Hypervariate Data Visualization

Bartholomaeus Steinmayr

Abstract— Both scientists and normal users face enormous amounts of data, which might be useless if no insight is gained from it. To achieve this, visualization techniques can be used. Many datasets have a dimensionality higher than three. Such data is called "hypervariate" and cannot be visualized directly in the three-dimensional space that we inhabit. Therefore, a wide variety of specialized techniques have been created for rendering hypervariate data. These techniques are based on very different principles and are designed for very different areas of application. This paper gives an overview of six representative techniques. For most techniques a rendering of a common dataset is provided to allow an easier comparison. Furthermore, an evaluation of the strengths and weaknesses of each technique is given. As an outlook, two papers dealing with quantitative analysis of visualization methods are presented.

Index Terms—hypervariate, information visualization, hypervariate data visualization, overview

1 Introduction

Modern information systems and embedded computers create an ever increasing stream of data. For example, Fleming [8] lists over a hundred different sensor types used in automotive applications today. The data from these sensors might help engineers to find defects in a car in an early stage. However, without effective ways to explore and gain knowledge from this data, "the databases become datadumps" [1].

This is where information visualization comes into play: Its main goal is to help the user gain insight into a given dataset [18]. Visualization has been applied for centuries, for example in the form of cartography. Over this time a variety of different methods has been created, catering to different application fields. In many of these fields fields, so called "hypervariate data" needs to be visualizated, requiring special techniques. In this paper, I will explore the visualization of hypervariate data and discuss some representative methods for visualizing hypervariate data.

2 Hypervariate data

In this paper, data will be considered as discrete samples of some source, which can consist of different entities, different periods of time or a combination of both [6]. In all cases, the result is a set of values for each datapoint. The cardinality of this set depends on the number of variables sampled. These values can be discrete or continuous. However, discrete values can still be mapped to real numbers. Therefore, a dataset with n samples of p variables can be considered as a set of n points in p-dimensional space $\mathbb{R}^p$ [2].

In case $p \leq 3$ the data can be directly visualized in a three-dimensional universe (the reasons why our universe is inherently three-dimensional can be found in [5]). However, in many cases $p \gg 3$ (see section 3). For this kind of data specialized visualization methods are necessary. To overcome the limited dimensionality of the space we inhabit, these methods commonly use special rendering methods and the interactive capabilities of modern computers. The main part of this paper focuses on an overview of different rendering techniques for hypervariate data, however, one interactive technique (4.2.4) will be discussed for completeness.

3 Application domains for hypervariate data visualization

As aforementioned, hyperdimensional data occurs in a multitude of instances. In the following section, I will explore a representative selection of application domains for hypervariate information visualization.

3.1 Everyday decision making

The dataset illustrating this example was provided by Ramos and Donoho [17]. It consists of 8 attributes (among others mileage, horsepower and country of origin) of 406 cars produced from 1970 till 1982.

Hypervariate data visualization is not restricted to scientific applications. Making informed decisions is one of the challenges of everyday life. Take for example the purchase of a car. The buyer faces a large number of possible choices. Each car is characterized by a number of attributes, most of which can be quantified. The intended application of the vehicle and the financial possibilies of the buyer usually dictate a tradeoff between these attribues. In this situation, information visualization methods can help to make that tradeoff.

3.2 Commercial data analysis

The example dataset for this scenario was presented by Kandogan [12]. It contains 21 attributes of customers of telecommunications company (among others number of calls to customer service, domestic and international charges and whether the user has quit the service).

Most companies, but especially those in the telecommunication sector, can collect data about their customers. If this data is properly understood, it can be used to optimize the business model. Kandogan proposed [12] that telecommunications companies need to understand why customers quit their service. If this is accomplished, customers with similar patterns can be identified and possibly retained through the use of various marketing methods.

3.3 Sociopolitical studies

A small dataset is presented by Kleiner et al. [14]. It contains the percentage of votes for republican party from six souther US states in six different election years. This relatively small dataset is used as an example because of the known behaviour of the states. However data like this also exists for all states and all election years. Analysis of these large datasets can be used to explain trends and shifts in the political climate of a country.

4 Representative techniques

In the following two sections a brief review of the representative techniques for hypervariate information visualization is given. These techniques have been selected with the goal of showing a wide spectrum of graphical approaches in mind. To simplify a comparison of the different techniques, they were applied to the common data from 3.1. This was already used in [12], [20], [16] and offers the advantage of being understable without special knowledge. From now on, this dataset will be referenced as "car example". Unless otherwise noted the illustrations were created by the author using custom implementations of the respective techniques.

