bresenpoint-Midpoint-algorithm

Figure 4: Schematic explanation for the circular pixel placement algorithm, which is used to avoid overlap of points at all.

Limitations

The generalized scatter plots described above use a circular arrangement of the overlapping points without taking the local distribution of data points into account. Therefore, analysts can only find overall patterns and trends. This overall information is important but not sufficient. Analysts want to find the relationships between two variables in the local high density area, such as on the longitude and the latitude in the west coast area in Figure 5. The other serious drawback of repositioning overlap points is that the original location of a replaced point is lost. As a result, the local correlation cannot be detected.

Analysts have difficulties to recognize the local structure on the west coast area from the display shown in Figure 5.

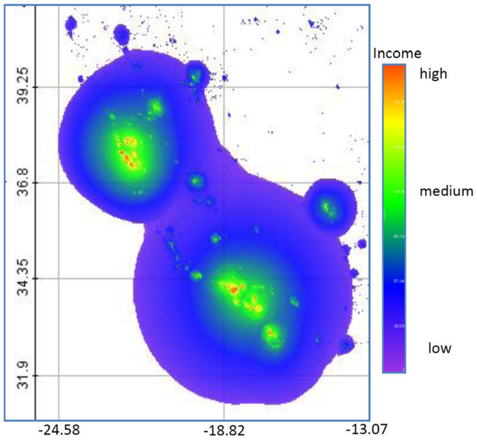


Figure 5: US Census west coast analysis using circular pixel placement to avoid overlapping of data points. (x-axis: Longitude, y-axis: Latitude, color: Income).

3. Generalized Scatter Plots with Ellipsoid Pixel Placement

As mentioned above, Generalized Scatter Plots have several limitations. The circular arrangement of the overlapping points, for example, introduces artifacts that are not related to the properties of the data, and there are many other ways to arrange the overlapping points. In this paper, we propose an ellipsoid point placement based on the local correlations of the overlapping data points to decrease the visual artifacts and follow a more data-driven approach.

We tackle also another problem of Generalized Scatter Plots, namely the merging of point clouds originating from different centers and furthermore enhance the visual mapping of a replaced point to its original position. In the example, shown in Figure 5, it is in some cases very difficult to determine the original positions of the data points. Our way to solve this problem is to add a modest degree of shading or more specifically, bump mapping [18] to visually encode the structure of the underlying data.

In the following subsections, we will discuss both techniques in more detail.

3.1 Ellipsoid Pixel Placement

A typical reason for using scatter plots is to search for local and global correlations and clusters. Global correlations are, for instance, patterns of the whole data set described by a global model (e.g., regression). Local correlations in contrast are patterns of parts of the data described by a local model (e.g., regression per cluster). By using a circular pixel placement, we lose information about the local correlations of the two variables, because they are hidden by the circular visual artefacts. In our new approach, we use an ellipsoid pixel placement to show the local properties of the data. The parameters of the rotated ellipsoid shape used to place the data overlapping data points is based on the results of a local correlation analysis to represent the strength and direction of the correlation.

The first task in determining the local correlations is to partition the data set into disjoint subsets with clusters of overlapping data points. As local correlations can vary over the whole data range it is important to partition the data set appropriately. A simple method would be to partition it into a regular grid, but this may again result in artefacts. We therefore decided to apply a partitioning clustering technique in order to determine the local correlations per cluster.

In our current implementation, we use the OPTICS algorithm [19] for partitioning the data set. We support the user in finding the right epsilon parameter by providing an interactive representation of the reachability and core distances. The user can set the threshold for partitioning and gets immediate visual feedback of the resulting clustering.

After partitioning the data set, we have to determine the local correlation of each partition. To enhance the visibility of the local correlations, we calculate the direction and strength of the correlation for each cluster by applying a Principal Component Analysis to determine the most dominant local correlations. As input for the PCA, we use the 2x2 correlation matrix for each cluster. The result of the PCA provides two eigenvectors (directions of the local correlation) and two eigenvalues (strength of

