XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data

Matthew O. Ward
Computer Science Department
Worcester Polytechnic Institute
Worcester, MA 01609

Abstract

Much of the attention in visualization research has focussed on data rooted in physical phenomena, which is generally limited to three or four dimensions. However, many sources of data do not share this dimensional restriction. A critical problem in the analysis of such data is providing researchers with tools to gain insights into characteristics of the data, such as anomalies and patterns. Several visualization methods have been developed to address this problem, and each has its strengths and weaknesses. This paper describes a system named *XmdvTool* which integrates several of the most common methods for projecting multivariate data onto a two-dimensional screen. This integration allows users to explore their data in a variety of formats with ease. A view enhancement mechanism called an *N-dimensional brush* is also described. The brush allows users to gain insights into spatial relationships over N dimensions by highlighting data which falls within a user-specified subspace.

1 Introduction

The major objectives of data analysis are to summarize and interpret a data set, describing the contents and exposing important features [6]. Visualization can play an important role in each of these objectives, both in qualitative evaluation of the data and in conjunction with focussed quantitative analysis. A given visualization technique is generally applicable to data of certain characteristics. This paper describes a system which has been developed for the display of *multivariate* data.

Multivariate data can be defined as a set of entities $\mathbf{E}$, where the i^{th} element e_i consists of a vector with $\mathbf{n}$ observations, $(x_{i1}, x_{i2}, ..., x_{in})$. Each observation (variable) may be independent of or interdependent with one or more of the other observations. Variables may be discrete or continuous in nature, or take on symbolic (nominal) values. Variables also have a scale associated with them, where scales are defined according to the existence or lack of an ordering relationship, a distance (interval) metric, and an absolute zero (origin).

When visualizing multivariate data, each variable may map to some graphical entity or attribute. In doing so, the type (discrete, continuous, nominal) or scale may be changed to facilitate display. In such situations, care must be taken, as a graphical variable with a perceived characteristic (type or scale) which is mapped to a data variable with a different characteristic can lead to misinterpretation.

Many criteria can be used to gauge the effectiveness of a visualization technique for multivariate data. Some of these are directly measurable, such as the number of variables or data points which can be displayed. Others require subjective evaluation and are thus difficult to quantify. The list below summarizes some of these criteria. In our studies of the various projection techniques we are examining these and other issues.

Constraints on dimensions: all of the projection techniques surveyed degrade in usefulness when the number of dimensions or variables exceeds a certain size.

Constraints on data set size: each data projection method allocates a certain amount of screen space for each data sample. As screen space is finite, there exists a limit for effective visualization.

The effect of data distribution: sparse data sets may lead to poor screen utilization, while highly clustered data may make it difficult to identify individual samples.

Occlusion: in many instances, different data points will map to the same location on the screen. It is important that the viewer be aware of these overlaps and perhaps have a strategy to obtain a view which avoids a given overlap.

Perceptibility: the goal of visualizing data is to try to understand the structure of the data or detect some data characteristics, such as anomalies, extrema, and patterns. These features may be more readily apparent in some projection techniques than others.

User interactions: visualization is incomplete without interaction. Each projection technique has a logical set of interactive capabilities for view modification and enhancement.

Interpretation guides: users need reference points such as keys, labels, and grids to help interpret the data and determine its context.

Use of color: color can be used to convey one or more variables of the data or to help highlight or deemphasize subsets of the data. As color perception varies both contextually and between individuals, it should be used with care.

3-Dimensional cues: other cues commonly found in 3-D graphics, such as shading, translucency, and motion, can be used to reduce the overall dimensionality, although often at some cost in interpretability.

2 N-Dimensional Data Visualization Methods

Many techniques for projecting N-dimensional data onto two dimensions have been proposed and explored over the years. This section presents an overview of four classes of techniques, and describes their implementation within XmdvTool.