- *Bartholomaeus Steinmayr is studying Informatics at the University of Munich, Germany, E-mail: bartholomaeus.steinmayr@campus.lmu.de*
- *This research paper was written for the Media Informatics Advanced Seminar on Information Visualization, 2008/2009*

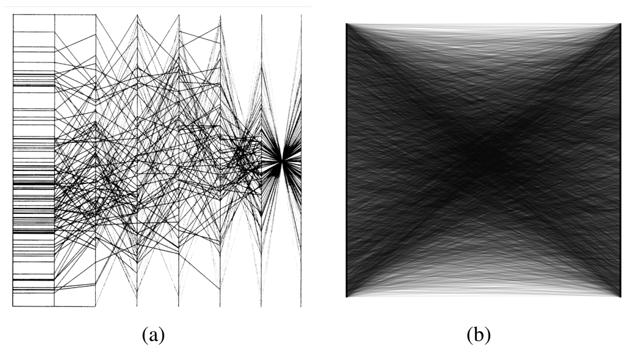


Fig. 1. (a) Correlated data, $\rho = 1, .8, .2, 0, -.2, -.8, -1$ [20]. 1(b) 10000 uncorrelated datapoints

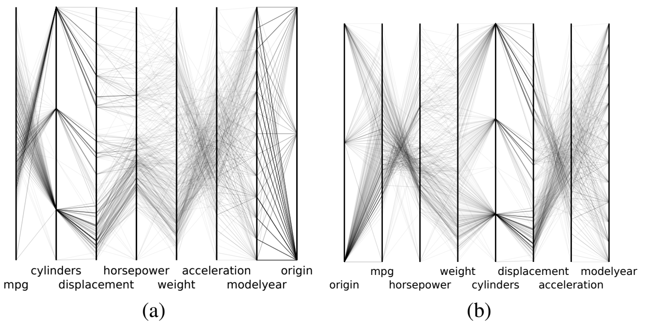


Fig. 2. Two permutations of the car example data from 3.1 Both graphs show the same data, but with different orders of the axes

4.1 Traditional projection techniques

Projection techniques work by transforming the p-dimensional coordinates of the points in the dataset and in the process reduce the dimensionality. The resulting points can then be directly rendered in the $\mathbb{R}^2$.

4.1.1 Parallel Coordinates

Parallel coordinates were first introduced by Maurice d'Ocagne in 1885 [4]. They gained wide popularity after Inselbergs independent discovery [10]. Inselberg designed parallel coordinates with multidimensional geometry in mind and the example application in [10] used parallel coordinates to implement an air traffic collision avoidance system. This technique has a mathematical basis which exceeds most other methods.

A point $p = (x_1, ..., x_p)$ is usually rendered in euclidian space with orthogonal coordinate axes. The main problem with this is the quick consumption of space. Therefore, this rendering method only works if $p \leq 3$, when no projection is applied. Parallel coordinates abandon the orthogonality of the coordinate axes. Instead, the axes are aligned in parallel on a plane, in an arbitrary order. Each point in $\mathbb{R}^p$ is then represented as a polyline through the values of each dimension on the respective axis. This has two fundamental advantages: Firstly, the space occupied by the plot is only linear with p. This means that data with dozens of dimensions can still be easily plotted. Secondly, there are no ambiguities in the representation of data, while still representing all dimensions. Therefore, in contrast to other projection techniques, there is no loss of information.

The parallel coordinate representation exhibits a fundamental duality with cartesian coordinates: As aforementioned, points in the cartesian space are mapped to lines in parallel coordinates. Furthermore, a line in $\mathbb{R}^p$ consists of an infinite number of points. A subset of n of these points can be represented in parallel coordinates. The line segments between the coordinate axes intersect in $p-1$ points (these points do not necessarily lie between the axis; in extreme cases the lines can be parallel, meaning an intersection at infinite distance). Therefore, a line in cartesian space is represented by points (or a single point, if $p = 2$) in parallel coordinates. Wegman calls this “point-line, line-point duality” [20].

Besides intersection points, the line segments between two axes also define an envelope. Using this envelope, the duality is valid for more complex objects: An ellipse in the plane defined by two axes results in a hyperbolic envelope if the axes are neighbours in the parallel coordinate plot. This has implications for the interpretation of statistical data in parallel coordinate plots. The scatterplot of two uncorrelated variables approximates an ellipse. Two perfectly correlated variables on the other hand form a straight line.

Using these properties, neighbouring axes in a parallel coordinate plot can easily be examined for correlations. Figure 1(a) illustrates this. It shows 8 variables, with correlations between the variables ranging from 1 to -1. Correlated data appears as parallel lines or lines intersecting in a single point. Parallel and intersecting lines respectively represent a positive and negative correlation. The hyperbolic form of uncorrelated variables is less obvious. Figure 1(b) shows a plot of 10000 uncorrelated random datapoints rendered with low opacity. This plot exhibits the hyperbolic envelope more clearly. However, because of the random nature of the data, a perfect hyperbolic curve is not to be expected.

Figure 2(a) shows a plot of the car example from 3.1. Multiple properties of the dataset can be seen from this plot: There is a strong negative correlation between mileage and number of cylinders. Displacement and horsepower of the engine are positively correlated, while weight and acceleration are negatively correlated. Furthermore, the majority of the cars in the dataset come from the US (origin 1). Further properties can be discovered when a different permutation of the axes is examined.