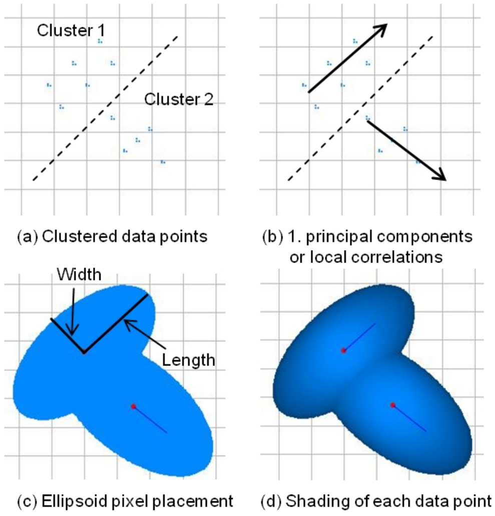

Cluster 1
Cluster 2
(a) Clustered data points
(b) 1. principal components or local correlations
Width
Length
(c) Ellipsoid pixel placement
(d) Shading of each data point

Figure 6: Overall process of ellipsoid pixel placement and shading.

the local correlation).

The strongest eigenvector as shown in Figure 6(b) with the highest eigenvalue is used for the orientation of the ellipse, which means that the first half axis (length) of the ellipse is parallel to the strongest eigenvector in Figure 6(c). The ratio of length and width is calculated by the ratio of the first two eigenvalues, which determines the second half-axis or the width of the ellipse. We allow a minimal ratio of 1:4 to avoid to narrow ellipses in case of a perfect linear correlation.

The rotation and aspect ratio of the ellipse is used as an input to a modified Bresenham algorithm which is adapted to drawing a rotated ellipsoid. We take the output of a standard ellipse algorithm and use affine transformations for the rotation. The pixel placement algorithm iterates over all data points ordered by value used for coloring and checks whether their original position is unoccupied. In cases where the data point has to be moved, positions on the ellipsoid are calculated and checked whether they are still unoccupied and can be used for the relocation of overlapping points. Algorithm 1 shows the pseudo code of our ellipse drawing algorithm which is a variant of the Bresenham algorithm [15], which is extended by affine transformation for the use of ellipsoid pixel replacement.

```
Algorithm: calcEllipsoidPoints(Point originalLocation, Number length)
 //List to store calculated points on ellipse
 List<Point> resultPoints;
 //Get ellipsoid information at current location
 Ellipsoid e = getEllipsoid(originalLocation);
 //Compute width according to the local correlation
 Number width = length*e.getAspectRatio();

 //Bresenham algorithm for ellipse plotting
 Number dx = 0, dy = width;
 Number length2 = length*length;
 Number width2 = width*width;
 Number error = width2 - (2*width - 1)*length2;
 do
   Point[ ] p = {(dx,dy),(-dx,dy),(-dx,-dy),(dx,-dy)};
   for i = 1 to 4
     p[i].rotate(e.getOrientation());
     p[i].translate(originalLocation);
     if p[i] does not violate paint borders then
       resultPoints.add(p[i]);
   next i
   if 2*error <  (2*dx+1)*width2 then dx++; error += (2*dx+1)*width2;
   if 2*error > -(2*dy-1)*length2 then dy--; error -= (2*dy-1)*length2;
 while dy >= 0

 return resultPoints;
end
```

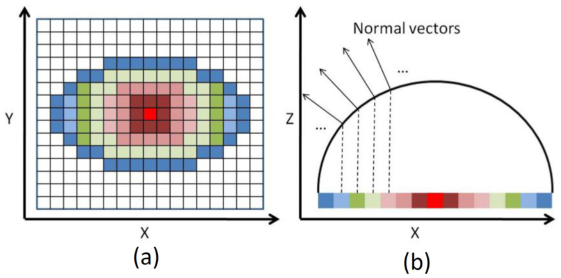
Algorithm 1: Determine ellipsoid points with extended Bresenham algorithm for pixel placement.

3.2 Combining Shading with Pixel Placement

A serious drawback of all pixel placement techniques is that the original location of the repositioned points is lost, even if they are placed at the nearest free location. Also, local correlations may be hidden as shown in Figure 6. We therefore add a visual encoding to the pixel placement to represent the origin by applying a variation of Bump Mapping [18] shown in Figure 6(d).

