# Intelligently Resolving Point Occlusion

Marjan Trutschl*
LSU Computer Science
LSU Health Sciences Center
Shreveport, LA

Georges Grinstein†
Computer Science Department
University of Massachusetts
Lowell, MA

Urška Cvek‡
Computer Science Department
University of Massachusetts
Lowell, MA

## Abstract

Large and high-dimensional data sets mapped to low-dimensional visualizations often result in perceptual ambiguities. One such ambiguity is overlap or occlusion that occurs when the number of records exceeds the number of unique locations in the presentation or when there exist two or more records that map to the same location. To lessen the affect of occlusion, non-standard visual attributes (i.e. shading and/or transparency) are applied, or such records may be remapped to a corresponding jittered location. The resulting mapping efficiently portrays the crowding of records but fails to provide the insight into the relationship between the neighboring records. We introduce a new interactive technique that intelligibly organizes overlapped points, a neural network-based Smart Jittering algorithm. We demonstrate this technique on a scatter plot, the most widely used visualization. The algorithm can be applied to other one, two, and multi-dimensional visualizations which represent data as points, including 3-dimensional scatter plots, RadViz, polar coordinates.

**CR Categories**: I.3.6 [Computer Graphics]: Methodology and Techniques; H.5.2 [Information Interfaces and Presentation]: User Interfaces; I.3.3 [Computer Graphics]: Picture/Image Generation; I.2.6 [Artificial Intelligence]: Learning -- Connectionism and Neural Nets

**Keywords**: data visualization, information visualization, design, data points, data density, occlusion, identifiable points, jitter, neural networks

## 1 Introduction

The explosion of data generated from physical and natural sciences, instruments, biomedical and other scientific research not only creates the need for analysis of the data and interpretation of results, but also the need for development of tools and methods that can handle this data. High data dimensionality is challenging, since typical representations are 2-dimensional or 3-dimensional and we have to resort to projections of multiple dimensions onto this space, in order to display the dimensions.

*e-mail: mtrutsch@pilot.lsus.edu
†e-mail: grinstein@cs.uml.edu
‡e-mail: ucvek@cs.uml.edu

IEEE Symposium on Information Visualization 2003,
October 19-21, 2003, Seattle, Washington, USA
0-7803-8154-8/03/$17.00 ©2003 IEEE

In addition, these data sets suffer from a variety of problems, from missing records to overlap and uncertainty. When occlusion is substantial, it becomes difficult to interpret the relationships between records, and the influence of occluded points cannot be perceived.

Whenever we have overlapping points, the intrinsic dimension and the display capability of the visualization is reduced. Methods that resolve overlap, based on data set itself, inherently increase the intrinsic dimension or the display capability of the visualization. Given an $n$-dimensional space, the intrinsic dimension of a visualization [Grinstein et al. 2001] is defined as the largest $k$, $k \leq n$ for which a set of $k$ unit vectors in an $n$-dimensional space can be uniquely identified (perceived) in the visualization. The intrinsic dimension of a two-dimensional scatter plot is 2: $n$ unit vectors project to 3 points, (0, 0) and either (0, 1) or (1, 0), only two of which obviously come from unique points.

The main goal of this work is to improve and design a new method for treatment of overlapping data, when records are located within the same physical space, either in linear or non-linear projections. We introduce an intelligent jittering algorithm based on the Kohonen self-organizing map, which produces spatial ordering of the overlapping records on the new jittered surface. This technique is geared towards a variety of two- and three-dimensional visualizations used in exploratory stages of knowledge discovery.

Occlusion in three-dimensional spaces has been extensively studied and approached similarly in two-dimensions [Wong and Bergeron 1997; Eick 2000]. In these spaces, occlusion is largely due to the placement of records behind the object the user is viewing. The most common approach is semi-transparency, where the object is made translucent and additional objects are seen through [Zhai et al. 1996]. In our physical space, where the projection is mostly two-dimensional, transparency of a single object provides only limited occlusion resolution. Another approach is to allow the user to manipulate data, in order to discover additional relationships, such as in the SDM system [Chuah et al. 1995]. Excentric labeling was devised to provide interactive labels for a selection of records by moving the cursor over the records, providing an insight into the values behind these records [Fekete and Plaisant 1999]. Additional techniques were introduced by Manson [1999], who utilized retinal properties (size, color and shape), secondary point properties (point border) and animation. The most common approach, used in several commercial packages, is to resolve occlusion using "jittering," displacing overlapping records over a wider area.