Figure 2(b) shows such a permuted a plot. Here, it can be seen that European carmakers (origin 2) offer the most fuel efficient cars, whereas American companies (origin 1) produce more low-mileage cars and Japanese more high-mileage cars. Furthermore a positive correlation between cylinder count, weight and displacement can be seen. The negative correlation between displacement and acceleration is surprising, but can be explained by the fact that high-displacement cars represent cars with a higher weight, which the increased horsepower can not make up for.

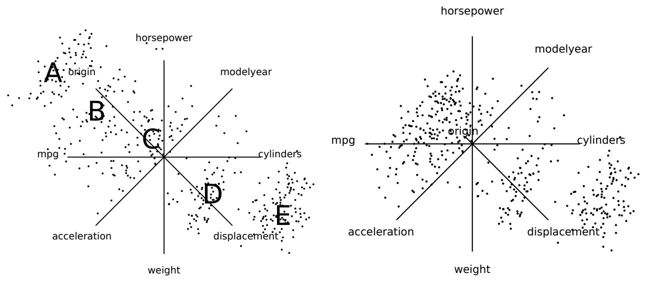
These graphs also illustrate the strengths and weaknesses of the parallel coordinate method.

Advantages:

- The values of all variables can be determined exactly without any ambiguities.
- Statistical properties of the dataset can be discovered easily.
- The distributions of the individual attributes over the entire dataset can be seen directly. Spence [18] calls this “attribute visibility”.
- Consumption of space is only linear with the number of variables.

Disadvantages:

- The extraction of statistical information from the dataset is highly dependent on the arrangement of axes. This can potentially be remedied with automatic reordering methods, as proposed by Peng et al. [16]. Despite these efforts, $\lceil (p-1)/2 \rceil$ (where the brackets represent the ceiling function) [20] permutations have to be examined for p dimensional data to visualize all coordinate pairs.
- It is claimed in [1] that datasets with more than 5000 items are likely to suffer from overdraw. This happens when the lines of many datapoints cover the same area and the density can no longer be estimated. This effect can be countered by drawing



(a) (b)

Fig. 3. Data from the car example visualized with star coordinates. (a) exhibits five clusters. In (b) the origin axis has been reduced to reveal that three clusters belong together.

the lines slighty transparently, as has been done in Figures 1(b), 2(a) and 2(b). However, with this rendering outliers are obscured easily (see for example the cars with three cylinders in 2(a)).

- Examining a single datapoint is complicated (or impossible, depending on the size of the dataset) without interactive tools. This makes comparing individual entities difficult.

4.1.2 Star Coordinate Plots

Star coordinate plots were first introduced by Kandogan in 2000 [11]. Despite the similar name, they differ significantly from star glyph plots (often just called star plots). Star plots [3] are similar to parallel coordinate plots. Instead of a parallel arrangement, the axes are ordered in a radial fashion (reminiscent of a star). The datapoints are then drawn as closed polygons with the vertices on the coordinate axes.

Star coordinate plots on the other hand are quite different. The basic principle of arranging axes in a star-like fashion remains. However, each axis is interpreted as a vector of unit length in euclidian $\mathbb{R}^2$. Data is then represented as points, by using the following method: First, all variables are normalized to unit scale. Then, for each p-dimensional datapoint, the normalized value of a variable is multiplied with the associated vector. The resulting p vectors are accumulated and the final result is rendered.

In contrast to parallel coordinate plots, the position of points in a star coordinate system are ambigious. Consider the example of a point sharing similar values on two axes pointing in opposites directions. The contributions to the final position cancel each other out and the user cannot determine the actual magnitude of the values.

Figure 3(a) shows the car example data visualized with star coordinates. The graph reveals the existance of five clusters (A-E) in the dataset. To further analyze the dataset, Kandogan added interaction techniques to the visualization. The most fundamental of these is the scaling of axes by changing the length of the vector representing a specific variable. In Figure 3(b) this transformation has been applied to give the origin vector a length of 0.1. As a result, the clusters A-C merged. This suggests that clusters A, B and C represent different countries of origin. Because D and E are unaffected by the transformation, they can be expected to be clusters of different engine configurations, originating from the same country.

Advantages:

- Representation of data as points generates little overdraw.

- Clustering and correlations of variables can be discovered easily with the application of the interative tools.

Disadvantages:

- Representation of data is ambiguous. Without the interactive extensions, the values of a single datapoint can not be determined. Furthermore, apparent clusters and correlations can be illusions caused by hidden attributes. Again, properties of a dataset can not be reliably deducted from a single view.

- Discovery of correlations potentially requires enabling and disabling of a number of axes in a similar magnitude as the permutations of parallel coordinates.

4.2 Extensive rendering techniques

The techniques listed here use various method to represent the p-dimensional points of the data. The common characteristic is the location of the structures representing the data points no longer necessarily encodes the values of the data points.

4.2.1 Hierarchical techniques

Hierarchical techniques work by imposing a hierarchy on the data attributes and using it in the rendering process. The following two techniques have been grouped in this category because of their similarity. Despite the fact that the resulting images look quite different, the underlying algorithms are similar.