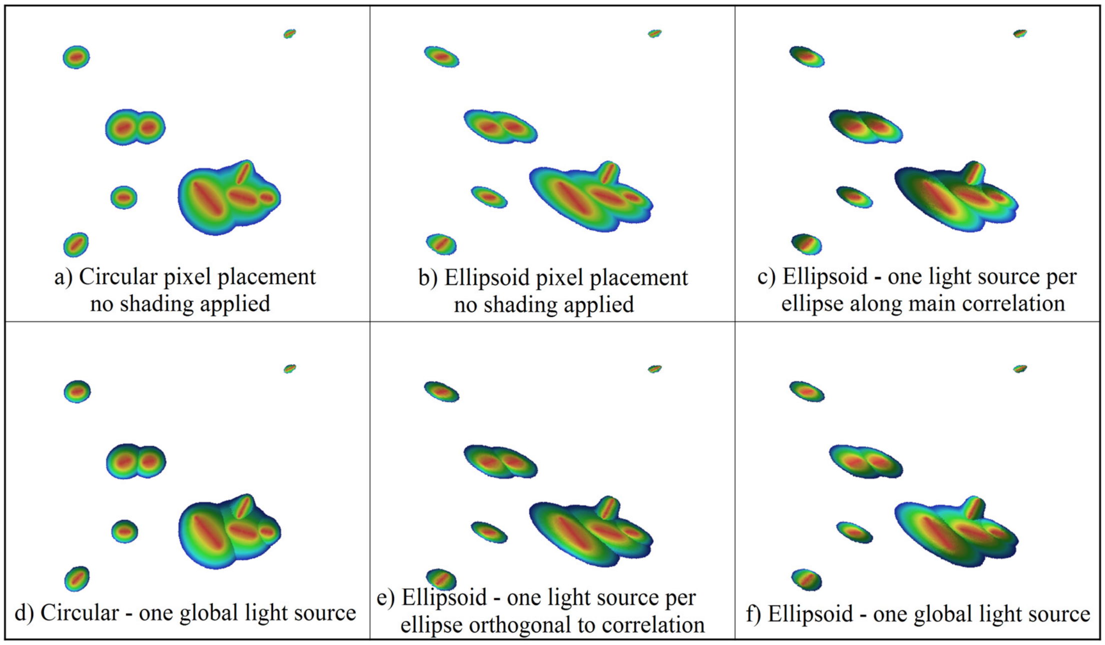
We assume the points belonging to an overlapping pixel to be a small hill and calculate the normal vectors accordingly (see Figure 8). In this way, after applying the pixel placement to all clusters we get a three-dimensional landscape out of the two-dimensional scatter plot. From the landscape we calculate the normal vectors, which are then used for light calculations, in our case Phong shading [17]. For each data point we determine how much light is reflected by the data point, and this information is blended on top of the point's color information. By the shading, the user perceives the center point of the hill and may easily conclude where the original location of the data point is located. Since the lighting decreases the expressiveness of the coloring, we allow the user to control the strength of the effect, providing the user with the possibilities either seeing the third dimension values or seeing the point's origin more clearly.

The shading technique is closely connected to the outcome of the ellipsoid pixel placement. In order to have a shading of each ellipsoid, we have to calculate the normal vectors for the light computation accordingly. Therefore, we determine the normal vectors for each data point by weighting the distance from the original position according to the half-axis of the ellipsoid to which it belongs. In calculating the illumination of each pixel, we only consider the diffuse term of the Phong Illumination Model [17], making the intensity proportional to the angle between the normal vector and the outgoing vector of the light source. Figure 7 depicts the shading algorithm. The parameter alpha can be adjusted by the user via an "Ambient Light" slider. If the slider is dragged to the very right, the shading does not change the intensity of the data points' color. Thus, the users can adjust the visualization result according to their needs.



(a) (b)

Figure 7: Schematic approach to calculate the normal vectors which are used for shading



a) Circular pixel placement no shading applied

b) Ellipsoid pixel placement no shading applied

c) Ellipsoid - one light source per ellipse along main correlation

d) Circular - one global light source

e) Ellipsoid - one light source per ellipse orthogonal to correlation

f) Ellipsoid - one global light source

Figure 8: Comparison of different shading variants, all providing a non-overlapping view to the data, even though it seems as if the ellipsoids merge and seem to overlap each other.

An important factor influencing the shading result is the location of the light source. Our application suggests four locations for the light source. Most patterns in the ellipsoid landscape are best visible if the scene is illuminated in a direction vertical or parallel to the orientation of ellipsoids. Therefore, the application assigns local light source to each ellipse in infinite distance, either parallel or vertical to the orientation of the ellipsoid. The user is free to select one of these suggested options or to choose manual another location for a global light source.

