The Cartographic Journal *Vol. 42 No. 1* *pp. 49–60* *June 2005*

REFEREED PAPER

Blending Aggregation and Selection: Adapting Parallel Coordinates for the Visualization of Large Datasets

Gennady Andrienko and Natalia Andrienko

Fraunhofer Institute AIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany;
Website: http://www.ais.fraunhofer.de/and
Email: gennady.andrienko@ais.fraunhofer.de

Many of the traditional data visualization techniques, which proved to be supportive for exploratory analysis of datasets of moderate sizes, fail to fulfil their function when applied to large datasets. There are two approaches to coping with large amounts of data: data selection, when only a portion of data is displayed, and data aggregation, i.e. grouping data items and considering the groups instead of the original data. None of these approaches alone suits the needs of exploratory data analysis, which requires consideration of data on all levels: overall (considering a dataset as a whole), intermediate (viewing and comparing collective characteristics of arbitrary data subsets, or classes), and elementary (accessing individual data items). Therefore, it is necessary to combine these approaches, i.e. build a tool showing the whole set and arbitrarily defined subsets (object classes) in an aggregated way and superimposing this with a representation of arbitrarily selected individual data items.

We have achieved such a combination of approaches by modifying the technique of parallel coordinate plot. These modifications are described and analysed in the paper.

1. INTRODUCTION

Large datasets pose a serious challenge to researchers in visualization. Techniques that proved to be effective in supporting the exploration of moderate amounts of data rapidly decline in their efficacy and, eventually, completely fail with increasing the number of data items to be analysed. Attempts to visualize large datasets in the same ways as small ones typically encounter at least one of the following obstacles:

1. Overplotting: visual elements representing different data items fit into the same position in the display, and, hence, some of them are occluded. This problem occurs to such visualization techniques as scatterplot, scatterplot matrix and parallel coordinate plot. To fight overplotting, transparent output is used in some implementations (see, for example, [T02]). While this approach may be useful for detecting clear-cut groupings of objects with close characteristics, it does not work so well when the distribution of characteristics is more dispersed. Besides, the user cannot properly estimate the number of data items fitting in this or that position within the plot area.
2. Decline in legibility: in order to represent all data items in one display, the corresponding visual elements are made so small that they become hardly visible and distinguishable. This problem often occurs on map displays.
3. Impossibility to represent the full dataset: visual elements representing all data items cannot be fitted into one display due to size limitations and, hence, only a part of the data can be seen. This problem arises, for example, in the display techniques representing data in a tabular form, including table lens and permutation matrix.

Two major approaches to solving these problems exist: data selection (zooming, focusing, filtering) and data aggregation. Data selection means that a display does not represent a dataset as a whole but only a portion of it, which is selected in a certain way. The display is supplied with interactive controls for changing the current selection, which results in showing another portion of the data. Thus, selection on a map is done through zooming and viewport shifting while rows and columns in a table display are selected for viewing using scrollbars. A data portion may also be selected by means of querying: only data items with certain properties, which are specified in a query, are shown. This technique, in particular, helps to reduce overplotting.

Data aggregation reduces the amount of data under analysis by grouping individual items into subsets, often called ‘aggregates’, and computing some collective characteristics of the aggregates. The aggregates and their characteristics (jointly called ‘aggregated data’) are then explored instead of the original data.

We argue that none of these approaches alone fully satisfies the needs of exploratory data analysis and that it is

DOI: 10.1179/000870405X57284

necessary to combine them. In order to substantiate our opinion, we need to refer to the theory of 'reading levels' formulated by Jacques Bertin [B83]. Let us briefly recite this theory.

The function of graphical representation of data is to provide answers to various questions concerning the data. The questions can be distinguished according to their 'levels of reading'. The level of reading indicates whether a question concerns a single data element, a group of elements taken as a whole, or all elements of a dataset. Accordingly, there are three *levels of reading*, elementary, intermediate and overall.

