Interactive Visual Analysis of Set-Typed Data

Wolfgang Freiler, Krešimir Matković, and Helwig Hauser

Abstract— While it is quite typical to deal with attributes of different data types in the visualization of heterogeneous, multivariate datasets, most existing techniques still focus on the most usual data types such as numerical attributes or strings.
In this paper we present a new approach to the interactive visual exploration and analysis of data that contains attributes which are of *set* type. A set-typed attribute of a data item – like one cell in a table – has a list of $n \geq 0$ elements as a value. We present the set'o'gram as a visualization approach to represent data of set type and to enable interactive visual analysis. We also demonstrate how this approach is capable to help in dealing with datasets that have truly many dimensions (more than a dozen or more), especially in the context of categorical data.
To illustrate the effectiveness of our approach, we present the interactive visual analysis of a CRM dataset with data from a questionnaire on the education and shopping habits of about 90000 people.

Index Terms—Interactive Visual Analysis, Multidimensional Multivariate Data, Categorical Data, Interaction, Focus & Context, Multiple Coordinated Views.

1 Introduction

Data and information, as commonly addressed by visualization, are often heterogeneous. A large amount of time is generally spent on data preparation and cleaning. Often data has to be reorganized as the methods at hand often only support few types of data. Columns by rows data tables, which often are the structural basis for data visualization, usually support numerical data. Often also nominal and categorical data are handled. More complex data components such as higher-dimensional values (vectors, tensors, etc.) or images, longer texts, or videos are supported increasingly often (even though still only in few cases). Set-typed data, however, which generally shows up quite naturally, is usually not treated in its inherent (set) form (i.e., without being transformed into another form, e.g., multiple binary columns). Cars (or products in general) have sets of properties, documents have sets of keywords, photos have tags, patients have symptoms, etc. Properties, keywords, tags, etc., have in common, that they are of set type by nature. Usually, sets are converted to multiple categorical (binary) dimensions, because those are manageable in most existing visualization frameworks.

We introduce interactive visual analysis of set-typed data. We provide a solution for treating set-typed data also as sets in the visualization. This makes it possible to deeply explore set properties, and to better understand correlations among the set elements. This is also much more intuitive, since set elements semantically belong to a set, and users are used to treat them as sets in real life. For visualization, we propose a new visual metaphor – here called the set'o'gram – to convey relevant information about a set-typed data dimension. To show that this approach also integrates with the powerful analysis concept of coordinated, multiple view, we also demonstrate, how standard views can be adapted to show set-typed data, such as the histogram, the scatterplot, and parallel coordinates. We note, however, that it were exactly our (moderately satisfying) experiences from "just adapting" standard views that led to the development of our set'o'gram.

Special attention is paid to interaction and analysis of set-typed data. Useful analysis procedures and brushing capabilities are described. We have implemented the proposed technology in a coordinated multiple view environment. The duality between multiple, binary (categorical) dimensions and set-typed data also allows for a very useful approach to dimension reduction. Given a dataset with multiple categorical dimensions, several of them can be grouped together, and represented as one, set-typed dimension, thus leading to dimension reduction in visualization.

The remainder of this paper is organized as follows: we discuss related work in section 2, before we detail on the new data model (section 3). In section 4, we introduce our new visualization approach, and in section 5 we describe how visual data analysis is supported. We demonstrate the utility of our approach by presenting an analysis of CRM data (more than 90000 questionaires) in section 5.

2 Related Work

According to our knowledge, there is no previous work on visualizing set-typed data (in its inherent form, i.e., without representing it differently). However, a number of relevant previous works are closely related (such as the visualization of categorical data). Hofmann [3], for example, developed several useful techniques for the visualization of categorical data. Mosaic plots are a useful tool when two categorical dimensions are investigated. Mosaic plots can be seen as nested histograms, using the width of bars rather than the height, to reflect the amount of data items as represented by a visual element. Due to the fact that mosaic plots divide the available screen-space recursively as proposed by Friendly et al. [2], the order of the dimensions has a significant impact on the visualization. Spence et al. [11] have developed the "Attribute Explorer", an application which helps finding interesting data records in large datasets by defining interesting ranges of values for various dimensions. A widget called LinkCrystal, allows different coloring of data records that satisfy different combinations of the user-supplied selections. With pixel bar charts, Keim et al. [6] showed a technique, that does not only use the size of a rectangle to represent data, but also uses the area to describe it further. Every data item is represented by a pixel with a color that displays a numerical value. In 2002, hierarchical pixel bar charts have been introduced [7] – they allow to display hierarchies by dividing bars and displaying the resulting thinner bars a bit lower. This way of displaying hierarchies works best with few subdivisions.

When Inselberg et al. [4] introduced Parallel Coordinates, they used this technique mainly to display high-dimensional geometry. Later on, it was used in data analysis, because a relatively high number of dimensions can be visualized conveniently in 2D visualization space. However, parallel coordinates are not really well-suited for categorical data, so various improvements have been made over time. Rosario et al. [10] proposed to build up a class tree, representing a hierarchy on the available categories. The distance measure obtained by this tree

- *Wolfgang Freiler is with VRVis Research Center Vienna, Austria, E-mail: freiler@vrvis.at.*
- *Krešimir Matković is with VRVis Research Center, Vienna, Austria, E-mail: matkovic@vrvis.at.*
- *Helwig Hauser is with University of Bergen, Norway, E-mail: Helwig.Hauser@uib.no.*

Manuscript received 31 March 2007; accepted 1 August 2007; posted online 27 October 2007.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

can be used to place similar categories next to each other. Kosara et al. [8] introduced parallel sets, a technique that combines the advantages of mosaic plots and parallel coordinates. Instead of lines, bars are used to connect the axes. The heights of these bars are scaled according to the number of data items they represent. By using different colors for each category of a single dimension, the bars can be traced when they split up at other axes. Parallel sets work best with multiple categorical dimensions with only few categories, otherwise split-up bars may become too small to visualized properly.