LeBlanc et al. [15] proposed a hierarchical technique called “dimensional stacking”. The method is restricted to data in which each dimension only consists of a finite set of values. This can be achieved by applying a binning process to continuous variables. Furthermore, the technique is based on the assumption that each datapoint in p-space is assigned a value, which means that the dataset is effectively a sampling of an function of p parameters. The data is rendered in the following way: First, dimensions of the data d1, d2 are assigned to the axes of the screen space. If the cardinality of the dimensions is less than the resolution of the display device (which is required by the technique), the discrete values divide the screen space into a grid of rectangles. Each of these rectangles has a specific value of the dimensions d_1 and d_2 assigned to each of its axes. This method can then be applied recursively until all dimensions are assigned. All but the outer two dimensions are repeated in the subdivided spaces. The frequency of the repetition depends on the order in which the dimensions are assigned. Dimensions divided early are therefore designated “slow”, whereas the dimensions divided later are called “fast”.

Figure 4(a) shows a rendering of the dataset from the car example. For the sake of simplicity, origin and modelyear were removed. All other dimensions were divided into two bins. Horizontal dimensions (slowest to fastes) are mileage, displacement and weight. Vertical dimensions are cylinders, horsepower and acceleration. The cluster in the lower left shows low-displacement, high-mileage cars. The cluster in the upper left is cars with a higher number of cylinders and a low mileage. However, the remaining structure represents all other cars in the dataset, without a very clear structure. As can be seen, even with the applied reductions of complexity, the emerging patterns are hard to interpret.

The main advantage of dimensional stacking is its efficient usage of space. There is no overdraw under any circumstances. However, there are also some disadvantages:

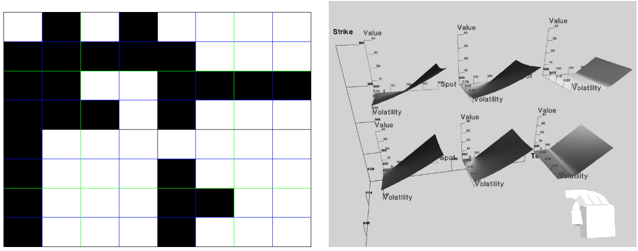
- The method works best only with datasets with a high number of datapoints.

- The resulting visualization highly depends on the speeds the user assigns to the variables. Again, a large number of permutations must potentially be sampled.

- The necessary binning process hides information (unless the cardinality of all variables is low enough to make binning obsolete).

“Worlds within worlds”, introduced in 1990 by Feiner and Beshers [7] used a similar recursive subdivision of space. However, it extends the concept to a three-dimensional user interface. This user interface consists of a stereoscopic display and an input device called DataGlove. This is a wearable device with a magnetic sensor detecting position and orientation of the wearer’s hand. Furthermore, it can detect the position of each finger, allowing the user to execute complicated gestures.

The main goal of this technique, like dimensional stacking, is the visualization of hypervariate functions, as opposed to mere point clouds.



(a) (b)

Fig. 4. (a) Dimensional stacking of the car example created with xmdv-tool [19]. (b): Six views of a function with Worlds within worlds [7]

The basic structure of "Worlds within worlds" is a cube. This is analogous to the screen space divided in dimensional stacking. Again, two of the axes of the cube are assigned a variable. All other variables are fixed to certain values. The function can then be evaluated and its result drawn on the third axis of the cube. This procedure eliminates the information from the fixed variables. It is therefore added back in a controlled fashion. The cube containing the graph is embedded into a larger cube. This cube has three of the fixed variables assigned to its axes. Using the DataGlove the user can move the embedded cube. The position inside the larger cube determines the values of the fixed variables. This allows the user to interactively explore the dataset in many dimensions. Embedding can performed until all variables have been represented. It is also possible to embed several views in the same group, allowing the user to compare two graphs with different parameters. Further interaction possibilities include rotation of the cubes to view the graph from different angles.

Figure 4(b) illustrates this: It shows the plot of a function for six different values of the fixed variable.

Advantages:

- The three-dimensional user interface requires less dimensional reduction than two-dimensional techniques.
- The use of the DataGlove allows the user to work in the three-dimensional representation very intuitively.

Disadvantages:

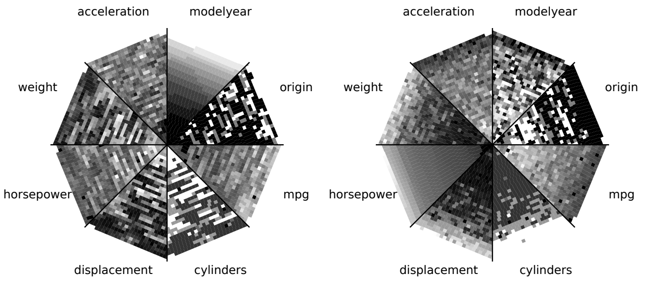
- This method requires highly specialized interface devices (DataGlove and stereoscopic display). It seems therefore unlikely that this method will gain widespread use very soon.
- The method is designed for multi-dimensional, continuous functions. However, in most information visualization problems, data exists as discrete point clouds.