4. Evaluation

The new pixel placement technique was developed to enhance the visibility of local patterns. Previous techniques, for instance, circular pixel placement, just used the nearest free position without regard for the underlying local correlations. For a comparison of both approaches we applied them to a data set containing several different local correlation patterns, as shown in Figure 8. The example shows the generated dataset containing a small number of clusters with random positions, random size, and random local correlations.

Obviously, the ellipsoid pixel placement (b) visualizes more of the underlying local correlation patterns than the circular one (a). But even more interesting from a data analyst's perspective is the visibility of the different directions of the local correlations.

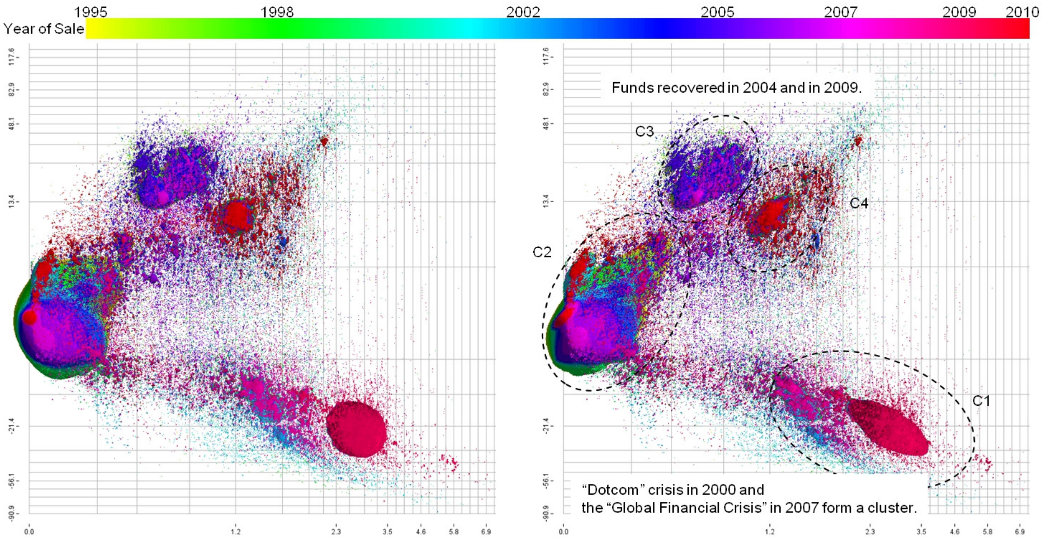
Another important issue is the effect of shading on the perception of pixel color. The shading indicates the origin of repositioned data points as this information is lost when replacing over plotting data points but it also modifies the point's color, making them darker if the light source does not illuminate them orthogonally. Figure 8 shows the effect of different shading variants. Figure 8 (c) shows the illumination with one light source per ellipse along the dominating local correlation of each cluster; in Figure 8 (e) the ellipsoids are illuminated orthogonal to the cluster orientation; and Figure 8 (f) shows a global illumination with only one light source orthogonal to the main correlation of the whole data set. Our expectation was that (c) would provide the best results since it best enhances the visibility of local correlations but the results of our informal evaluation show that (e) and (f) provide better results depending on the application scenario as each of them enhances the visibility of different local correlations. For completeness Figure 8 (d) depicts the illumination applied to the circular pixel placement result of (a)

Our technique, therefore, uses the shading from the overall main direction as the default but also allows selecting the perspective orthogonal to the main direction. Advanced users may change the position of the light source to any position they want. In addition, the users may control the strength of the shading or switch the shading off completely to see the original color of the data points better. We also use color maps that are based more on the change of the color hue than on intensity. The problem with an intensity-based colormap is that differently colored points may become identically colored because of the shading.

5. Applications

5.1 Financial analysis

The main task of financial analysis is to support the investor in the decision process of purchase and sale. In order to explore the whole market and to capture the development over time, we can apply our



(a) Circular pixel placement with shading (b) Ellipsoid pixel placement with shading

Figure 9: The temporal performance-risk analysis of 130.000 American funds from 1995 to 2010. (x axis: risk of an investment, y axis: performance of an investment, color: investment period of one year, e.g. red: purchase 2009, sale 2010).

technique in the performance-risk analysis. The performance captures the wins and losses over a certain holding time (from purchase to sale). Funds, which have many up- and downturns during this time, are very risky. This can be measured by the standard deviation over the holding time. Figure 9 shows the performance (y-axis) and risk (x-axis) of about 130,000 American funds over 15 years. For each of those funds, we determined the performance for one-year intervals. Consequently, each of the funds will result in 14 data points (15 one-year intervals from 1994 to 2010). The color represents the year of sale or respectively the begin of the one-year interval. The red color, for example, encodes the performance/risk of the funds from 2009 (year of purchase) to 2010 (year of sale). We chose a rainbow color scale in order to visually separate the years, which are considered in our case to be ordinal instead of continuous.