Bertin claims that a visualization is effective if it permits immediate extraction of the necessary information, i.e. finding the answer to the observer's question at a single glance, with no need to move the eyes or attention and to involve the memory. Bertin uses the term 'image' to refer to 'the meaningful visual form, perceptible in the minimum instant of vision'. An optimal visualization contains a single image providing the answer to the observer's question. However, exploratory data analysis (in Bertin's terms, 'information processing') typically does not deal with a single question, for which a data representation could be optimized. 'With this function *(i.e. supporting information processing)*, the graphic is an experimental instrument leading to the construction of collections of comparable images with which the researcher 'plays'. We class and order these images in different ways, grouping similar ones, constructing ordered images to discover the synthetic schema which is at once the simplest and most meaningful'. That is, the ultimate goal of information processing is to discover a simple and meaningful synthetic schema standing behind the data, i.e. to understand the phenomenon characterized by the data in its whole. This, evidently, requires the overall level of reading. However, on the way to this understanding, an explorer needs to classify and order, detect similarities and differences, with the intermediate and elementary reading levels being necessarily involved in these activities.

Hence, in order to be suitable for exploratory data analysis, a visualization needs to support all reading levels. For this purpose it is required that the visualization, on the one hand, is comprehensive, on the other hand, contains the smallest possible number of memorizable images. Comprehensiveness means avoiding any prior reduction of the information (e.g. by classification), using the 'complete information, which alone provides all the givens for pertinent correlations and choices […] But also it matters that all types of comparisons and classings are possible and easy. The most useful questions will obviously involve the overall level of reading, where their answer will be found in a limited number of comparable images' [B83, p. 164].

Let us evaluate from this perspective the two approaches to the visualization of large data volumes, that is, selection and aggregation.

Selection significantly reduces the information that can be perceived at once, since only a portion of data is visible at any moment. However, interactive controls allow the user to alter the current selection, so that any individual data item can be eventually accessed. Therefore, it is possible to find answers to questions requiring the elementary level of reading, i.e. questions about individual items. The user can also explore groups of simultaneously visible data items, which means that the intermediate reading level is also partly supported. We say 'partly' because it is difficult to compare a currently visible group to another group or to the whole dataset. The overall reading level is not supported because there is no way to see all data items simultaneously.

Aggregation groups the original multitude of data items into a significantly smaller number of aggregates, which can be visualized all together in a single view thus enabling the overall reading level. Characteristics of the aggregates can be easily explored and compared; hence, the intermediate reading level is also supported. However, the grouping of data items into aggregates is done prior to the visualization, and each aggregate is typically represented by one graphical element, such as a bar in a histogram or a sector of a piechart. As a result, the user cannot consider arbitrary subsets of data items that do not coincide with the aggregates displayed. Since dealing with rigid, once and forever defined classes or groups is incompatible with the philosophy of exploratory data analysis, contemporary data aggregation tools are characterized by high user interactivity, which allows an analyst to choose and dynamically change the level of aggregation (i.e. how large the aggregates are), the method of aggregation (i.e. how individual items are grouped into aggregates), and the functions for deriving characteristics of the aggregates from those of their members (i.e. sums, ranges, or various statistics). A good example of such a tool is Treemap developed by the research team of Ben Shneiderman [Sh92].

While substitution of original data by aggregated facilitates the processes of simplification and abstraction and, hence, gaining an overall understanding of a phenomenon, this is achieved at the cost of substantial information loss, specifically, discarding individual data items. In Bertin's terms, an aggregated data display is not comprehensive, and the elementary reading level is completely disabled.

Hence, neither data selection nor data aggregation provides a fully satisfactory solution to the problem of visualizing large datasets. A suitable compromise could be achieved by means of combining the approaches. Thus, a display could represent summary information concerning the entire dataset and its subsets (aggregation) and, at the same time, individual characteristics of selected data items (selection). This allows the user, in particular, to compare these individual characteristics with those of the whole set and the subsets. The display must be supplied with interactive facilities allowing the user both to re-aggregate the data in different ways and to change the current selection of individual data items.