4.2.2 Pixel-oriented techniques

The common idea of pixel-oriented technique is that each pixel of the display represents one data value. Because of the high resolution of modern displays, very large datasets can be visualized.

One such method is "Circle Segments", introduced by Ankerst et al. [1]. Circle segments renders p-dimensional data in the following way: First, the rendering space is divided into p equiangular segments. This means that only data with at least three dimensions can be visualized with circle segments (otherwise the segments would not be triangular). Each of these segments represents one data dimension. The data points are then rendered by following a drawline through the segments. This line starts in the center of the display. It is advanced one pixel at a time perpendicular to the line that halves the segment. For each pixel, a data value is visualized according to some coloring scheme. When the drawline hits the border of a segment, it is advanced outwards by one pixel and the direction is reversed.

Because of this iterative rendering, the sorting of the data is crucial. If the data points were rendered in a random order, each segment could be expected to contain noise, which makes it hard to gain insight into the data. The data therefore has to be sorted. This can either happen implicitely (for example by gathering the data in a chronological fashion) or explicitely (by sorting it based on a given dimension). The segment by which the dataset is sorted exhibits a smooth (depending on its cardinality) gradient in the rendering. Dimensions that are positively correlated will show (approximately, depending on the strength of the correlation) the same gradient. Negatively correlated dimensions will show the inverse gradient.



(a) (b)

Fig. 5. Circle segments renderings of car example data. Left: Sorted by model year. Right: Sorted by horsepower

Ankerst et al. [1] used stock market data to compare circle segments to traditional line graphs. This data consists of the prices of fifty stocks recorded at approximately 5000 points in time. The line graph of the data was created by using the horizontal axis for time and the vertical axis for price and drawing 50 lines into this coordinate system. This exposes two problems: Firstly, most screens are not wide enough to represent the 5000 points in time, which means that values have to be discarded or averaged, resulting in a loss of information. Secondly, rendering 50 lines in a single coordinate system causes overdraw and makes it very hard to distinguish the different stocks.

With circle segments however, most screens offer enough resolution to render all data values. Furthermore, no overdraw happens. These two facts mean that data is rendered without loss of information.

Figure 5(a) shows a rendering of the car example. It is originally sorted by modelyear (as can be seen by the smooth gradient in the modelyear segment). From this view few properties of the dataset can be deducted, illustrating the importance of the sorting of the data. There appears to be a slightly positive correlation with mileage and a negative one with horsepower. This means that there might be a trend towards more environmentally friendly cars.

Figure 5(b) shows the same dataset, but sorted by horsepower. The inverse correlation with modelyear is visible again and now a lot more information is visible. There appear to be positive correlations with weight, displacement and cylinders. The surprising negative correlation with acceleration already mentioned in 4.1.1 shows up as well. Finally, it can be seen that the top 10% horsepower cars come from origin 1 (US).

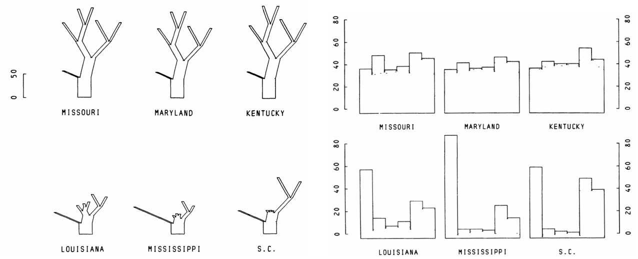
These examples illustrate the advantages and disadvantages of the circle segments technique.

Advantages:

- Analyzing correlations is intuitive in circle segments.
- The example from [1] shows that circle segments are a good technique for visualizing datasets with many ($>$ 1000) points and ($>$ 10) dimensions.

Disadvantages:

- In comparison to other techniques, circle segments has even more options for arranging the data. For one, the assignment of dimensions to the segments can be changed. This is analogous to for example parallel coordinates. Furthermore, the ordering of the dataself is crucial to the information discovery process. It might therefore be required to examine a multitude of different



(a) (b)

Fig. 6. Vote data visualized by trees and castles. [14] 6(a): Tree method. 6(b): Castle method.

dimensional and data orders, which is only feasible in an interactive environment.

- Evaluating the properties of a single datapoint or direct comparison of two points is impossible. This is caused by the fact that the values of a single datapoint are spread over the six segments and it is not possible to visually identify the pixels belonging to one point. This is a weakness of all pixel-based techniques.
- Extracting the exact value of a datapoint from the coloring of a pixel is impossible.
- Datasets need to be ordered in some way or contain at least one attribute which defines a natural ordering.

4.2.3 Iconographic techniques

Iconographic visualization techniques work by representing each point of a dataset as a symbol, the properties of which carry the information.

One example for this class of techniques is "trees and castles" created by Kleiner and Hartigan [14]. A problem all methods discussed so far suffer from, is the fact that the user has to decide how to order the data. Kleiner and Hartigan proposed to solve this by generating the visualization based on an automatically calculated clustering of the variables. They suggested usage of a method called complete linkage, which works as follows: The distance of all pairs of variables is determined by an euclidian measure. The two variables closest to each other form the first cluster. The distance of a cluster and a variable is defined as the maximum distance between any variable in the cluster and that variable. Successively the two closest clusters and/or variables are merged until all variables are grouped.