The paper is organized as follows. First, we present the basic intuition and mathematics behind occlusion, followed by the description of our new Smart Jitter approach and its benefits. We demonstrate the Smart Jitter on the very simple Fisher Iris data set using a scatter plot. The paper concludes with a summary and future work.

## 2 Background

Overlap or occlusion occurs when a data set of *n*-dimensional records is mapped to an *m*-dimensional visualization space, where $m < n$ or even $m << n$. Overlap Ω occurs when the number of records *r* in a data set exceeds the number of physical points in a visualization of size $v_x$ by $v_y$ (in case of a rectangular two-dimensional visualization).

$$\Omega : if\ r > v_x \cdot v_y \tag{1}$$

Overlap may also occur when there are at least two records $r_i$ and $r_j$ that are not unique with respect to their dimensional values *x* and *y* or their non-linearly projected *x* and *y* positions.

$$\Omega : if\ \exists\left(r_{i_x} = r_{j_x} \wedge r_{i_y} = r_{j_y}\right) \tag{2}$$

Both identities may be modified to reflect the properties of a 3-dimensional overlap.

$$\Omega : if\ r > v_x \cdot v_y \cdot v_z \text{ and}$$

$$\Omega : if\ \exists\left(r_{i_x} = r_{j_x} \wedge r_{i_y} = r_{j_y} \wedge r_{i_z} = r_{j_z}\right) \tag{3}$$

Overlap can also occur in scatter plots when the two presented dimensions have identical values. Moreover, visualizations based on non-linear projections from an *n*-dimensional space to an *m*-dimensional display (such as RadViz [Hoffman et al. 1999] and Star Coordinates [Kandogan 2000]) most often result in multiple overlapping records regardless of the fact that their dimensional values are unique. All visualizations exhibit similar behavior when analyzing large, high-dimensional data sets. In general, they fail to handle overlapping records and crowding, when there are a lot of points displayed in a limited display space.

If we assume that we have a two-dimensional visualization, a record maps to position (*x*, *y*) on the display. The most common technique to solve the overlap problem is "jittering," which offsets the record from its mapped location (*x*, *y*) on the two-dimensional output surface to location (*x'*, *y'*) with $x' = x \pm \Delta x$ and $y' = y \pm \Delta y$, $\Delta x$ and $\Delta y$ representing randomly generated offset distances. If more than two points map to the same (*x*, *y*) location, the jitter algorithm randomly generates $\Delta x$ and $\Delta y$ for each of the overlapping records, keeping the $\Delta x$ and $\Delta y$ within a predefined range. Chambers et al. [1983] described jitter in scatter plots that adds random noise to one or both of the variables using two sets of equally spaced values from -1 to 1 and utilizing fractions of the variable range to calculate the offset. Cleveland and McGill [1984] and later Cleveland [1993] described jittering as adding small random variables, in addition to moving points, transformations, open circles and sunflowers as approaches to address overlap.

A more advanced jittering algorithm may also keep track of each jittered point, minimizing possible new overlaps, since it is statistically possible for a random number generator to produce identical offsets $\Delta x$ and $\Delta y$ being generated for two or more overlapping points. The amount of jitter can follow a distribution, such as a normal distribution. However, if the number of records is greater than the number of unique locations within the predefined jittering surface of size $\Delta x_{range} \cdot \Delta y_{range}$, the jittering process unavoidably results in new overlaps.

Such handling of overlapping records is considered efficient, if the goal is strictly to show the number of records mapping to a particular area, but fails to provide an insight into the exact relationships among the instances that have been overplotted. The main weakness of such jitter technique is spatial displacement that is not driven by the data, causing difficulties in interpretation.

To emphasize the overlap effect, we created a simple data set by modifying the well-understood Fisher Iris flower data set (4 dimensions and flower type) [Fisher 1936] by rounding the fractional part of *sepal length* and *sepal width* dimensional values to the nearest value, with *petal length* and *petal width* dimensions left unchanged. The data set is interesting because in the original data set we cannot find a clear boundary between the three types of flowers when using two pairs of dimensions at a time. Petal width and petal length are very closely related dimensions with only limited correlation existing between those two dimensions and sepal length. Figure 1 is a parallel coordinate display [Inselberg and Dimsdale 1990] showing the modified set. The result is a reduced number of unique values, leading to multiple overlapping points.