For a practical realization of this idea, two ways are possible:

1. Take some display technique representing individual data items as a basis and modify it so that it could also show aggregated information.
2. Take some aggregated representation technique and extend it with a possibility to display individual characteristics.

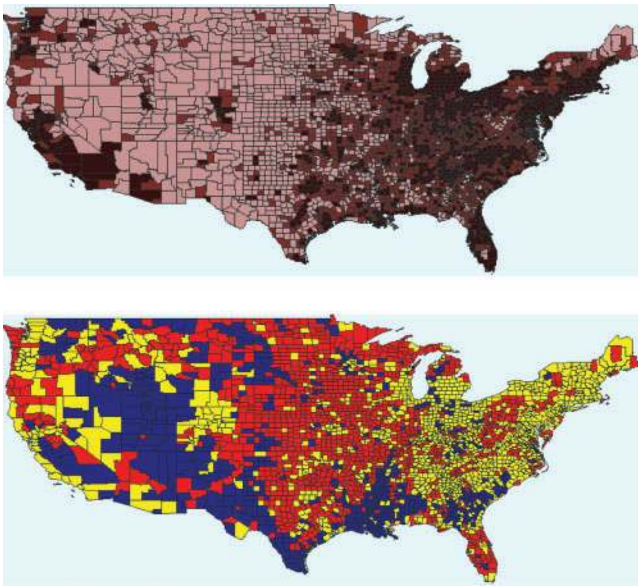


Figure 1. Two example classifications of the counties of the USA

In this paper, we investigate the possibilities for extending some popular data visualization techniques (scatterplot and matrix of scatterplots, table lens, frequency histogram, box plot, and parallel coordinate plot) to combined representation of aggregated and individual information. Then, we describe in more detail our own extensions of the parallel coordinate plot technique.

We do not introduce any restrictions or assumptions concerning how aggregates are defined except that each data item must belong to at most one aggregate. This means that there should be a general mechanism for reflecting arbitrary aggregations, or classifications, and dynamic reaction to changes of the aggregates (classes), for example, like the mechanism for class propagation in the system CommonGIS [AA03].

As an example, Figure 1 shows two different classifications of the counties of USA. On the upper map, the counties are divided into three classes according to the population density in 1999. The variation of colour is used to distinguish the classes on the map: light corresponds to low density and dark to high density. The breaks between the intervals of low, medium, and high population density have been selected so that the resulting classes consist of approximately equal numbers of counties.

The lower map represents the results of grouping the counties into three clusters on the basis of six attributes representing proportions of different age groups in the population: below five years old, 5–17, 18–29, 30–49, 50–64, 65 or more years. The clustering was done using the method SimpleKMeans as it is implemented in the data mining system Weka [WF99].

The set of counties with their demographic characteristics and these two example classifications will be used throughout the paper for testing various display techniques. The set is rather large: it consists of 3140 objects. While this is sufficient for our study, we take into account that much larger sets exist and, hence, any technique should be evaluated in terms of the possibility to scale it to larger sets.

The model analysis task is to compare the classes of countries with respect to the age structure of the population. For the first classification, this would allow one to see whether the age structure is somehow related to population density. For the second classification, this could help an analyst to understand the results of clustering, i.e. what the groups defined by the data mining algorithm mean.

Let us now evaluate the most popular techniques for data visualization in terms of their suitability for large datasets and the possibility to modify them so that questions of all three reading levels could be answered. Our particular focus is the possibility of comparison of arbitrary classes of objects (data items), an activity involving the intermediate reading level, although we also pay attention to the other two levels. We assume that classes are defined independently of the displays used to represent them, and only information about the classification results (specifically, what objects each class consists of and what colour is assigned to it) is transferred to each display.