This process creates a tree graph (in the mathematical sense of the word) where the leaves represent variables and the nodes represent clusters of increasing size (towards the root). This tree graph is then used as the basis for two different rendering methods. The first method ("trees") renders the datapoints as actual trees. The method begins by rendering the root of the graph as the stem. The graph is then traversed and nodes further down are rendered as branches emerging from the stem. The information of the variables is encoded in the width and length of the branches. The width is proportional to the number of variables above a branch, while the length is proportional to the average value of all variables above a branch.

The second method ("castles") is similar to a bargraph. The variables are arranged in the same order as their appearance in the leaves of the tree graph. The tree graph is then rendered, similar to the previous method, using the following template: All "branches" have zero angle. Each branch ends at a distance v from the bottom of the graph. v is the minimum value of all variables above the branch, minus a factor $q * d$. q is the number of branches between the current branch and the smallest variable. d is an abitrary treshold value. The result is a set of bars, were the height of each bar corresponds to the value of one variable. Furthermore, the structure of the tree graph can be seen by the dividing lines between the bars.

Figures 6 illustrates both techniques with the same dataset. It contains the percentage of republican votes in six US states in the years 1932, 1936, 1940, 1960, 1964 and 1968. Both methods show that the results in the first three states are quite similar. Furthermore, the exceptionally high republican result in 1964 in the last three states can be detected.

Overall, several advantages and drawbacks emerge.

Advantages:

- The clustering of the variables is directly visible in the structure of the glyphs.
- The overall structure of the trees can still be compared with a large number of variables.
- The user does not have to decide how to arrange the variables.

Disadvantages:

- All iconographic techniques require large amounts of space for a single datapoint. Aside from the space requirements, the display quickly gets confusing for larger number of points (¿10). Therefore, only small datasets can be visualized.
- The correlations between variables cannot be analyzed in detail.
- The comparison of individual values is difficult for the trees rendering method, even within the same tree.

4.2.4 Integration of human interaction

Some of the problems of hypervariate information visualization can be solved by allowing the user interactively manipulate the representation of the data. One method designed around this paradigm is "dust & magnet", invented by Yi et al. with the goal of creating an "easy-to-learn and easy-to-use" [21] visualization method. The basic rendering technique can be compared to star coordinates (see 4.1.2). The main similarity is that data is represented as points which are called "dust". The concept of coordinate axes is abandoned. Instead, dimensions are represented through the metaphor of magnets.

After loading a dataset, all points of dust are in the center of the screen, occluding each other. The magnets representing the data attributes can be placed on the display by the user via dragging and dropping. The dust is then attracted to the magnets. The strength of the attraction is defined by the relative value of the variable assigned to the magnet. In contrast to a real magnet, the attraction does not depend on the distance of points to the magnet. A physically accurate behaviour would result in all datapoints sticking to the magnet beyond a certain distance. Furthermore, the attraction does not apply an acceleration to the dust as a real magnet would, but instead applies a constant speed. To allow the user to view a stable snapshot of the system, the dust is only animated when one of the magnets is dragged with the mouse. When several magnets are on the screen, all of them attract the dust particles at the same time. The direction of travel of the dust is calculated by vector summation of the "attraction vectors" of the magnets. This is again a similarity to star coordinate plots.

The fundamental feature which is different from the methods discussed before is that dust & magnet is essentially designed with user interaction in mind. By arranging the magnets on the screen, a wide variety of analyzation tasks can be accomplished. For example, when the magnets are arranged in a circular fashion, dust & magnet effectively emulates star coordinate plots. Another possibility is arranging the magnets of desirable and undesirable attributes in opposite directions. As a result, the datapoints separate, with the points conforming to the desirable attributes moving towards that magnet. Another possibility is to move magnets for desirable attributes into one half of the screen and those undesirable ones in the other. The individual attributes can then be spaced out evenly. This way, an even more detailed selection can be performed and it is possible to rank the desirable attributes. A tradeoff between attributes can then be achieved. This is illustrated in the paper by the example of selecting a suitable breakfast cereal.

Aside from these basic interactions, dust & magnet provides further options for customizing the visualization. The overall strength of each magnet can be adjusted. This is the equivalent of adding a weighting to the dimension this magnet. For each magnet a repellent treshold can be defined. Dust particles with an attribute value below this treshold are repelled from the magnet instead of being attracted. The data can be filtered by defining ranges for attributes. Datapoints with values outside these ranges are not displayed. Lastly, the information of attributes can be encoded in the size and/or color of the dust particles.

However, two problems occur in the visualization: Occlusion of datapoints and lack of reproducability. Occlusion occurs because in after loading a dataset, all datapoints are in the center of the screen. Furthermore, similar datapoints will form groups in similar locations, especially in large datasets. In this situation, it is difficult to distinguish the points and to estimate the number of points in a cluster. To solve this, the authors have implemented a mechanism called "shaking" the dust. This functionality gradually spreads out the dust over the course of several invocations.