2.1 Scatterplots

Scatterplots are one of the oldest and most commonly used methods to project high dimensional data to 2-dimensions. In this method, $N*(N-1)/2$ pairwise parallel projections are generated, each giving the viewer a general impression regarding relationships within the data between pairs of dimensions. The projections are generally arranged in a grid structure to help the user remember the dimensions associated with each projection. Many variations on the scatterplot have been developed to increase the information content of the image as well as provide tools to facilitate data exploration. Some of these include rotating the data cloud [12], using different symbols to distinguish classes of data and occurrences of overlapping points, and using color or shading to provide a third dimension within each projection.

The procedure for generating scatterplots within XmdvTool is quite straightforward. The display window is divided into an N by N grid, and each data point results in N^2 points being drawn, using only two dimensions per view. Columns and rows in the grid are labeled according to the dimension they represent.

Figure 1 presents a seven dimensional data set using scatterplots. Note that plotting each dimension against itself along the diagonal provides distribution information on the individual dimensions. The data set contains statistics regarding crime in Detroit between 1961 and 1973, and consists of 13 data points. The data set was obtained via anonymous ftp from unix.hensa.ac.uk in the directory /pub/statlib/datasets. Some dimensions of the original set have been eliminated to facilitate display using scatterplots. The dimensions and their ranges are given in Table 1. Linear structures within several of the projections indicate some correlation between the two dimensions involved in the projections. Thus, for example, there is a correlation between the number of full-time police, the number of homicides, and the number of government workers (with a corresponding negative correlation in the percent of cleared homicides).

One major limitation of scatterplots is that they are most effective with small numbers of dimensions, as increasing the dimensionality results in decreasing the screen space provided for each projection. Strategies for addressing this limitation include using three dimensions per plot or providing panning or zooming mechanisms. Other limitations include being generally restricted to orthogonal views and difficulties in discovering relationships which span more than two dimensions. Advantages of scatterplots include ease of interpretation and relative insensitivity to the size of the data set.

2.2 Glyphs

The definition of a *glyph* covers a large number of techniques which map data values to various geometric and color attributes of graphical primitives or symbols [10]. Some of the many glyph representations proposed over the years include the following:

Dimension	Minimum	Maximum
Full-time police per 100,000 population	255.	400.
Unemployment rate	0.	12.
Number of manufacturing workers in thousands	450.	620.
Number of handgun licenses per 100,000	100.	1200.
Number of government workers in thousands	120.	250.
Percent homicides cleared by arrests	50.	100.
Number of homicides per 100,000	0.	60.

Table 1: Dimensions of the Detroit data set.

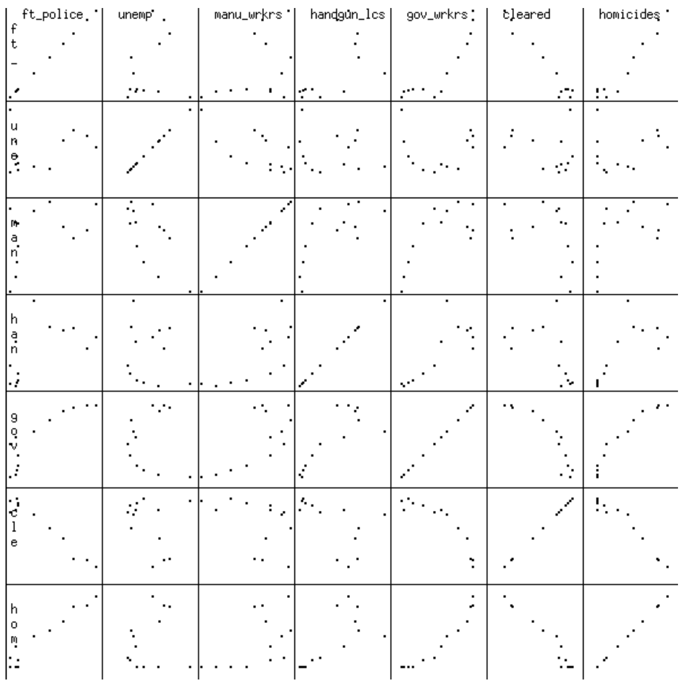

Figure 1: The Detroit data using Scatterplots. Correlation between pairs of dimensions manifest themselves as linear structures.