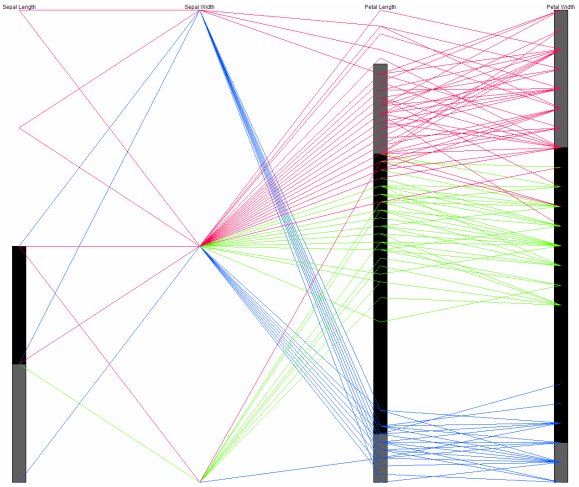


Figure 1: Parallel coordinates of the modified Iris flower data.

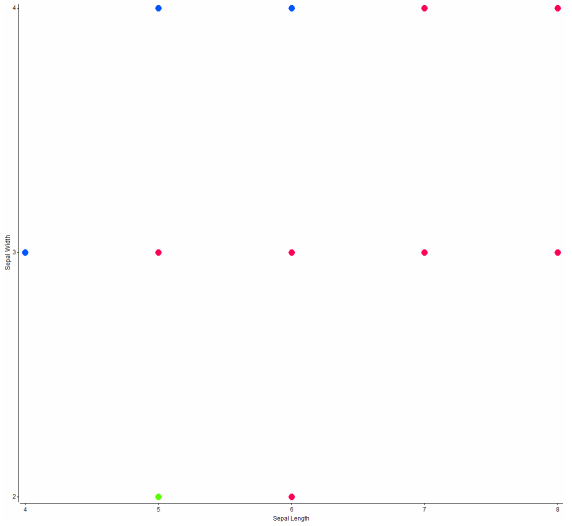


Figure 2: Scatter plot of Fisher Iris flower; sepal length and sepal width mapped to *x* and *y* coordinates. Occlusion results in only one type of flower being displayed at each location.

The classic scatter plot display in Figure 2 uses *sepal length* and *sepal with* as the *x* and *y* axes, respectively. The display contains only eleven points, representing 150 instances, with the remaining 139 instances overlapping at these eleven locations. The overlap is present because the sepal length dimension contains five unique values while the sepal width dimension contains three unique

Proceedings of the IEEE Symposium on Information Visualization 2003 (INFOVIS'03)
0-7695-2055-3/03 $ 17.00 © 2003 IEEE

values. This visualization provides little insight into the number of records at each of the locations, or the relationships between individual records. Additionally, the color (type of flower) of each of the points reflects only the last record that was plotted at a location. These records need to be spatially reorganized or jittered in order to address these problems. When the records are randomly jittered, the result displays all or most of the 150 records (Figure 3).

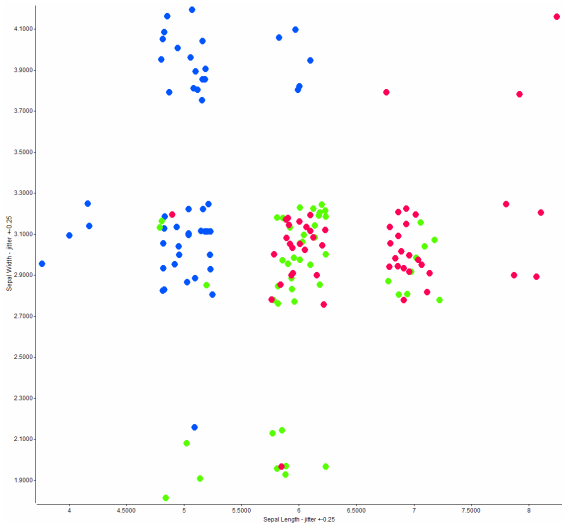

Figure 3: Overlapping records are randomly jittered to alleviate occlusion. Records jittered between -0.25 and 0.25.