The sparse area in the middle of the subfigures in figure 9 indicates that high risks implicate very high performances (negative and positive). The more interesting finding is that the funds form four clusters (C1-C4). C1 seems to cover the big financial crisis in the recent years like the "Dotcom" crisis in 2000 and the "Global Financial Crisis" from 2007-2009, which are clearly separated from the other clusters. In contrast to them, this cluster shows a clear negative local correlation of performance and risk. C4 shows interesting trends as well. It is surprising that right after the crisis most of the funds recovered very fast at the cost of high risks. This indicates some "gambling" in those years. The other few funds of 2010 can be found in C2, which shows low risk but positive performance. Cluster C3 contains the years 2005-2007. After the "Dotcom" crisis the market recovered slowly from 2004 on, ending in a "boom" in 2007 just before the "Global Financial Crisis".

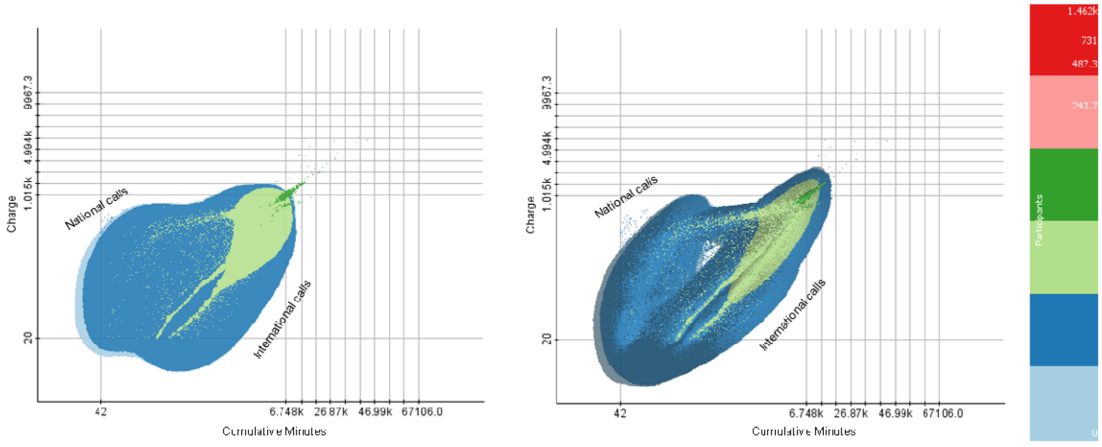
5.2 Telephone Service Usage Analysis

The application uses a telephone data set containing 37,787 telephone conference records. IT service mangers use the data to analyze the usage patterns and correlations among different attributes (i.e., charges, the duration of the call, and the number of participants) in order to detect potential for cost savings. Figure 10 shows both the results from the circular and the ellipsoid pixel placement including shading for shifted points. Both scatter plots are arranged such that call duration is mapped to the x axis and charge is mapped to the y axis. The color of the pixels represents the number of participants.

Analysts are able to use the generalized circular scatter plots shown in Figure 10(a) to find the calling distribution and overall correlation between the duration of the calls, the costs, and the number of participants. In the high density areas, however, it is difficult to determine the local relationships between these variables area even if distortion is used. Data points that overlap each other in traditional scatter plots form circular clusters in the generalized scatter plot. The ellipsoid pixel placement adds information about the local correlation of the overlap data points and the shading encodes the relationship to their original position. As a result, the partitioning between the two clusters (national and international calls) which are merged in the circular case becomes clear with our new ellipsoid pixel placement.