2. CLASS COMPARISON WITH DIFFERENT DATA DISPLAYS

We include the following types of data displays in our evaluation:

- Scatterplot (suitable for two attributes) and scatterplot matrix (suitable for more than two attributes);
- Table lens [RC94];
- Parallel coordinate plot ([I85], [I98]);
- Frequency histogram (suitable for a single attribute) and a group of coordinated histograms representing different attributes ([TS98]);
- Box-and-whiskers plot, or, shorter, box plot [T77].

A common feature for the first three display types is that they represent individual characteristics of objects whereas the remaining two techniques provide only general (aggregate) information about the distribution of attribute values throughout the set of objects.

There may be two basic approaches to representing arbitrary data subsets (classes) on different types of graphical displays: either represent all classes within a common display area or clone a display as many times as there are classes and show each class on a separate display copy. Both approaches have their pluses and minuses. Display multiplication eliminates overlapping of information pertinent to different classes and is therefore more beneficial for an individual consideration of each class. One can also easily compare general patterns of distribution of characteristics in the classes. However, comparison of attribute values is more complicated than in the case when all classes are represented on one and the same display. Another disadvantage of display multiplication is that it uses much more screen space. This problem becomes especially serious when many attributes are involved in analysis. Such display techniques as scatterplot matrix, coordinated histograms and box plots already include multiple plots or charts, and further multiplication may cause significant difficulties for analysis. Thus, if there are N attributes and M classes, the user will have to analyse and compare N × M displays.

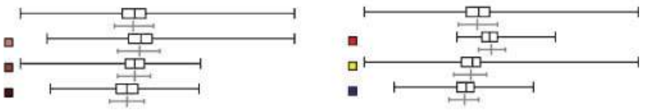


Figure 2. Representation of class characteristics by box plots. The group of box plots on the left corresponds to the classification of the counties of the USA according to the population density. On the right, the results of clustering are reflected. All box plots show the distribution of values of the attribute 'Proportion of age group 50-64 years'. The topmost box plots correspond to the entire set of counties, and the plots below represent the classes. In addition, mean values and standard deviations for the respective object sets are shown below each box plot

From now on, we shall mostly focus on the approach with showing classes on the same display. Typically, display elements representing different classes are differently coloured, which allows one to distinguish between the classes. This approach is applicable to all the techniques under consideration except for the box plot.

Figures 2 to 6 demonstrate how object classes can be reflected on the different types of displays by the example of the classifications of counties of the USA shown in Figure 1.

As we have mentioned, a box plot display can represent classes only by means of display multiplication (Figure 2). The visualization supports analysis on the overall (entire set of objects) and intermediate (object classes) levels, but cannot represent individual characteristics of selected objects. Other disadvantages are too coarse aggregation and the necessity of using multiple plots in order to represent several attributes.

Figure 3 demonstrates representation of classes on a histogram. One can see the overall pattern of value distribution in the entire set of objects and compare it to the patterns for the classes. There are several bottlenecks in this visualization:

- The shape of a histogram depends on the chosen granularity.
- The technique supports well the comparison of one class (aligned to the baseline at the bottom) to the whole set. Making comparisons between classes is difficult.
- When working with large sets of objects, it is necessary to zoom the histogram for seeing details (for example, on the right and left ends of ranges of normally-distributed attributes). However, zooming destroys the overall view.
- Access to individual data instances is practically impossible.

In Figure 4, one can see how classes can be represented on a scatterplot. The problem with this display is overplotting: many points (probably, of different colour) may overlap. Therefore it is impossible to guarantee that the pattern perceived from such a display (if any) is correct.