## 3 Self-Organizing Map

The Self-Organizing Map (SOM) is an unsupervised, topology preserving neural network that maps a set of *n*-dimensional vectors to a two-dimensional topographic map [Kohonen 1982]. The training of an unsupervised neural network is data-driven, without a target condition that would have to be satisfied (as in a supervised neural network). A SOM combines an analytic and graphical technique to group data onto a low-dimensional display and organize the data into groups by this projection. The Kohonen SOM is similar to a k-means clustering algorithm, extending it by providing a topology-preserving mapping and placing similar objects in neighboring clusters.

The learning of the SOM is the process in which we form a nonlinear projection of the records onto a map. The self-organizing grid or map consists of an array of output nodes (neurons), each of them associated with an *n*-dimensional weight vector $m_i$ (corresponding to the *n* dimensions of the input data set). Initial values of $m_i$ may be randomly selected, preferably from the data set. Each record is positioned on the map, one by one, until the data set is exhausted. The assignment of weight vectors is formed in an unsupervised learning process, and the records are randomly drawn from the input distribution and presented to the network one at a time.

A record is mapped onto the SOM by calculating the similarity between the input vector and node *i*'s weight vector $m_i$. Each node *i* receives the same input vector and produces a single similarity value. The input record maps onto the best-matching (winning) node c, based on the largest similarity or the smallest distance (depending on the implementation). The weight vectors of nodes topographically close to the winning node (up to a certain geometric distance) adjust their weight vectors, "learning" something about the input. The adjustment depends on the size of the neighborhood, the value of the neighboring function and the learning function. This results in a local relaxation or smoothing of the neighborhood, which with continued learning leads to global ordering. This process is repeated until the output map converges to a stable or organized state when the average error falls below a pre-specified value or a certain number of iterations have been reached. The self-organizing process works by repeated refinement and progressively smaller values of the learning function.

Numerous SOM algorithms and extensions have been developed in a multitude of fields, which include engineering, military, and biomedical applications. Investigations include self-adaptive and incremental learning neural networks (SANN) that would replace the static topology networks [Nour and Madey 1996; Fritzke 1994], tree-structured SOM network architecture [Koikkalainen and Oja 1990], alternate neural-network based projections [Oja 1982; Kraaijveld et al. 1995; Merkl and Rauber 1997; Su and Chang 2000]. Some of these approaches aim to determine the shape and size of the self-organizing structure during the learning process and are targeted towards specific domains. Distances between neighboring output nodes on the final SOM map are also explored, such as in the Unified Distance Matrix (U-Matrix) described by Ultsch and Vetter [1994]. The U-Matrix displays weights at each cell with respect to its neighbors, and clusters and sparse areas are represented with light and dark colors, respectively. Most SOMs use a rectangular lattice display, although hexagonal and irregular examples are also used. For more details, please see [Kohonen 1995].

## 4 Smart Jitter

The goal of the Smart Jitter algorithm is to provide self-organization within classic visualizations. These may be linear or nonlinear projections of multi-dimensional data, such as scatter plot, polar charts, RadViz, and others. It is important to emphasize that Smart Jitter is not a technique that would determine the *x* and *y* coordinates for each input vector, but is rather a refinement method for spatial organization of overplotted input vectors. In general, our jittering approach can be applied to reduce occlusion given any record placement strategy (method to find the *x*, *y* position) on a two-dimensional display. We could use any dimension pair or mapping onto the *x* and *y* dimensions, or any other mapping onto a two-dimensional surface. The algorithm harnesses dimensional information of an input vector to provide local spatial organization while maintaining a relatively accurate *x* and *y* location on the low-dimensional surface of a matrix visualization.

The algorithm maps input vectors with similar properties to the same or neighboring output nodes of the display. Input vectors located in proximity of each other are *likely* to correlate more than vectors located farther apart. The correlation factor depends on the weight vectors of neighboring output nodes.

Displacement around an overplotted (*x*, *y*) position is driven by the dimensions of the overplotted data using a self-organizing map algorithm. The display surface is replaced with two grids; a secondary output grid within a primary output grid. The grid sizes (or resolution) are specifiable, or can be data-driven, based on the distribution and number of overplotted points, or on the overall number of displayed records. In Figure 4 we show how we grid the surface of a scatter plot, where the *x* and *y* dimensions correspond to dimensions 1 and 4 of the data set, respectively. Each primary output node contains a grid of secondary output

Proceedings of the IEEE Symposium on Information Visualization 2003 (INFOVIS'03)
0-7695-2055-3/03 $ 17.00 © 2003 IEEE

nodes, creating a grid within a grid. In this example each primary output node contains a secondary output grid of 25 nodes, or 5x5 secondary nodes. We map a record onto the primary output node $W_p$ as determined by the record's values of dimensions 1 and 4.