Another problem is reproducability. The user will move magnets and add new ones, adjust their strenghts, shake the dust and more before arriving at a solution for her visualization problem. It is hard to recreate these steps to achieve the same result. To solve this, two solutions were implemented: The "center dust" feature will move the dust back to center of the screen. The "attract dust" feature allows the user run the simulation without dragging the magnets. The dust then moves according to its attraction to the magnets, but only for a very short distance.

Overall, dust & magnet exhibits the following strengths and weaknesses.

Advantages:

- Dust & magnet allows easy and iterative approach to selecting datapoints from a dataset. The criteria for this selection need not be accurately defined, but can rather be developed interactively.
- The usage of a metaphor makes the basic principle of the technique easily comprehensible. The underlying mechanics are similar to star coordinate plots. However, those require the user to understand basic vector math. In contrast, dust & magnet only requires playful experience with the behaviour of magnets. This makes dust & magnet a well suited technique for everyday decision making of people without experience in information visualization.

Disadvantages:

- Occlusion of dust requires a manual operation (apply "shake dust") to resolve.
- The features for supporting reproducability need to be improved. Even when the dust is centered, complicated movements of the magnets will be hard to reproduce. The inclusion of a recording feature would be a valuable addition. This feature would allow the user to recall the simulation and magnet movements and replay it or allow others to comprehend the process leading to the presented results.
- Dust & magnet is not very well suited for statistical analysis of data. In the user study presented in the paper, one task was to decide whether there was a correlation between two variables of a dataset. Only 50% of the participants were able to answer this correctly.

5 Evaluation of different techniques

All methods mentioned before are viable options for visualizing data. Some of them seem to perform better than others, depending on the type of data being analyzed.

However, few user studies were conducted to evaluate the proposed visualizations. The only one appeared in [21] and was performed with a small group of only 6 participants. Furthermore, the authors did not provide a quantified assessment of the technique. To quote Kleiner and Hartigan: "We have not carried out any formal psychological experiments, although the results of such experiments could be useful in comparing techniques and for suggesting modifications of techniques. We are far from knowing what makes a good picture." [14]

Keim et al. claimed that "until we develop a basis for evaluation, we will not be able to get beyond this current demonstrational stage" [13]. As a foundation for such an evaluation, they proposed a method for generating test data with predefined properties. This poses a problem: Visualization techniques are used to discover structure in the data, but there is no precise definition of what structure in general even means. However, in the field of statistics, such exact definitions exist. Therefore, the introduced method is restricted to such statistical data. The proposed technique distinguishes between dimensions which have a natural order and can be organized in dense linear arrays and those that are unordered. The user can then define the number of dimensions and the number of arrays. For example, by setting the number of dimensions to three and the number of arrays to two, one could create a simulation of image data.

Furthermore, the value ranges and distributions of each dimension can be specified. The correlations between dimensions are defined through functional dependencies. That means that the values of a dependant dimension are created by applying a perturbation on the values of the independent dimension. In this fashion, any kind of linear or non-linear correlation can be simulated. Furthermore, rectangular regions for which these data generation parameters apply can be defined. By creating such regions, the user can simulate clusters in the data.

These controls allow the creation of datasets with known and adjustable parameters. Such datasets are a fundamental requirement for precise evaluation of visualization techniques. For example, one could gradually change a parameter of the dataset and determine the minimal value for which this parameter is noticed by the users. The result is a measure for the sensitivity of a method for a certain characteristic.

However, Keim et al. left providing an actual methodology for evaluating visualization techniques as future work. A first approach to defining an actual metric for visualization methods was presented by Grinstein et al. [9]. This metric measures three properties of a visualization method: [9]

- Given an n-dimensional space, the **intrinsic dimension** is the largest integer k for which all vectors in a set of k "unit vectors" [9] can be uniquely identified in the visualization. What is meant with the slighty imprecise wording "unit vectors" is the set of vector of the form $(0,...,1,0,0)$. That is, the vectors defining the coordinate axes of the n-space.

  For example, in a two-dimensional scatterplot, all unit vectors are mapped to the screen coordinates $(0,0)$, $(1,0)$ or $(0,1)$. If $n > 3$, only $(1,0)$ and $(0,1)$ uniquely identify a vector, giving an intrinsic dimension of two.

- The **intrinsic record ratio** is defined as k/n. n is again the dimensionality of the underlying space. The set of p binary vectors is the set of 2^p p-dimensional vectors for which each component is either 0 or 1. k is the largest value for which all 2^n binary vectors can be identified in the n-dimensional space [9]. This statement is somewhat obscure, as the second part does not depend on k in any way.

  The intrinsic record ratio of a two-dimensional scatterplot is $4/2^n$ since only the four vectors $(0,0)$, $(1,0)$, $(0,1)$ and $(1,1)$ can be identified.

- The **intrinsic coordinate dimension** is the largest k for which k coordinates of any vector in the given n-space can be determined.

  The intrinsic coordinate dimension of a two-dimensional scatterplot is two, since only the dimensions on the axes of the plot can be determined.