3 Data Model

The visualization process generally follows a model described by Card et al. [1]. After the data is acquired from simulations, measurements or questionnaires, it has to be prepared and stored in a convenient format for visualization. Data tables are often used for abstract data because of their simplicity. Most average users are familiar with spreadsheet applications like Microsoft Excel, where attributes are listed in columns and each row represents one data record. A data record contains one value per dimension, which is referred to as a data item. A spreadsheet can also be interpreted as a multidimensional space with each row representing a point in space.

Depending on the values each dimension can contain, we distinguish between different data types. There can be one dimensional values like numeric, categorical or nominal values, values of a higher dimension, like vectors, series of values or even objects like full texts or pictures.

There are still datatypes that have not been dealt with from a visualization perspective. One of these datatypes is the *set* as we know it from mathematics. A set is a collection of an arbitrary number of elements, that can also be empty. Such a set of elements can be used as a value to list various attributes that are assigned to a data item. The dimension itself is defined by a superset containing all available elements.

Set-typed data is not a new thing, there are many situations in real life, where we can find sets. Whenever we can map an arbitrary number of semantically connected attributes to data items, using sets is the most natural way to retain the characteristics of the data. Currently such data has been stored using multiple categorical dimensions, which has various disadvantages. Naturally connected items which belong together are disconnected. It is not intuitive and on the same time increases data dimensionality.

To keep the terminology in this document consistent, some terms concerning sets have to be defined. A dimension in a multidimensional multivariate dataset containing set-typed values is called a set-typed dimension. It provides a group of possible elements, that can be assigned to all values in this dimension. If we refer to a value in a set-typed dimension, which contains a subset of all possible elements, we call it a set. We end up with datasets that contains data records in its rows and dimensions in its columns. Each data record contains one value in each dimension. If this dimension is set-typed, the value itself are sets and can be composed of multiple elements.

To demonstrate, how sets can be explored and visualized, we consider a simple dataset containing data about used cars that are for sale in Columbus, Ohio. In this paper we will refer to it as the ”Columbus Dataset”. These cars have various parameters like price or mileage, that can be stored as numerical values. Manufacturer or model names can be stored in categorical dimensions. However, we are also interested in features, like air-conditioning or cruise control. Since each car can have any combination of all available features, we have two possibilities here. We can either store each feature in a single categorical dimension that contains boolean values like ”true” and ”false”, or we can create one set-typed dimension that stores all features of a car in a list of elements. Besides reducing the number of dimensions, we can also rearrange elements and have their order reflected in each view.

4 Integrating Sets in Visualization

The data type *set* has been integrated into a coordinated multiple view framework which supports interactive visual analysis of multivariate data. This framework offers a variety of views using established information visualization techniques. To enable interactive visual analysis of set-typed data, we extended three standard views, which are often used to analyze tabular data, i.e., the histogram view, the 2D scatterplot and the parallel coordinates view.

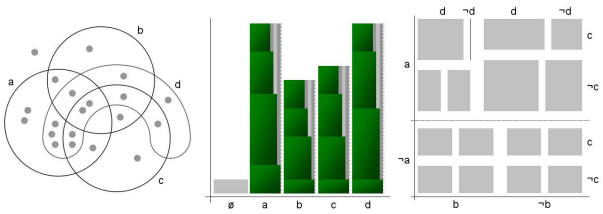

Fig. 1. These three images show different visualization methods for set-typed data. The picture on the left shows a Venn diagram with four shapes. Each shape defines an area, where data items that contain a specific element can be placed. A data item containing two elements must be placed in the intersection area of the two respective shapes. To provide room for each possible combination of elements, new shapes representing new elements have to be placed very carefully. When a new shape is added, it has to cover some part of each area in the view to provide room for data items containing this element. This visualization is very intuitive for a small number of shapes, but it becomes confusing when adding very soon when new elements are added. The picture on the right shows a mosaic plot [3] representing the same dataset. Here the available screenspace is divided recursively for each possible element in our set-typed dimension. The size of the two halves represents the number of data records in this rectangle. However, large amounts of screen space are needed for a higher number of elements, and the resulting visualization heavily depends on the order of subdivisions. It is very easy to estimate the total number of data records containing a, but much harder to do that for data records containing d. The picture in the middle displays the same dataset in a set'o'gram. This visualization shows one bar for each possible element in our set-typed dimension. One data record containing multiple elements is represented by multiple bars in the view. The blocks provide additional information on how many other elements are in the sets represented by a block. However, to find out which other elements are contained in these sets, interaction is needed.

Our visualization framework supports item-based visualization like scatterplots and parallel coordinates, as well as frequency-based visualization like histograms. While in item-based visualization each data item is represented by a visual element, e.g., a dot in a scatterplot, frequency-based visualization represents cardinality or frequency of items with particular characteristics. These two types of visualization behave differently when set-typed dimensions are visualized. In an item-based visualization, each data item is represented by a visual item. The item's position represents its values in one or more dimensions. If a set-typed dimension is visualized, there usually are multiple elements in each set (per data item). This translates into drawing multiple visual items for one data item on the screen, e.g., one graphical element for each feature of a car. In frequency-based visualization, data items are divided into categories, especially when discrete data are addressed. In case of set-typed dimensions, a set containing multiple elements accordingly belongs to multiple categories, also. The frequencies of multiple categories are increased for a single data item.

With the traditional approach, each boolean attribute would have two possibilities, e.g., a car either has a feature or not. In case of set-typed data, cars can have any combination of features, or no feature at all. Just as in mathematics, we also have to account for empty sets, accordingly. We therefore embed an explicit representation of the *empty set* in the visualization. We utilize a differentiated (deemphasized) visual appearance for this special category.