Mapping is first performed into the primary output nodes on the original projection of the data. Self-organization is repeated for every primary output node within the grid display. The secondary mapping first randomly initializes weight vectors of each secondary output node in the primary output node. The distance between an input vector and each output weight vector in this secondary grid is calculated, and the winning secondary output node $W_s$ is determined based on the smallest Euclidean distance.

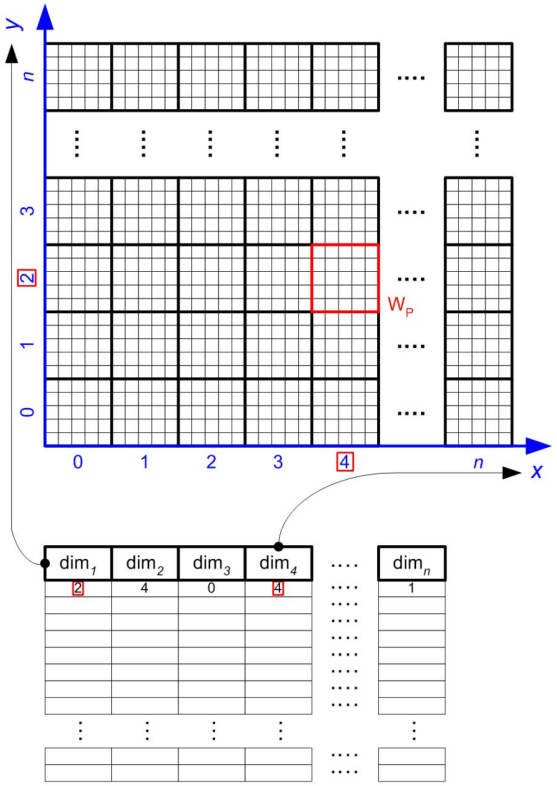


Figure 4: Secondary output grid within a primary output grid. $W_p$ is one primary output node.

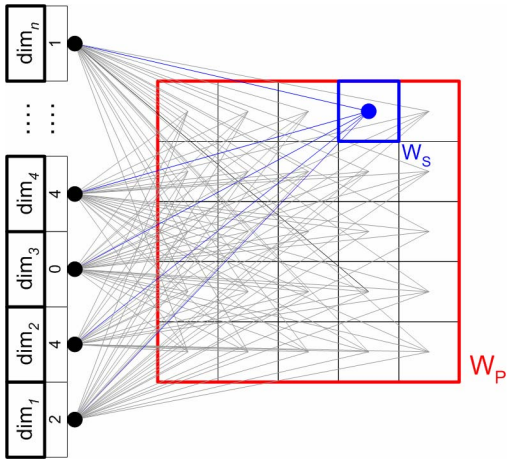


Figure 5: Mapping of a record into a secondary output node.

The record is mapped into that winning node $W_s$ (Figure 5) and the weight vector of the winning node adjusted, in addition to limited functional adjustment of the neighboring secondary weight vectors, depending on the neighborhood function. Adjustment of weights and self-organization is not limited to a single primary grid, but is rather driven by the neighborhood function and the properties of the records. This process repeats for every primary output node, in successive training passes through the input data set.

## 5 Application of Smart Jitter

To demonstrate our Smart Jitter algorithm, we use the modified Fisher Iris flower data set as described in Section 2. We let the output of a classic scatter plot self-organize as shown in Figure 6.

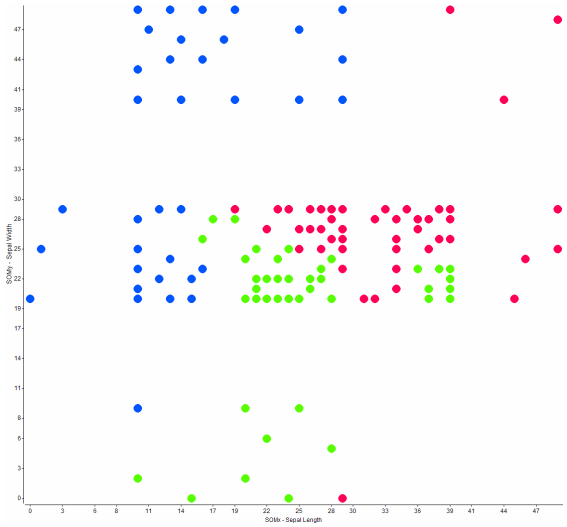

Figure 6: Smart Jitter plot of the modified Iris flower data set with 5x5 primary and 10x10 secondary self-organizing matrix.