- Faces, where attributes such as location, shape, and size of features such as eyes, mouth, and ears are controlled by different data dimensions [5].

- Andrews glyphs, which map data to functions (e.g. trigonometric) of N variables [1].

- Stars or circle diagrams, where each glyph consists of N lines emanating from a point at uniformly separated angles with lengths determined by the values of each dimension, with the endpoints connected to form a polygon [13].

- Stick figure icons, where the length, orientation, and color of N elements of a stick figure are controlled by the dimensional values [7].

- Shape coding, where each data point is represented by a rectangle which has been decomposed into N cells and the dimensional value controls the color of each cell [4].

In XmdvTool, we use the star glyph pattern [13]. The user can choose between either uniformly spaced glyphs or using two of the dimensions to determine the location of the glyph within the window. Each ray of the glyph has a minimum and maximum length, determined either by the user (for glyphs with data-driven locations) or by the size of the view area (for uniformly spaced glyphs). A key for interpreting the dimensions is included in a separate window.

Figure 2 shows an example of glyphs in XmdvTool using the same data set as in Figure 1. The evolution of the shape over time indicates both trends and anomalies. For example, the clear protrusion in the direction associated with cleared homicides (257 degrees) found in the earlier shapes evolves into a concavity over time.

Glyph techniques are generally limited in the number of data elements which can be displayed simultaneously, as each may require a significant amount of screen space to be viewed. The density and size constraints of the elements, however, depend on the level of perceptual accuracy required. Also, it can be difficult to compare glyphs which are separated in space, although if data dimensions are not being used to determine glyph locations, the glyphs can be sorted or interactively clustered on the screen to help highlight similarities and differences. Most of the glyph techniques are fairly flexible as to the number of dimensions which can be handled, though discriminability may be affected for large values of N (greater than 20 or so).

2.3 Parallel Coordinates

Parallel coordinates is a technique pioneered in the 1970's which has been applied to a diverse set of multidimensional problems [8]. In this method, each dimension corresponds to an axis, and the N axes are organized as uniformly spaced vertical lines. A data element in N-dimensional space manifests itself as a connected set of points, one on each axis. Points lying on a common line or plane create readily perceived structures in the image.

In generating the display of parallel coordinates in XmdvTool, the view area is divided into N vertical slices of equal width. At the center of each slice an axis is drawn, along with a label at the top end. Data points are generated as polylines across the N axes.

Figure 3 shows an example of the Parallel Coordinates technique using the same data set as in Figure 1. Clustering is evident among some of the lines, indicating a degree of correlation. For example, the X-shaped structure between the axes for cleared cases and homicides indicates an inverse correlation, and the nearly parallel lines betwen the axes for manufacturing workers and handgun licenses suggests a relatively constant increase in the rate of handgun ownership as manufacturing jobs increase (some exceptions exist, however).

The major limitation of the Parallel Coordinates technique is that large data sets can cause difficulty in interpretation; as each point generates a line, lots of points can lead to rapid clutter. Also, relationships between adjacent dimensions are easier to perceive than between non-adjacent dimensions. The number of dimensions which can be visualized is fairly large, limited by the horizontal resolution of the screen, although as the axes get closer to each other it becomes more difficult to perceive structure or clusters.

2.4 Hierarchical Techniques

Several recent techniques have emerged which involve projecting high dimensional data by embedding dimensions within other dimensions. In the 1-D case [11], one starts by discretizing the ranges of each dimension and assigning an ordering to the dimensions (dimensions are said to have unique "speeds"). A background color is also associated with each speed. The next step is to divide the screen into C_0 vertical strips, where C_0 is the cardinality of the dimension with the slowest speed. The strips are colored according to that speed. Each of these strips is then divided into C_1 strips and colored accordingly. This is repeated until all dimensions have been embedded and the data value associated with each cell can be plotted on the vertical axis.