In mathematics, Venn diagrams are often used to illustrate a set. By drawing partly overlapping shapes, different element combinations are assigned to areas. However, with an increasing number of set elements, this approach becomes impractical. It is very hard to make all possible combinations of elements visible, and provide enough room

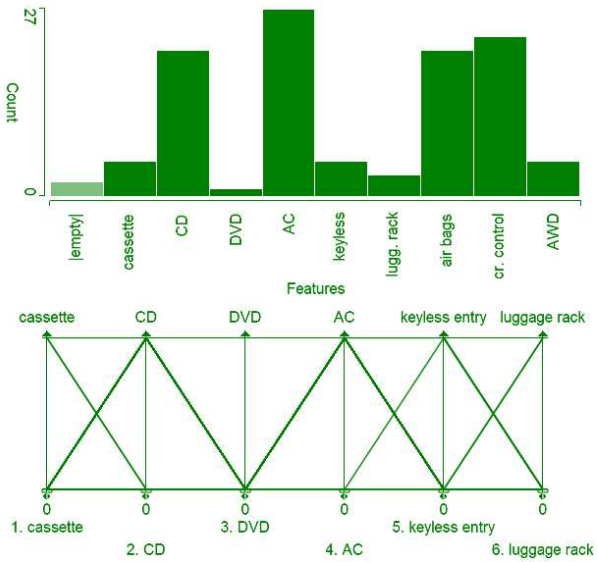

Fig. 2. The upper image shows a histogram displaying one set-typed dimension of the Columbus dataset. Note that cars with multiple features are represented by multiple bars. It is easy to see, which features occur often, and which are rare. However we cannot see which features occur in combination with others. The lower picture shows a parallel coordinates view displaying six features of the cars in our dataset. Parallel coordinates are not well-suited for visualizing dimensions with low cardinalities, because the amount of congruent lines can not be estimated well. The only thing we could tell from this visualization is that there is no car with a DVD player that does not have air-conditioning.

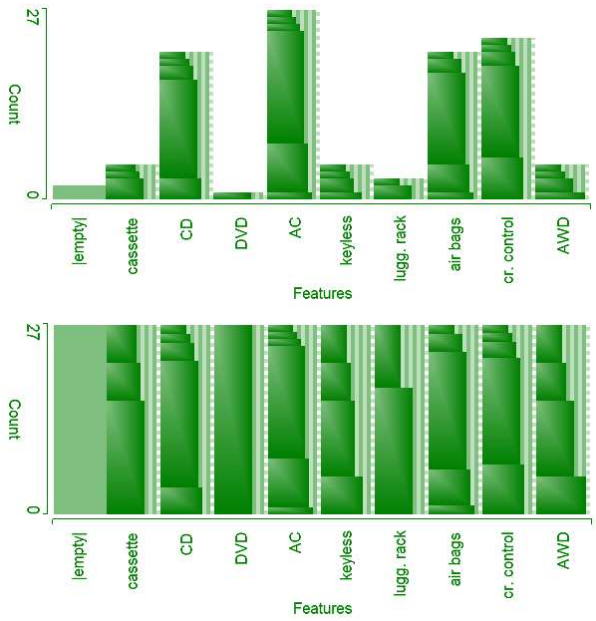

Fig. 3. The upper image shows a set'o'gram of the Columbus dataset depicting the set-typed dimension "Features". The bright bars in the background show the same information as in the histogram. The blocks show the number of different elements in each set. The block of full width in the rightmost bar, for example, represents a car with all-wheel drive but no additional features. The narrower blocks on the bars' tops represent cars with many features. The lower image shows the same set'o'gram in relative mode. In this mode, all bars are scaled to the same height, which makes it easy to compare blocks within single bars.

for data items in areas where these items belong. For example, if we add an element e to figure 1, the additional shape has to cover parts of all areas in the view, without covering one single area completely.

Another approach to combine multiple categorical dimensions are mosaic plots [3]. Here, the available area is subdivided into multiple slices. If we consider set elements, we have to split the area into two parts for each elements, recursively. This works well for few elements, but if we subdivide each rectangle too often, we need large amounts of screen space, and our visualization becomes similar to a treemap [5]. A treemap is well suited for hierarchies, because there the order of subdivisions is already given. A set with multiple elements does not necessarily have a hierarchy, therefore we do not know, which element should be taken into account for the first split. This is an important fact, because the first splits are very well visible in the visualization, while later subdivisions are harder to see.

Because of the problems mentioned before, we decided not to visualize each and every possible combination of set elements. Instead, we will visualize frequencies of elements, and in a second step, we will visualize additional information on the cardinality of sets. This approach needs less screen space and scales well with a higher number of elements. While treemaps would need 2^n rectangles to visualize sets with n elements, we only need n rectangles to show frequencies and n^2 rectangles to visualize cardinalities.

4.1 Sets in Histograms

The histogram is a very useful instrument to examine the distribution of data values of different types. It is well suited for categorical data and can be nicely extended to visualize sets, as shown here. A histogram is composed of vertical bars representing all possible elements of a set-typed dimension. Because one set-typed value usually contains multiple elements, it generally contributes to multiple bars in the histogram. As a consequence, the sum of all histogram contributions generally exceeds the total number of data items. Histogram bars are compared to each other to understand the frequency of each possible element in a set-typed dimension. If we wanted to display the same information as in the histogram in figure 2 without the use of set-aware visualization techniques, we would either require more views (i.e. one histogram for each attribute), or views that can display more than three dimensions, like parallel coordinates. However, figure 2 shows that parallel coordinates are not well suited for dimensions with low cardinalities.

If we use one histogram for each dimension with only two different values, we end up with only two bars representing the values "true" and "false". The set-enabled histogram is simply a collection of all "true" bars of all those "boolean" histograms. The additional empty set bar is not simply a collection of all "false" bars, however. It represents only those items which have "false" values for all features, otherwise the data records represented by this bar would not show up in our visualization at all.

4.2 The Set'o'gram

Although we have already gained some condensation and also some additional information about set-typed data at this point, we still miss other valuable information such as which data items have a (set) value with just one particular element, or maybe two specific ones. Since questions of this kind are quite natural in the analysis of set-typed data, we have to support them in the visualization, also. For conveying more details about the distribution of set-typed data, we utilize the space provided by the individual histogram bars (other approaches such as PixelBarCharts [7] also do this).

One important aspect of interactive visual analysis is to identify deeper relations between elements. Elements show up individually or in combination with each other. The set'o'gram, an extended histogram dedicated to set-typed data has been designed to visualize such relations (see figure 3). Each bar except the first one (which represents the empty set) contains blocks. The maximal number of blocks in a bar equals the maximal number of possible elements in the set-typed dimension. Starting from the bottom, the first block indicates

the number of data items having only one element, i.e., the one which is represented by the bar which contains this block. The second block represents data items which show up in combination with exactly one other element. This (other) element can be different for each of the data items represented by the block, but none of them comes in combination with fewer or more other elements. The next block represents all data items containing two additional elements, and so on. Of course, not all of these blocks have to exist in each bar. Without making use of any interaction techniques we can already gain additional information from the set'o'gram. Wide blocks indicate set elements that do not occur with other elements, tall narrow blocks represent elements that often occur in combination with other elements. This can already be valuable information about a dataset.