Figure 6 shows a 5x5 primary and 10x10 secondary (5x5x10x10) Smart Jitter grid of the modified Iris data set. Iris *setosa* records are colored red, Iris *versicolor* green and Iris *virginica* blue. Each input vector is first placed on the primary grid, as determined by the value of sepal length and sepal width dimensions and boundaries of the primary grid of the range of values. Each primary output node contains a 10x10 secondary grid. These 100 secondary output nodes are initialized with random weight vectors which are adjusted with every input vector mapping. The records are mapped to the closest secondary output node as determined by the shortest Euclidean distance between an input vector and the weight vectors of the secondary output nodes within a single primary output node (in our case 100 secondary weight vectors). The algorithm first maps every input vector until the data set is depleted, and then repeats the training process until the target state is reached.

Figure 6 is extended with grid lines that mark the edges of output nodes as shown in Figure 7 displaying the Smart Jitter visualization with lines of separation among the secondary output nodes. These lines are a visual tool for identification of distances among neighboring output nodes and extend the U-Matrix approach by displaying the records mapped to output nodes combined with inter-nodal distance representations. This feature is a recommended extension for dense primary output grids with large secondary output grids, which take up most of the white space that is otherwise used as an indicator of grid edges. As the lines of separation indicate, it is possible that there is a large

Proceedings of the IEEE Symposium on Information Visualization 2003 (INFOVIS'03)
0-7695-2055-3/03 $ 17.00 © 2003 IEEE

difference between two neighboring secondary output nodes within the same primary output node.

Figure 7: Distance between neighboring output nodes as indicated by lines of separation.

An example of close and distant records within the same primary output node is provided in Figure 8. We selected records of all three flower types. We were guided by the lines of separation and selected one *setosa*, three *versicolor* and additional *virginica* records. Figure 9 provides a supplemental view of these records using a parallel coordinate display. Parallel coordinates show that these records differ only on petal length and petal width dimensions. Their values on the sepal length and sepal width dimension are the same, thus placing them in the same primary output node. Smart Jitter self-organization separated most of the *virginica* (blue) records from the other four records. Two records within the same primary output node are not necessarily more similar than two records that belong to two different primary output nodes. The *setosa* record in red is more similar to the records in the neighboring primary output node (in grey) than to other records in its primary output node. We also demonstrate that pairs of instances spatially equally distant on the output grid are not necessarily equally similar. The lines of separation show that the weight vectors of secondary output nodes of *virginica* (blue) records are much more similar than the weight vectors of secondary output nodes with *versicolor* and *setosa* records, as confirmed by parallel coordinates (Figure 9).

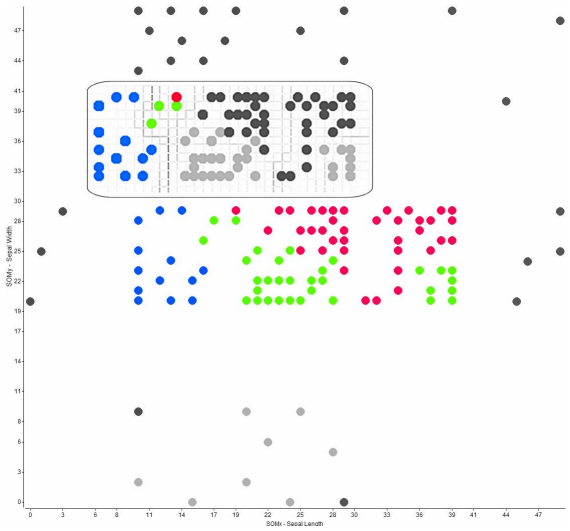

Figure 8: Records in this example were selected based on lines of separation.