In 2-D [9], an analogous technique called *Dimensional Stacking* involves recursively embedding images defined by a pair of dimensions within pixels of a higher-level image. Unlike the previous system, however, data is not restricted to functions, thus making this technique amenable to a wider range of data types. In *Worlds within Worlds* [2], each location in a 3-D space may in turn contain a 3-D space which the user may investigate in a hierarchical fashion. The most detailed level may contain surfaces, solids, or point data.

XmdvTool requires three types of information to project data using dimensional stacking. The first is the cardinality (number of buckets) for each dimension. The range of values for each dimension is then decomposed into that many equal sized subranges. The second type of information needed is the ordering for the dimensions, from outer-most (slowest) to inner-most (fastest). Dimensions are assumed to alternate in orientation. The last piece of information used is the minimum size for the plotted data item (the system will increase this value if the entire image can fit within the view area). Each data point then maps into a unique bucket, which in turn maps to a unique location in the resulting image. If the image generated exceeds the size of the view area, scroll bars are automatically generated to allow panning. A key is provided in a separate window to help users understand the order of embedding, and grid lines of varying intensity provide assistance in interpreting transitions between buckets at different levels in the hierarchy.

The sparseness of the data set of Figure 1 makes uncovering relationships difficult using Dimensional Stacking. Figure 4 shows a denser set consisting of 3-D drill hole data with a fourth dimension representing the ore grade found at the location (more than 8000 data points). Longitude and latitude are mapped to the outer dimensions, each with cardinality 10. Depth and ore grade map to the inner dimensions (ore grade is the vertical orientation), with cardinality 10 and 5, respectively. There is a clear region in which the ore grade improves with depth, and other places where digging had stopped prior to the ore grade falling significantly. By adjusting the cardinalities and ranges for the various dimensions, a more detailed view of the data may be obtained [16].

The hierarchical techniques are best suited for fairly dense data sets and do rather poorly with sparse data. This is due to the fact that each possible data point is allocated a specific screen location (with overlaps avoidable by careful discretization of dimensions), and as the dimension of the data increases, the screen space needed expands rapidly. In contrast, the techniques described earlier generally do well with sparse data over high numbers of dimensions, though scatterplots are constrained somewhat it the maximum manageable dimension. The major problem with hierarchical methods is the difficulty in determining spatial relationships between points in non-adjacent dimensions. Two points which in fact are quite close in N-space may project to screen locations which are quite far apart. This is somewhat alleviated by providing users with the ability to rapidly change the nesting characteristics and discretization of the dimensions.

3 N-Dimensional Brushing

Another useful capability of XmdvTool is N-dimensional brushing [15]. **Brushing** is a process in which a user can highlight, select, or delete a subset of elements being graphically displayed by pointing at the elements with a mouse or other suitable input device. In situations where multiple views of the data are being shown simultaneously (e.g. scatterplots), brushing is often associated with a process known as **Linking**, in which brushing elements in one view affects the same data in all other views. Brushing has been employed as a method for assisting data analysis for many years. One of the first brushing techniques was applied to high dimensional scatterplots [3]. In this system, the user specified a rectangular region in one of the 2-D scatterplot projections, and based on the mode of operation, points in other views corresponding to those falling within the brush were highlighted, deleted, or labeled. Brushing has also been used to help users select data points for which they desire further information. Smith et. al. [14] used brushing of images generated by stick figure icons to obtain higher dimensional information through sonification for the selected data points.