In order to distinguish the blocks easily, we vary the blocks' widths. This is necessary since not all blocks have to exists in each bar. With an increasing number of elements in a set, the width of the blocks is reduced. While the first block is of full width, the widths of the other blocks is reduced by a fixed value for each additional element in its sets. Alternating color stripes make it easier to compare the widths. If a set'o'gram displays brushed data, the corresponding amount of data is highlighted in each block and in each bar. To avoid the brushed area of a bar being hidden, one stripe next to it is never covered with blocks. This stripe is also drawn with alternating colors to make it look different from regular bars. Because empty sets do not contain elements, the empty set bar does not contain any blocks (see figure 3).

Another possibility to represent cardinalities is dividing the blocks' widths by the number of elements in the respective sets. In such a visualization, a block representing data items containing two elements, has a width of one half of a bar. Because each one of these data items is represented in two different blocks (once in the bar of its first element, and second time in the bar of its second element) of half width, the amount of screenspace used for blocks is equal for all data items. This avoids overrepresenting items containing many elements, which is sometimes desired. However, in sets with many elements, the differences between block widths become very small, which makes them hard to distinguish.

To help users to distinguish single blocks, they are separated. Especially in sets containing many elements, these blocks can be very small, therefore it is not good to separate them using lines or any other *additional* graphics. Instead we chose to use a color gradient for each block, similar to the cushion approach [12]. The more sophisticated cushion approach also represents the level information, for example, in hierarchies. By making the upper left corner brighter, the borders of all blocks are easy to recognize.

Sometimes it can be very useful to compare blocks inside a bar. Especially blocks in very low bars are hard to compare to each other. To simplify this, the set'o'gram also supports a *relative* mode as shown in figure 3. In this mode, all bars are scaled to the same height. In relative mode, blocks inside bars can easily be compared to each other or to the enclosing bar. It is also very useful to show the relative amount of brushed data for each bar. However, blocks between different bars cannot be compared to each other, because they can be scaled differently.

5 Analysis of Set-Type Data

Up to now we have shown how interested analysts can gain more information out of set-typed data using an innovative visualization approach. In the following, we demonstrate how appropriate interaction and analytic procedures can lead to even deeper insight. Two main tasks can be identified in related analysis procedures. Users want to explore selected sets themselves. Additionally, they aim at exploring correlations of set-type data with other dimensions in the dataset. When analyzing sets is not enough, for example, when comparing prices of cars with many features to those with few, multiple views and interaction techniques are needed.

Highlighting selected blocks of interest, for example, makes the analysis of sets a lot easier and enables additional insights. The main idea is to allow the user to place the mouse pointer over a block, and the respective block is highlighted. Furthermore, all corresponding

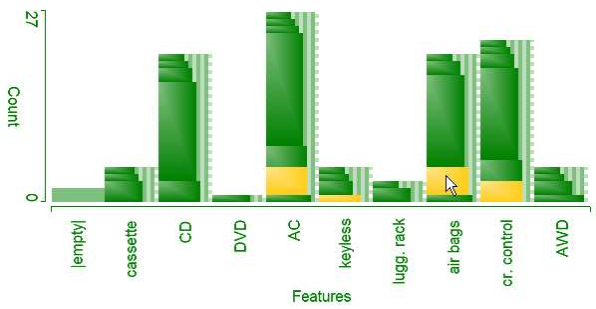

Fig. 4. When the mouse cursor is moved over a block in the set'o'gram, this block is highlighted along with parts of other related blocks.

block parts in other bars are highlighted as well. For illustration we consider a simple example. If the user points to the block which represents cars with airbags and two other features, (figure 4) the corresponding parts of blocks in other bars are highlighted as well. The user can then see, that all these cars also have air-conditioning, one has keyless entry, and three have cruise control. Such relations can be very interesting when analyzing a dataset.

In order to support the analysis of overall correlations in the data set, we use coordinated multiple views. Just as in standard linked views we support brushing (graphical selection of a subset of data in a view) and linking (highlighting corresponding points in all other views). All views are linked and the set'o'gram highlights blocks depending on the selection in other views.

While brushing & linking are state of the art techniques [9], that are used by many others for data exploration, also, there are some specialities to keep in mind when working with sets. The main difference between sets and conventional data types like numbers or categories is that each element of a set must be represented in a view. In case of histograms, this means, that a data item containing multiple elements has to be represented by multiple bars - brushing and linking is affected, accordingly! In scatterplots and parallel coordinates, each element is represented by a separate visual item.

If a data item is brushed, all of its visual representations, including all elements in set-typed dimension, are brushed, too. As mentioned in the previous chapter, these elements can be represented by multiple visual items, that are not necessarily connected to each other (graphically). When brushing one of these items, other items belonging to the same set have to be visually highlighted, also, even if they are outside the brush geometry. As an example we may look for cars with all-wheel drive, and most of these cars will also have other features. The corresponding parts of the visualization that represent them have to be highlighted too.

The newly introduced histogram extension supports brushing of complete bars and brushing of separate blocks. Again, as in every other view that displays sets, there generally are highlighted areas also outside the brush geometry. As another source of information, the set'o'gram does not only highlight parts of bars, but also parts of blocks (see figure 5 for an example).

6 Demonstration

To demonstrate the usefulness of set-typed data, we analyze a large CRM dataset provided by one of our industry partners. This dataset contains information collected with a questionnaire on education, technical devices and shopping habits of more than 93000 people. Its main challenge is the high number of dimensions. The questionnaire contained many multiple choice questions, which can be very useful, because the test persons can specify information in a very detailed way. Instead of picking their favorite shop, users can indicate multiple shops they frequently visit. It is also possible to indicate multiple educational institutions instead of only checking the last one.