Analysts are able to learn important additional facts from the data, demonstrating the additional value of our ellipsoid scatter plots. By assessing the single data points analysts were able to make the following observations as meta-information about the calls like provider or time are available and each data point is accessible:

1. The left curve (national calls) illustrates that the most expensive calls have a high volume (many



(a) 60 % distortion , circular pixel placement

(b) 60 % distortion , ellipsoid pixel placement

Figure 10: Telephone service scatter plots with the same amounts of distortion, but different pixel placement techniques. Figure (a) shows the result of the circular arrangement and Figure (b) the results from the application of the new ellipsoid pixel placement combined with shading. The new pixel placement technique splits the correlation (see empty spot in the middle), whereas they are merged by the circular pixel placement. In both figures are no overlapping points at all. (x-axis: call duration, y-axis: call charges, color: the number of participants (logarithmic scale)).

data points) and correlate with the duration time. However, there is a wide distribution incharges. Interestingly, national calls are more expensive than international calls. But there seems to be a quantity discount for national calls as the slope of the charge decreases with increasing duration.

2. The right section (international calls) contains the international calls. There are two green / blue lines representing different service providers (AT&T and Sprint). The rightmost curve has the highest number of calls (AT&T representing the thickest curve).
3. The thickness of the curves in figure 10(b) reveals the number of national and international calls. From the comparison of the detailed structure of the curves, we learn that the international calls have a clear charge structure for each provider (solid lines) while the charges for national calls are more scattered and depend on other parameters not shown in the visualization (e.g., time of the day).
4. The new ellipsoid pixel placement separates the local patterns better. While both figures 10(a) and 10(b) were generated with the same amount of distortion, the ellipsoid arrangement and shading enables the user to differentiate the patterns better.

5.3 US Census Data Analysis

To show the general applicability and power of our technique, we also applied our technique to the block level median household income data of the US census data set. Figure 1(a) shows a traditional scatter plot with longitudes and latitudes mapped to the x and y coordinate axes, resulting in a normal map with a maximum data-induced overlap. Figure 1(b) shows the circular placement without any overlap; Figure 1(c) shows our ellipsoid pixel placement without shading; and Figure 1(d) shows our ellipsoid pixel placement with shading. Colors indicate household income with blue for low income, green for medium and red for high. The visualizations in Figures 1(c) and 1(d) show the advantage of our ellipsoid pixel placement and shading technique. High income areas that are completely hidden in the traditional scatter plot in Figure 1(a) become visible. A nice feature is that the local distribution patterns remain intact and can be easily detected and interpreted by the analyst. At the same time, depending on the overlap level, all data records are visible and accessible for further inspection. Analysts are able to easily see the longitudinal and latitudinal correlations with strong coastal orientations on the east and west coasts. Applying the shading helps in separating the different population centers from each other, which are merged by the pixel placement techniques.

6. Conclusions

In this paper, we enhance general scatter plots using ellipsoid pixel placement and shading for visualizing large volumes of high density data. Our techniques map each data point to one pixel on a display and each data point is accessible to the user for visual queries. The ellipsoid pixel placement algorithm does not only solve the overlap problem of data points, but also retains the direction of local correlations. Furthermore, the combination of shading with pixel placement nicely encodes the point's original locations. The effect of the shading and lighting of the data points may be controlled by the user. Users can easily adjust the light to change the shading either to see the third dimension values correctly or to see the point's original location.

We have applied our new techniques to three real business data sets dealing with American fund analysis, and telephone services. Also, we have used ellipsoid placement and shading to visualize US census data with known geographical structures. A series of informal evaluations have been conducted to determine how the ellipsoid pixel placement improves the existing techniques and how much shading/lighting is best. Our future work includes further studies to evaluate the effect of different degrees of overlap, distortion, and shading, and to further optimize these parameter settings. Furthermore, we want to measure the distance the points are moved comparing circular and ellipsoid pixel placement.

References

[1] Keim, D. A., Hao, M. C. Dayal, U., Janetzko, H., and Bak, P., Generalized Scatter Plots. Information Visualization Journal (IVS), 2009.

[2] Bowman, A. W. and Azzalini, A. (1997). Applied Smoothing Techniques for Data Analysis: the Kernel Approach With S-Plus Illustrations. Oxford University Press, Oxford, USA, 1997.

[3] Bowman, A. W. and Azzalini, A. (2003). Computational aspects of nonparametric smoothing with illustrations. Computational Statistics and Data Analysis, 42(4): 545 –560, 2003.

[4] Carr, D. B., Littlefield, R. J., Nicholson, W. L., and Kuttkefuekdm, J. S. Scatterplot Matrix Techniques for Large N, "Journal of American Statistics Association", 82, 424-436. 1987.