These metrics can be applied to almost all visualization techniques. For techniques using color to encode values, the intrinsic coordinate dimension can not be determined. However, the authors did not discuss how the results of these metrics translate into fitness of a method for a specific application. For the techniques tested by the authors, the measures mostly assumed their extreme values [9]. This further complicates ranking of different methods.

6 Conclusion

Users can nowadays choose from a very wide range of visualization methods. However, finding the "right tool for the job" is still mostly intuition. The same is true for the creation of new methods. All of this could change with the advent of accurate metrics and testing methodologies for visualiziation techniques. Nevertheless, I believe that the human mind will remain at the center of qualitative information processing. This means however, that an evaluation of a visualization method will have to take account of this human factor. To achieve this, one either has to perform user studies or would need access to an extensive model of human cognition. Since such a model is not likely to be developed very soon, reliable classification of techniques will require large user studies. These studies are complicated and expensive though and will thus remain reserved for only the most popular of methods.

References

[1] M. Ankerst, D. Keim, and H. Kriegel. Circle Segments: A Technique for Visually Exploring Large Multidimensional Data Sets. *Dim*, 1501:1, 1996.

[2] A. Bartkowiak and A. Szustalewicz. Some modern techniques for viewing multivariate data-a comparative look. In *Workshop on Intelligent Information Systems VIII, Ustron, Poland*, pages 7–11, 1999.

[3] J. Coekin. A versatile presentation of parameters for rapid recognition of total state. In *Proceedings of the IEE International Symposium on Man-Machine Sy. rtems*, 1969.

[4] M. d'Ocagne. *Coordonnes Parallles et Axiales: Mthode de transformation gomtrique et procd nouveau de calcul graphique dduits de la consideration des coordonnes parallllles*. Gauthier-Villars, 1885.

[5] P. Ehrenfest. In that way does it become manifest in the fundamental laws of physics that space has three dimensions? *Koninklijke Nederlandsche Akademie van Wetenschappen Proceedings*, 20(1), 1918.

[6] L. Fahrmeir, R. Kunstler, I. Pigeot, and G. Tutz. *Statistik*. Springer, 2004.

[7] S. Feiner and C. Beshers. Worlds within worlds: metaphors for exploring n-dimensional virtual worlds. In *Proceedings of the 3rd annual ACM SIGGRAPH symposium on User interface software and technology*, pages 76–83. ACM New York, NY, USA, 1990.

[8] W. Fleming. Overview of automotive sensors. *IEEE Sensors Journal*, 1(4):296–308, 2001.

[9] G. Grinstein, M. Trutschl, and U. Cvek. High-dimensional visualizations. In *Proceedings of the 7th Data Mining Conference-KDD*, 2001.

[10] A. Inselberg, B. Dimsdale, I. Center, and C. Los Angeles. Parallel coordinates: a tool for visualizing multi-dimensionalgeometry. In *Visualization, 1990. Visualization'90., Proceedings of the First IEEE Conference on*, pages 361–378, 1990.

[11] E. Kandogan. Star Coordinates: A Multi-dimensional Visualization Technique with Uniform Treatment of Dimensions. In *Proceedings of the IEEE Information Visualization Symposium*, 2000.

[12] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 107–116. ACM New York, NY, USA, 2001.

[13] D. Keim, R. Bergeron, and R. Pickett. Test Data Sets for Evaluating Data Visualization Techniques. *Perceptual Issues in Visualization*, pages 9–22, 1994.

[14] B. Kleiner and J. Hartigan. Representing Points in Many Dimensions by Trees and Castles. *J. AM. STAT. ASSOC.*, 76(374):260–269, 1981.

[15] J. LeBlanc, M. Ward, and N. Wittels. Exploring N-dimensional databases. In *Visualization, 1990. Visualization'90., Proceedings of the First IEEE Conference on*, pages 230–237, 1990.

[16] W. Peng, M. Ward, and E. Rundensteiner. Clutter Reduction in Multi-Dimensional Data Visualization Using Dimension Reordering. In *Proceedings of the IEEE Symposium on Information Visualization*, pages 89–96. IEEE Computer Society Washington, DC, USA, 2004.

[17] E. Ramos and D. Donoho. ASA Cars Dataset. In *American Statistical Association Second Exposition of Statistical Graphics Technology*, 1983. `http://lib.stat.cmu.edu/datasets/`.

[18] R. Spence. *Information Visualization: Design for Interaction*. Prentice-Hall, Inc. Upper Saddle River, NJ, USA, 2007.

[19] M. Ward. XmdvTool: integrating multiple methods for visualizing multivariate data. In *Proceedings of the conference on Visualization'94*, pages 326–333. IEEE Computer Society Press Los Alamitos, CA, USA, 1994.

[20] E. Wegman. Hyperdimensional data analysis using parallel coordinates. *Journal of the American Statistical Association*, 85(411):664–675, 1990.

[21] J. Yi, R. Melton, J. Stasko, and J. Jacko. Dust & Magnet: multivariate information visualization using a magnet metaphor. *Information Visualization*, 4(4):239–256, 2005.