If we have to stick with categorical data dimensions, we have to create one dimension for each shop, and one for each educational in-

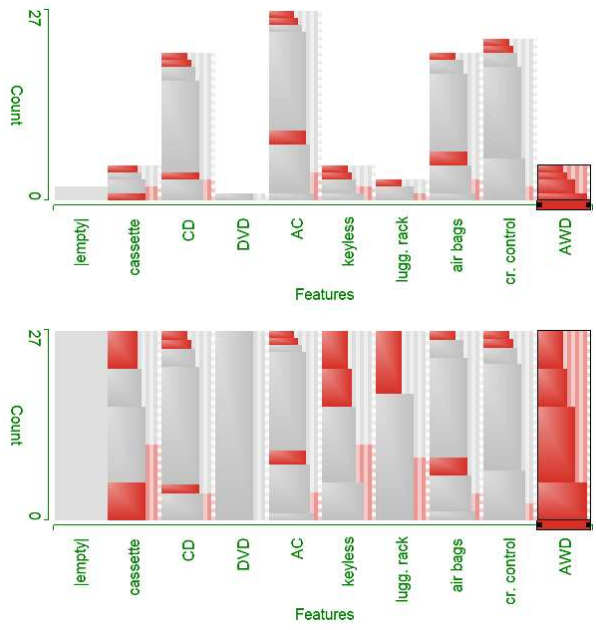

Fig. 5. The upper image shows a set'o'gram of the Columbus dataset. After brushing a bar in a set'o'gram we can see the brushed amount of data for each block and each bar in the view. As expected, tall bars like "AC" have a larger brushed area than short bars, like "keyless". In the lower image, the view is switched to relative mode, and each bar is scaled to the same height. This view shows, that a relatively large amount of the bar "keyless" has been brushed.

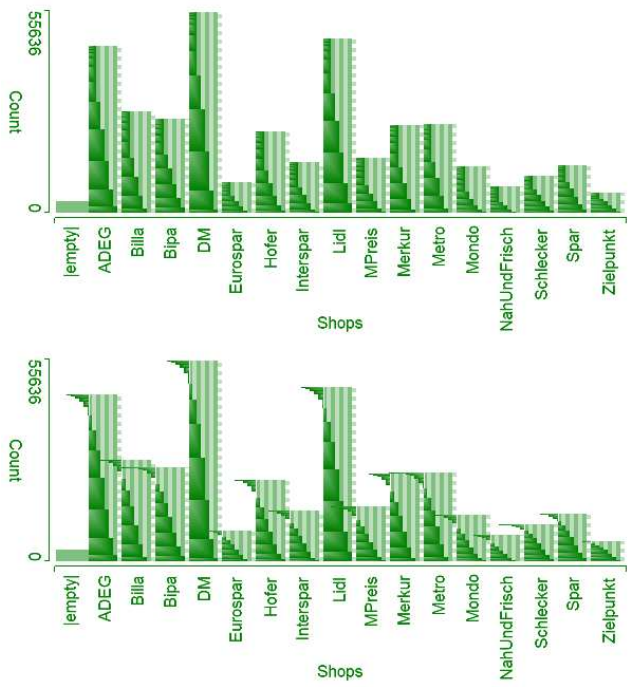

Fig. 6. The upper picture shows a set'o'gram displaying peoples' shop preferences. Unfortunately, the bars become too thin to give the upper blocks their correct size. Therefore, the minimum width is set to at least one whole stripe of the bar. The lower picture shows what happens, if we do not check the blocks' widths. The right edges of the upper blocks are moved to the left and are no longer inside the bars. While this is considered an artefact, it contains more information than the regular visualization.

stitution, which results in a huge amount of dimensions that are hard to visualize. By grouping coherent dimensions to set-typed dimensions, we can reduce their number and make them easier to visualize, too. Table 1 shows a summary on all set-typed dimensions that have been created. 65 categorical dimensions have been converted into ten set-typed dimensions which results in a reduction of 55 dimensions. Although all set elements are stored as strings, while the categorical values were only "0" and "1", the size of the dataset has shrunk from 23 MB to 18 MB.

Shops When looking at a set'o'gram containing information on shop preferences (see figure 6, we can immediately gain much information. Generally, there are only few people that selected one or two shops. Most people purchase items at different shops, probably comparing price and quality. For example, the discounter "Hofer" has the highest number of customers in our dataset, but they only have a very small amount of "exclusive" customers. In contrast to that, "Billa" and "Spar" have a much larger number of customers who buy their everyday necessities only in that particular shop. However, even there the number of exclusive customers is quite low. As a consequence, we could reason, that cheap special offers are a bad method to attract customers, because they will probably not buy the more expensive products. Maintaining fair prices for a wide range of products is probably a better approach.

Set	Description	Valid Values
Bildung	Educational Institutions	9
Technik	Technical Devices	8
Geschaeft	Frequently visited Shops	16
Hunde_TR	Dry Dog Food Brands	5
Hunde_NA	Canned Dog Food Brands	5
Katzen_TR	Dry Cat Food Brands	5
Katzen_NA	Canned Cat Food Brands	5
WaschmittelArt	Kinds of Detergent	4
TReiniger	Cleaning Utilities Brands	4
Binden	Sanitary Napkins	4

Table 1. 65 categorical columns are reduced to ten set-typed dimensions. The number of dimensions in the whole dataset has been reduced from 105 to 50. This is a good example for efficient dimension reduction.

Washing Agents Figure 7 shows two set'o'grams displaying customers preferences on types of washing agents. One fact is quite obvious when looking at it: people tend to stick with their regular type of washing agent, because in each block, the full width bar is the highest one. Only customers of liquid washing agents (third bar) tend to use other agents too. A possible consequence for marketing departments could be, that liquid washing agents should not be advertised as a sole solution, but as an addition to other types. Although the view gives no information on preferred brands, we could also suppose, that it may be more likely for customers to switch brands than to switch agent types. If new types of washing agents are produced, the production of conventional ones should not be discontinued, because customers might rather change to another brand than to the new type of washing agent.