[5] Cleveland W. S. The Many Faces of a Scatterplot. Robert McGill Journal of the American Statistical Association, Vol. 79, No. 388 (Dec. 1984), pp. 807-822.

[6] Unwin A., Theus, M., and Hofmann, H., Graphics of Large Datasets: Visualizing a Million Series, Statistics and Computing 2006, XIV, Springer, New York / US.

[7] Robertson, G., Fernandez, R., Fisher, D., Lee, B., and Stasko, J., Effectiveness of Animation in Trend Visualization. IEEE Transaction on Visualization and Computer Graphics, Vol. 12, No. 5, September/October 2008.

[8] Wilkirnson, L. The grammar of graphics, New York, Springer, 1999., Ohio, USA.

[9] JMP 8 Software. www.jmp.com/software, New 64-bit computers and visual analytics tools.

[10] Büring, T., Gerken, J. and Reiterer, H. User Interaction with Scatterplots on Small Screens. IEEE Transaction on Visualization and Computer Graphics, Vol. 12, No. 5, September/October 2006.

[11] Keim, D. A., North, S. C., Panse, C., and Schneidewind, J. Efficient Cartogram Generation: A Comparison. Information Visualization, pp. 33-36, MA, October, 2002.

[12] HexBin Scatter Plots released by R System in January, 2009, https://stat.ethz.ch/pipermail/r-help/2009. Documented http://rss.acs.unt.edu/Rdoc/library/hexbin/doc/hexagon_binning.pdf.

[13] Bachthaler S. and Weiskopf D. Continuous Scatterplots, IEEE Transactions on Visualization and Computer Graphics, Vol. 14, No. 6, November/December 2008.

[14] Chan, Y., Correa, C., and Ma, K. Flow-based Scatterplots for Sensitivity Analysis. IEEE Symposium on Visual Analytics Science and Technology. October, 2010. Salt Lake City, Utah, USA.

[15] Zingl, A. The Beauty of Bresenham's Algorithm, Vienna, Austria, 2011, http://free.pages.at/easyfilter/bresenham.pdf

[16] Parzen, E. On Estimation of a Probability Density Function and Mode, Annals of Mathematical Statistics, Vol. 33, No. 3, pp. 1065-1076, 1962.

[17] Phong, B. T. Illumination for computer generated pictures, Communications of the ACM, Vol. 18, No. 6, 1975.

[18] Blinn, J. F. Simulation of wrinkled surfaces, ACM SIGGRAPH Computer Graphics, Vol. 12, No. 3, pp. 286-292, 1978.

[19] Ankerst, M. and Breunig, M.M. and Kriegel, H.P. and Sander, J. OPTICS: Ordering points to identify the clustering structure. ACM SIGMOD Record, Vol. 2, pp. 49-60, 1999.

[20] Van Wijk, J.J., and Van de Wetering, H. Cushion treemaps: visualization of hierarchical information, IEEE Symposium on Information Visualization, pp. 73 – 78, 1999.

[21] Willems, N., H. van de Wetering, J.J. van Wijk Visualization of Vessel Movements. Proceedings EuroVis, Vol. 28, No. 3, pp. 959-966, 2009.

[22] Bertini, E. and Santucci, G.. Give chance a chance: modeling density to enhance scatter plot quality through random data sampling. IEEE Symposium on Information Visualization, 5(2):95–110, 2006.

[23] Jerding, D. F. and Stasko, J. T. The information mural: a technique for displaying and navigating large information spaces. IEEE Transactions on Visualization and Computer Graphics, Vol. 4, No. 3, pp. 257–271, 1998.

[24] Trutschl, M., Grinstein, G., and Cvek, U. Intelligently Resolving Point Occlusion, IEEE Symposium on Information Visualization, pp. 131– 136, 2003.

[25] Aris, A. and Shneiderman, B. Designing semantic substrates for visual network exploration, Information Visualization Journal, Vol. 6, No. 4, pp. 281–300, 2007.

[26] Maciejewski, R., Rudolph, S., Hafen, R., Abusalah, A., Yakout, M., Ouzzani, M., Cleveland, W.S., Grannis, S.J., Ebert, D.S.. A Visual Analytics Approach to Understanding Spatiotemporal Hotspots. IEEE Transactions on Visualization and Computer Graphics, 16(2): 205-220, March/April 2010.