In XmdvTool, the notion of brushing has been extended to permit brushes to have dimensionality greater than two. The goal is to allow the user to gain some understanding of spatial relationships in N-space by highlighting all data points which fall within a user-defined, relocatable subspace. N-D brushes have the following characteristics:

Brush Shape: In XmdvTool, the shape of the brush is that of an N-D hyperbox. Other generic shapes, such as hyperellipses, will be added in the future, as well as customized shapes, which can consist of any connected arbitrary N-D subspace.

Brush Size: For generic shapes the user simply needs to specify N brush dimensions. The mechanism used by XmdvTool to perform this, albeit primitive, is to use N slider bars.

Brush Boundary: In XmdvTool, the boundary of a brush is a step edge. Another possibility would be a ramp, with many possibilities for the shape of the ramp. Another interesting enhancement could be achieved by coloring data points according to the degree of brush coverage (where it falls along the ramp).

Brush Positioning: Brushes have a position which the user must be able to easily and intuitively control. In the general case, the user needs to specify N values to uniquely position the brush. This is done in XmdvTool via the same sliders employed in size specification.

Brush Motion: Although XmdvTool currently supports only manual brush motion, we hope to implement several forms of brush path specification in the future.

Brush Display: N-dimensional space is usually quite sparse, thus it is useful at times to display the subspace covered by the brush on the data display. The location can be indicated either by the brush's boundary or a shaded region showing the area of coverage. In XmdvTool, brushes are displayed as shaded blue-grey regions, with data points which fall within the brush highlighted in red.

Brush size and position are currently specified in a rather simplistic manner. The user selects the dimension to be adjusted, and then changes the brush size or position via a slider. There are many opportunities for allowing the user to directly manipulate the brush in the display area, although each procedure would need to be customized based on the projection method in use. For example, the user could move or resize one dimension of the brush by dragging the edge or center of the brush along one of the axes of the Parallel Coordinate display, or set the location of the brush by selecting one of the glyphs. Direct manipulation of the brush will be one of the features incorporated into future releases of XmdvTool.

4 Summary and Conclusions

This paper has presented an overview of the field of multivariate data visualization and has introduced a software package, named XmdvTool, to help users experiment with different N-dimensional projection techniques. Each of the techniques has its strengths and weaknesses in regards to the types of data sets for which it is most appropriate. The number of dimensions in the data as well as the range of values and distribution within each dimension all play important roles. The goal of the user in examining the data, whether it be for patterns, anomalies, or dependencies, is also important in gauging the relative usefulness of a technique. One of the long-term goals of this research effort is to create a benchmark for evaluating multivariate data visualization tools using data sets with a diversity of characteristics. The evaluation criteria listed in Section 2 (as well as other criteria) will be employed in the assessment process, along with studying the performance of human subjects in locating structure in data sets using different tools.

Future development of XmdvTool will include both generic view-enhancement techniques (additional brush capabilities, panning, zooming, clipping) and methods related to the specific projection techniques. Some of these will include experimenting with different glyph structures, interactively changing the order of dimensions, and dynamic control over the binning for dimensional stacking. Some of these capabilities have already been implemented into N-Land [16], a software package for exploring the capabilities of dimensional stacking developed by the author and his colleagues, and thus should be relatively easy to incorporate.

XmdvTool is written in C using X11R5, Athena Widgets, and the Widget Creation Library (Wcl-2.5), and will be made available on anonymous ftp (wpi.wpi.edu) in the near future. Interested parties should contact the author at matt@cs.wpi.edu or check in the /contrib/Xstuff directory at the above mentioned site for the file named XmdvTool.tar.Z.

References

[1] Andrews, D.F., "Plots of high dimensional data", *Biometrics*, Vol. 28, pp. 125-136, 1972.

[2] Beshers, C., Feiner, S., "AutoVisual: rule-based design of interactive multivariate visualizations," *IEEE Computer Graphics and Applications*, Vol. 13, No. 4, pp. 41-49, 1993.

[3] Becker, R.A., Cleveland, W.S., "Brushing Scatterplots," from *Dynamic Graphics for Statistics* (eds. W.S. Cleveland and M.E. McGill), Wadsworth, Inc., Belmont, CA, 1988.