Computers & Internet Access In figure 8 we tried to reveal connections between education, computer usage and the number of children in a household. Especially offline users are statistically different to overall computer and internet users. While computer and internet usage correlates with education, there is a noticable difference between households with children and those without. However, the correlation with education disappears when looking at computer users without internet access. While internet access is most commonly found in families with one child, computers without internet access is comon in families with four children.

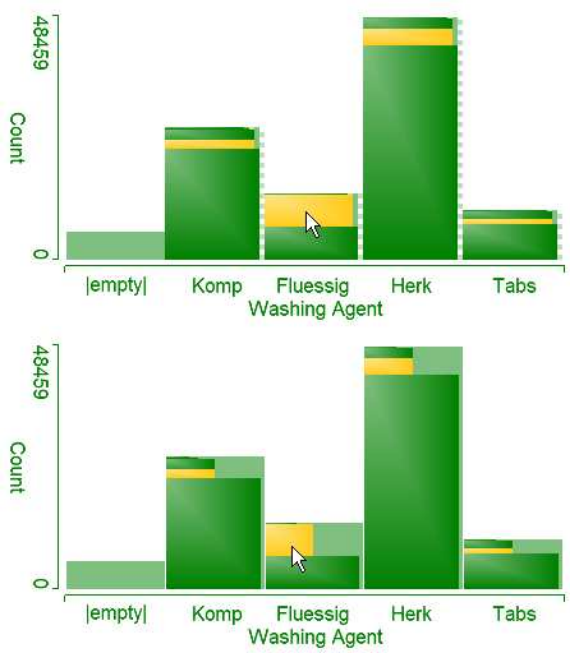

Fig. 7. Two set'o'grams showing which types of washing agents customers prefer. Both views represent the same data, the upper one is drawn subtracting fixed values from the bars' widths. In the lower one, the bars' widths are divided by the number of elements in the sets they represent. The first mode is used throughout this paper and is best suited for a high number of elements. The second mode puts a stronger emphasis on area preservation and shows the amount of full width blocks even more obvious.

Data Cleaning Datasets can, especially when the data was provided by many people, contain errors. People may overlook some checkboxes, they can misunderstand questions, refuse to give information on specific topics or even provide wrong information intentionally. In some cases, wrong information can be detected or even corrected.

A set'o'gram of all educational levels can be seen in figure 9. The educational levels are roughly sorted from the lowest to the highest. The first block in the bar representing university graduates shows, that about 40% of all university graduates did not check any other educational institutions. While we can assume, that these people have primary education, we do not know, which type of high school they graduated from. In such a case, missing primary education could be corrected, but missing secondary education can only be detected. It seems that many people just check the highest type of education they have, assuming that their educational development is not of any interest.

Figure 10 shows histograms depicting dog owners and dog food customers. Three persons indicate that they buy all kinds of dog food without even owning a dog. Although it is not necessary to own a dog in order to feed one, these data record can be discarded because it is very likely that the information provided by the user is not true. In fact, further research showed, that one of these persons also pretends to be a customer of all available shops and has reached all educational levels.

7 Discussion

This section deals with some aspects that came up during the design and implementation of visualization methods for set-typed data. Implementation took place in ComVis, a coordinated multiple view application featuring different data types and visualization techniques. A screenshot of a workplace featuring set'o'grams, histograms, scatterplots and parallel coordinates views is shown in figure 13.

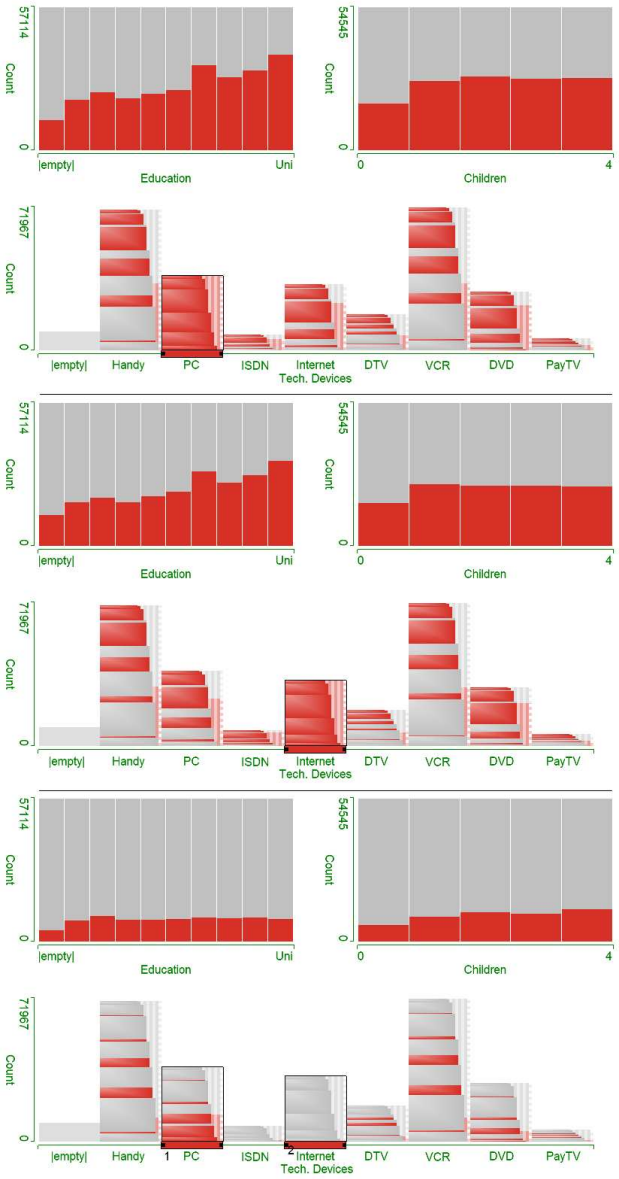

Fig. 8. All three sections show the same views: a histogram displaying relative brushed amounts of all educational levels, another one depicting the number of children, and a set'o'gram displaying technical devices. In the first section, personal computers have been brushed. The education histogram shows, that people with higher education are more likely to own a computer. The other histogram shows, that households with children are more likely to own a computer, than those without. The highest amount of computer users lives in households with two children, however, these differences are very slight. In the second section, internet users have been brushed. The upper left histogram looks very similar. The brushed sections are a bit lower, but we can still observe, that internet usage increases with higher education. The upper right histogram tells us that households with one child are most likely to have internet access. The relative amount of internet users decreases very slightly in families with more children, but childless households are still the most unlikely group for internet access. In the third section, "offline" computer users have been brushed by first brushing computer users, and then subtracting internet users. Surprisingly, the education's influence seems to disappear – all educational levels have about the same relative amount of offline computer users (with the empty set as an exception). As another surprise, the highest relative amount of computer without internet access can be found in families with four or more children.