We expected that an increase in the size of the primary and secondary output nodes would result in a display with larger separation of less dense areas. Figure 10 shows a primary output grid with 100 nodes, each containing 25 secondary output nodes. The result is eleven populated primary output nodes with instances spatially ordered within the secondary output grid. Higher density indicates larger number of records mapping to the same area while the location of each record indicates the distance between the records. Primary output nodes with highly correlated instances do not contain lines of separation (these lines are white or very light gray). Primary output nodes whose records are less correlated would be spatially dispersed with darker lines of separation.

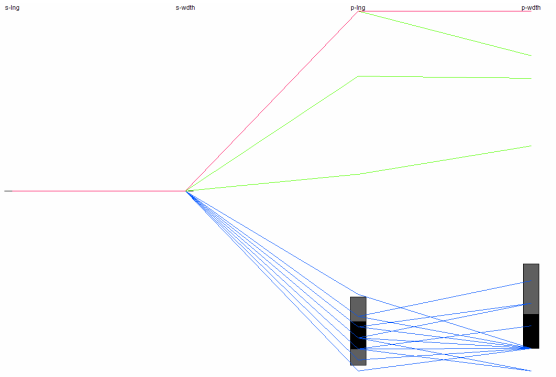


Figure 9: Parallel coordinates of selected records.

Primary output nodes without any input vectors mapped to them have a large number of very dark lines of separation, showcasing the random initialization of these vectors without adjustment according to the SOM neighborhood equation. If we increase the number of secondary output nodes, we would provide for more detailed separation of input vectors that map within the same primary output node. This would be beneficial in densely populated areas to minimize the overlap, and provide for clearer relationships among the records.

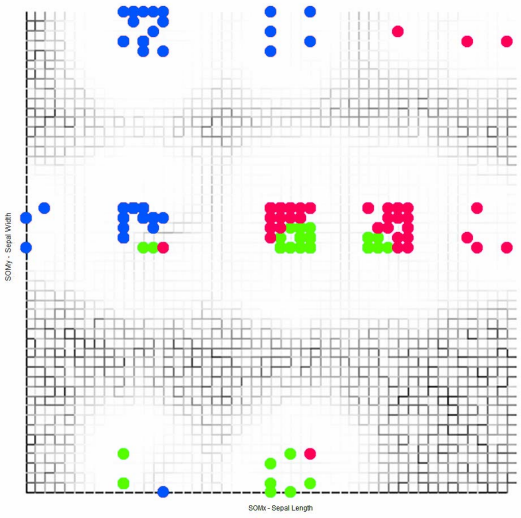

Figure 10: Large Smart Jitter of the modified data with 10x10 primary and 5x5 secondary grid.

## 6 Conclusion

Smart Jitter provides an increased visual scalability and offers higher intrinsic dimension than the classic visualization, as showcased on the scatter plot. It provides a novel approach for intelligent spatial ordering of overlapped records, based on modified Kohonen SOM algorithm. The resulting mapping more efficiently alleviates crowding and overlaps, and emphasizes the relationships among the neighboring multi-dimensional records. The user can approach perceptual ambiguities associated with occlusion and gain insight into multi-dimensional data sets in one visualization.

Proceedings of the IEEE Symposium on Information Visualization 2003 (INFOVIS'03)
0-7695-2055-3/03 $ 17.00 © 2003 IEEE

The best spatial ($x$, $y$) ordering of input vectors is achieved through the use of Smart Jitter with a large number of primary nodes containing a large number of secondary output nodes. These secondary output nodes provide a large jittering surface, accommodating a large number of overlapped instances. The size of the Smart Jitter output grid (primary and secondary nodes) is limited by the display capabilities of the output device. Each of the output nodes has to be at least one pixel large, in order to accommodate a single pixel representing an input vector.

## 7 Future Directions

Evaluations in progress include:

- The integration of Smart Jitter with other visualizations
- Developing an interactive Smart Jitter lens to provide for the exploration of data within visualizations
- Efficiently determining the size of the output
- Developing adaptive and non-matrix space divisions (Voronoi diagram, circular and octagonal)

## 8 References


CHAMBERS, J. M., CLEVELAND, W. S., KLEINER, B., AND TUKEY, P. A. 1983. *Graphical Methods for Data Analysis*, Wadsworth.

CHUAH, M., ROTH, S., MATTIS, J., AND KOLOJEJCHICK, J. 1995. SDM: Selective Dynamic Manipulation of Visualizations, In *Proceedings of ACM Symposium on User Interface Software and Technology*, 61-70.