[4] Beddow, J., "Shape coding of multidimensional data on a microcomputer display," *Proceedings of Visualization '90*, pp. 238 - 246, 1990.

[5] Chernoff, H., "The use of faces to represent points in k-dimensional space graphically," *Journal of the American Statistical Association*, Vol. 68, pp. 361-368, 1973.

[6] Everitt, B.S., *Graphical Techniques for Multivariate Data*, Heinemann Educational Books, Ltd., London, 1978.

[7] Grinstein, G., Pickett, R., Williams, M.G., "EXVIS: an exploratory visualization environment," *Graphics Interface '89*, 1989.

[8] Inselberg, A., Dimsdale, B., "Parallel coordinates: a tool for visualizing multidimensional geometry," *Proceedings of Visualization '90*, pp. 361 - 378, 1990.

[9] LeBlanc, J., Ward, M.O., Wittels, N., "Exploring N-dimensional databases," *Proceedings of Visualization '90*, pp. 230 - 237, 1990.

[10] Littlefield, R.J., "Using the GLYPH concept to create user-definable display formats," *Proc. NCGA '83*, pp. 697-706, 1983.

[11] Mihalisin, T., Gawlinski, E., Timlin, J., and Schwegler, J., "Visualizing multivariate functions, data, and distributions," *IEEE Computer Graphics and Applications*, Vol. 11, pp. 28 - 37, 1991.

[12] Tukey, J.W., Fisherkeller, M.S., Friedman, J.H., "PRIM-9, an interactive multidimensional data display and analysis system," in *Dynamic Graphics for Statistics (W.S. Cleveland and M.E. McGill, eds.)*, Wadsworth and Brooks, 1988.

[13] Siegel, J.H., Farrell, E.J., Goldwyn, R.M., Friedman, H.P., "The surgical implication of physiologic patterns in myocardial infarction shock," *Surgery*, Vol. 72, pp. 126-141, 1972.

[14] Smith, S., Bergeron, R.D., Grinstein, G., "Stereophonic and surface sound generation for exploratory data analysis," Proc. CHI '90: Human Factors in Computer Systems, pp. 125-132, 1990.

[15] Ward, M.O., "N-dimensional brushes: gaining insights into relationships in N-D data," submitted for publication, 1993.

[16] Ward, M.O., LeBlanc, J.T., Tipnis, R., "N-Land: a graphical tool for exploring N-dimensional data," to be published in *Proceedings of CG International '94*, 1994.

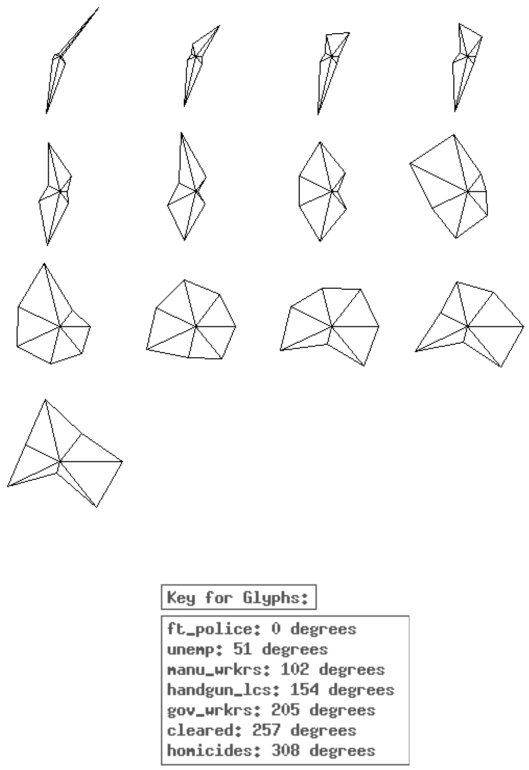

Figure 2: The Detroit data using the Star glyph representation. Key provides associations of dimensions with line orientation.