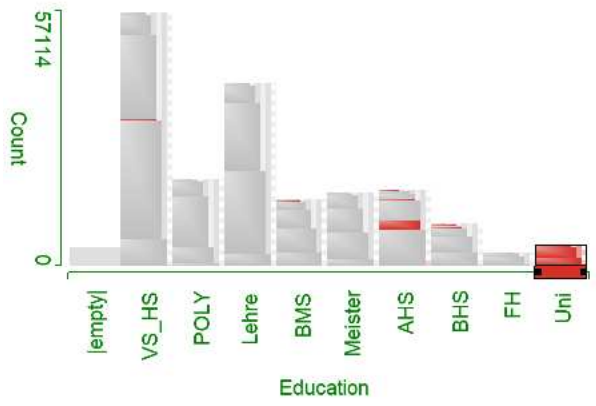

Fig. 9. A set'o'gram of educational levels reveals some errors in the dataset. The full width blocks should not exist in some bars, because primary education is a prerequisite for secondary education which is again mandatory for university students (rightmost bar). Most university graduates did not indicate primary education, though.

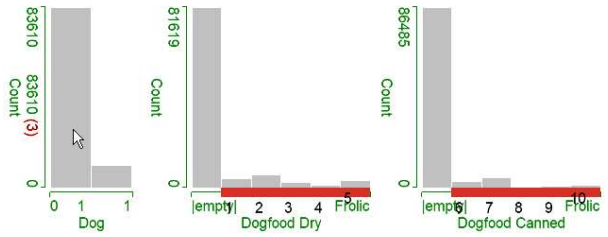

Fig. 10. The two histograms on the right show the distribution of various brands of dry and canned dog food. Each single brand has been brushed, and all these brushes have been combined using "and" operators yielding people who buy – or pretend to buy – all brands of dog food. The histogram on the left shows dog owners on the right and people without dogs on the left. The mouse cursor is positioned over the left bar, and the label at the left axis shows that three people pretend to buy all kinds of dog food without having a dog.

Apart from histograms, we also extended 2D scatterplots (figure 11) to handle sets. If sets are visualized in a scatterplot, all available elements are placed in equal intervals along an axis. Of course it is possible to use sets for both dimensions. This is very similar to visualizing categorical data, the only difference is, that the first element on a set-axis is the empty set and that individual data items are (usually) represented by multiple points (similar to the replication effect in histograms) Scatterplots are very useful to find correlations between two dimensions. If two set-typed dimensions are visualized, there are lots of congruent points which makes analysis harder. We found it especially useful to investigate the relation of one set-typed dimension with a second dimension of other type.

The parallel coordinates view (figure 12) is very useful, if multiple dimensions have to be displayed at one time. Each data item is represented by a line strip intersecting all axes at positions, which represent the actual values of the data item in each of the depicted dimensions. Because parallel coordinates are also an item-based visualization, the main issue of sets is, again, that one data item can contain multiple elements. Just as there are multiple points in scatterplots for single data items, there are multiple lines for single data items also in parallel coordinates. A line representing a data item with multiple elements in one dimension has to intersect the set-typed axis at multiple positions. Therefore, the line has to be split up at the previous axis (if there is one) and joined afterwards. If two set-typed axes are placed next to each other, each assigned element of the first axis has to be connected with the ones on the second axis. Because the number of line segments next to set axes can increase significantly it is possible to divide the opacity of a line segment by the number of segments belonging to this data item. Similar to the situation with scatterplots, we found it most useful to work with visualization configurations without two set-typed dimensions next to each other.

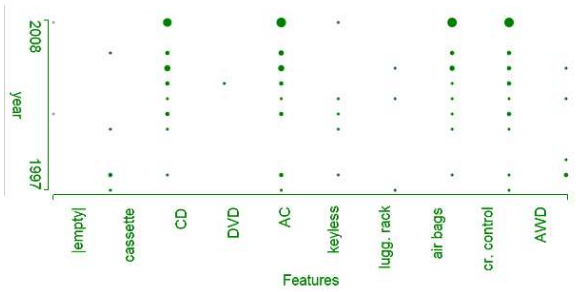

Fig. 11. The 2D scatterplot can be adapted for set-typed data, if positions on the axis are assigned to elements. Sets can contain multiple elements, leading to multiple points in the scatterplot (even for one single data item).

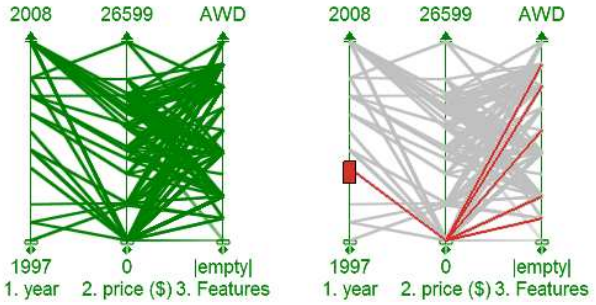

Fig. 12. This parallel coordinates view shows three dimensions of the Columbus dataset. The first two dimensions are categorical and the third one is set-typed. If a data record contains multiple elements in its set, the line representing this record has to split up and intersect the set-typed axis at multiple positions. This can be anticipated because of the higher density of lines between the second and the third axis, and it becomes obvious in the second image: here, only one data record (i.e. one car) has been selected - and multiple lines connect the available features of this vehicle.