CLEVELAND, W. S., AND MCGILL, R. 1984. Graphical Perception: Theory, Experimentation and Application to the Development of Graphical Methods, *Journal of the American Statistical Association 79*, 387, 531-554.

CLEVELAND, W. S. 1993. *Visualizing Data,* Hobart Press, Summit, NJ.

EICK, S. G. 2000. Visual Discovery and Analysis, In *IEEE Transactions on Visualization and Computer Graphics 6*, 1, 44-58.

FEKETE, J.-D., AND PLAISANT, C. 1999. Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualization, In *Proceedings of CHI'99*, ACM, New York, 512-519.

FISHER, R. A. 1936. The Use of Multiple Measurements on Taxonomic Problems, *Annals of Eugenics 7*, 179-188.

FRITZKE, B. 1994. Growing Cell Structures - a Self-Organizing Network for Unsupervised and Supervised Learning, *Neural Networks 7*, 9, 1441-1460.

GRINSTEIN, G. G., TRUTSCHL, M., AND CVEK, U. 2001. High-Dimensional Visualizations, In *Proceedings of the Visual Data Mining Workshop KDD'01*, ACM, New York.

HOFFMAN, P., GRINSTEIN, G. G., AND PINKNEY, D. 1999. Dimensional Anchors: a graphic primitive for multidimensional multivariate information visualizations, In *Proceedings of the Workshop on New Paradigms in Information Visualization and Manipulation (NPIV),* ACM Press, 9-16.

INSELBERG, A., AND DIMSDALE, B. 1990. Parallel coordinates: a Tool for Visualizing Multidimensional Geometry, In *Proceedings of IEEE Visualization 1990*, IEEE Computer Society Press, 361-378.

KANDOGAN, E. 2000. Star Coordinates: A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions, *Late-Breaking Hot Topics of IEEE Information Visualization 2000*, IEEE Computer Society Press, 4-8.

KOHONEN, T. 1982. Self-Organized Formation of Topologically Correct Feature Maps, *Biological Cybernetics 43*, 59-69.

KOHONEN, T. 1995. *Self-Organizing Maps,* Springer Series in Information Sciences 30.

KOIKKALAINEN, P., AND OJA, E. 1990. Self-Organizing Hierarchical Feature Maps, In *International Joint Conference on Neural Networks IJCNN'90*, 279-284.

KRAAIJVELD, M. A., MAO, J., AND JAIN, A. K. 1995. A Nonlinear Projection Method Based on Kohonen's Topology Preserving Maps, *IEEE Transactions on Neural Networks 6*, 3, 548-559.

MANSON, J. 1999. *Occlusion in Two-Dimensional Displays: Visualization of Meta-Data,* University of Maryland, College Park.

MERKL, D., AND RAUBER, A. 1997. Alternative Ways for Cluster Visualization in Self-Organizing Maps, In *Proceedings of the Workshop on Self-Organizing Maps WSOM'97,* Helsinki University of Technology, Finland, 106-111.

NOUR, M. A., AND MADEY, G. R. 1996. Heuristic and Optimization Approaches to Extending the Kohonen Self-Organizing Algorithm, *European Journal of Operational Research 93*, 2, 428-448.

OJA, E. 1982. A Simplified Neuron Model as a Principle Component Analyzer, *Journal of Mathematical Biology 15*, 267-273.

SU, M.-C., AND CHANG, H.-T. 2000. Fast Self-Organizing Feature Map Algorithm, *IEEE Transaction on Neural Networks 11*, 3, 721-733.

ULTSCH, A., AND VETTER, C. 1994. Self-Organizing-Feature-Maps Versus Statistical Clustering: A Benchmark, *Technical Report 9*, Department of Mathematics, University of Marburg, Marburg, Germany.

WONG, P. C., AND BERGERON, R. D. 1997. 30 Years of Multidimensional Multivariate Visualization, In *Scientific Visualization: Overviews, Methodologies & Techniques*, IEEE Computer Society Press, Los Alamitos, 3-33.

ZHAI, S., BUXTON, W., AND MILGRAM, P. 1996. The Partial-Occlusion Effect: Utilizing Semitransparency in 3D Human-Computer Interaction, *ACM Transactions on Computer-Human Interaction 3*, 3, 254-284.



Proceedings of the IEEE Symposium on Information Visualization 2003 (INFOVIS'03)
0-7695-2055-3/03 $ 17.00 © 2003 IEEE