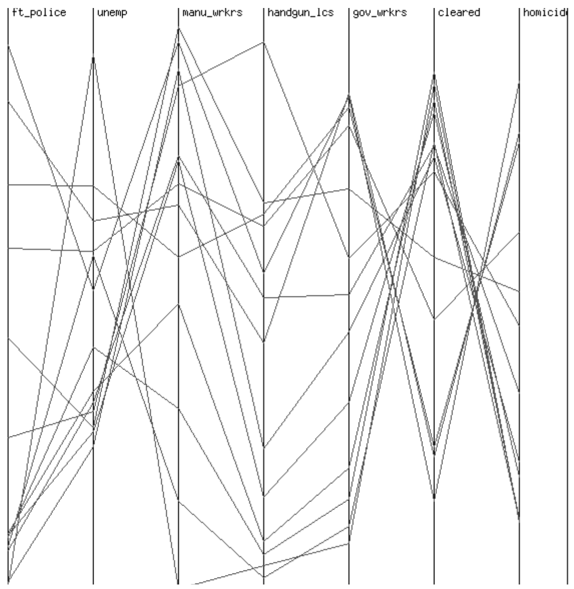

Figure 3: The Detroit data using the Parallel Coordinates representation. Inverse correlations can be seen between the number of government workers versus the percent of cleared homicides, as well as between the percent of cleared homicides versus the number of homicides.

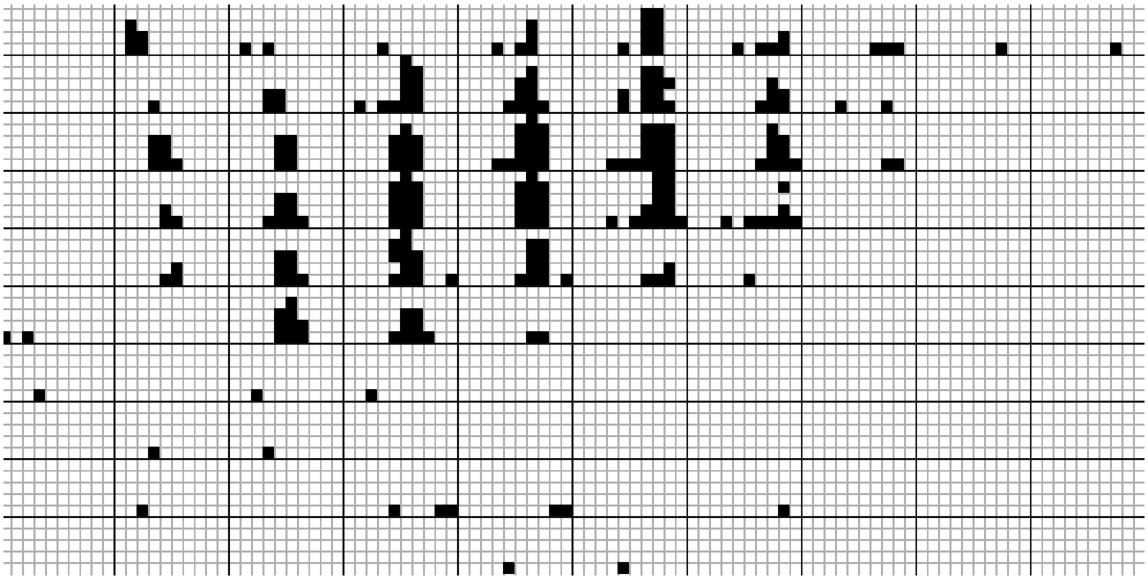

Figure 4: Four-dimensional data set using dimensional stacking. The data consists of ore grades with three spatial dimensions. Inner dimensions show ore grade and depth. Outer dimensions show longitude and latitude. Highest levels of ore grade are seen in the third to fifth sections horizontally and the sixth to eighth sections vertically.