8 Summary and Conclusions

Visualization research has progressed to a nicely mature level in several parts of the overall scientific challenge (of visualization) and many good solutions are available these days. Still there are problems which not yet have been addressed enough and the native visualization of non-standard data types still is a challenging and in large part unsolved problem. Enabling state-of-the-art approaches to interactive visual analysis for set-typed data is rewarding as advanced information drill-down into complex information spaces becomes possible even with limited screen size and dimensionality. At the (relatively small) "cost" of some unconventional visualization characteristics – in our case, for example, the fact that individual data items show up multiple times in the visualization – simplified procedures for advanced visual analysis can be gained. We consider this promising, also with respect to other, non-standard data types, such as complex numbers or tensors, for examples. This paper demonstrates, in our eyes, that moderate adaptations of proven, existing visualization techniques can lead to a more native and more effective visual analysis.

References

[1] S. K. Card, J. D. Mackinlay, and B. Shneiderman. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999.

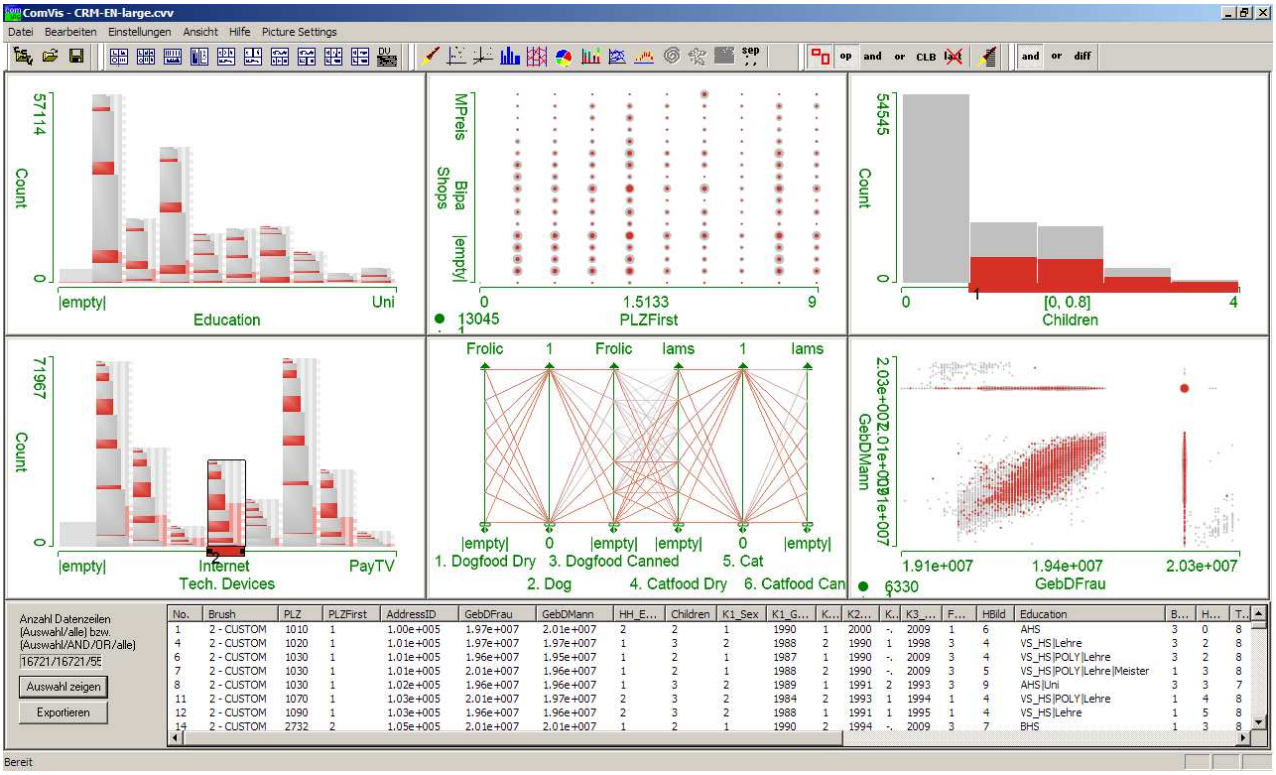

Fig. 13. A session of the ComVis visualization tool featuring different visualization techniques.

[2] M. Friendly. Mosaic Displays for Multi-Way Contingency Tables. *Journal of the American Statistical Association*, 89(425):190–200, 1994.

[3] H. Hofmann. Exploring categorical data: Interactive mosaic plots. *Metrika*, 51(1):11–26, 2000.

[4] A. Inselberg and B. Dimsdale. Parallel coordinates: a tool for visualizing multi-dimensional geometry. In *Visualization, 1990. Visualization '90., Proceedings of the First IEEE Conference on*, pages 361–378, San Francisco, CA, Oct. 1990.

[5] B. Johnson and B. Shneiderman. Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In *Visualization, 1991. Visualization '91, Proceedings., IEEE Conference on*, pages 284–291, San Diego, CA, Oct. 1991.

[6] D. Keim, M. C. Hao, J. Ladisch, M. Hsu, and U. Dayal. Pixel bar charts: a new technique for visualizing large multi-attribute data sets without aggregation. *2001. INFOVIS 2001. IEEE Symposium on Information Visualization*, pages 113–120, 2001.

[7] D. A. Keim, M. C. Hao, and U. Dayal. Hierarchical pixel bar charts. *Transactions on Visualization and Computer Graphics*, 8(3):255–269, July/Sept. 2002.

[8] R. Kosara, F. Bendix, and H. Hauser. Parallel sets: interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(4):558–568, July/Aug. 2006.

[9] J. C. Roberts. State of the Art: Coordinated & Multiple Views in Exploratory Visualization. In G. Andrienko, J. C. Roberts, and C. Weaver, editors, *Proceedings of the 5th International Conference on Coordinated & Multiple Views in Exploratory Visualization (CMV2007)*. IEEE Computer Society Press, July 2007.

[10] G. E. Rosario, E. A. Rundensteiner, D. C. Brown, M. O. Ward, and S. Huang. Mapping nominal values to numbers for effective visualization. *Information Visualization*, 3(2):80–95, 2004.

[11] R. Spence and L. Tweedie. The attribute explorer: information synthesis via exploration. *Interacting with Computers*, 11(2):137 – 146, 1998.

[12] J. J. van Wijk and H. van de Wetering. Cushion treemaps: Visualization of hierarchical information. In *INFOVIS*, pages 73–78, 1999.