Representation of classes on a table lens display can be done by means of colouring table rows according to the

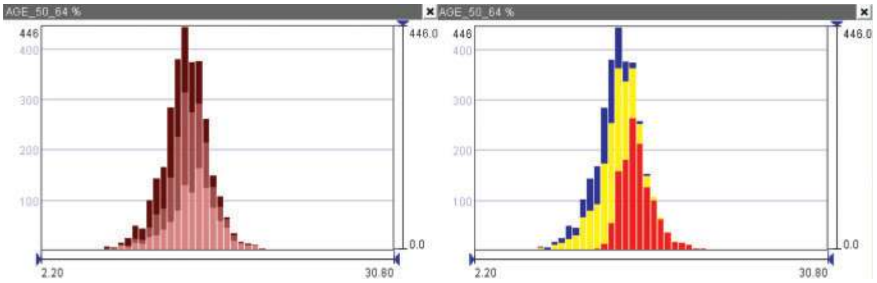


Figure 3. Histogram representation of the attribute 'Proportion of age group 50-64 years' built with the granularity of 50 bins. The bars are divided into coloured segments according to the number of objects belonging to each of the classes. On the left, the classification according to population density is represented, on the right — the results of clustering

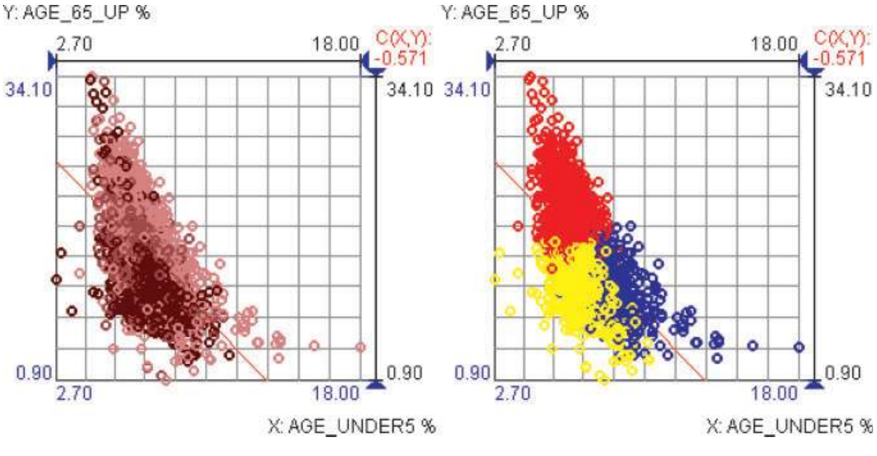


Figure 4. Scatter-plots of the attributes 'Proportion of age group less than 5 years old' and 'Proportion of age group 65 years and more'. The points are coloured according to the classes by population density (left) and to the clusters resulting from the cluster analysis (right)

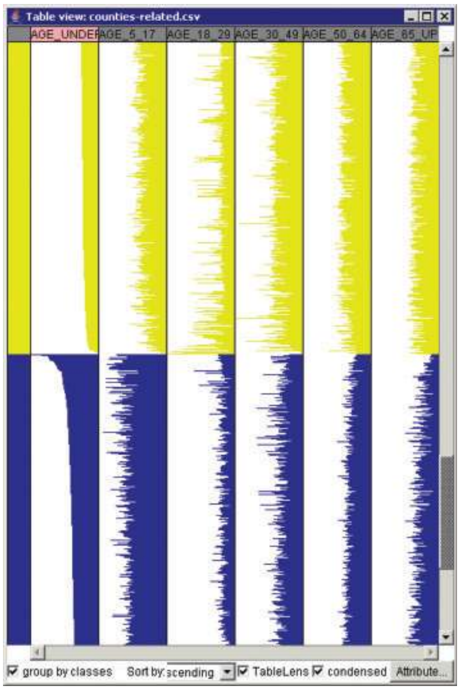


Table view: counties-related.csv

☑ group by classes Sort by: scending ☑ TableLens ☑ condensed

Figure 5. Table lens representation with grouping of rows by classes (results of cluster analysis). Only a part of data is visible due to the limitations of the screen size

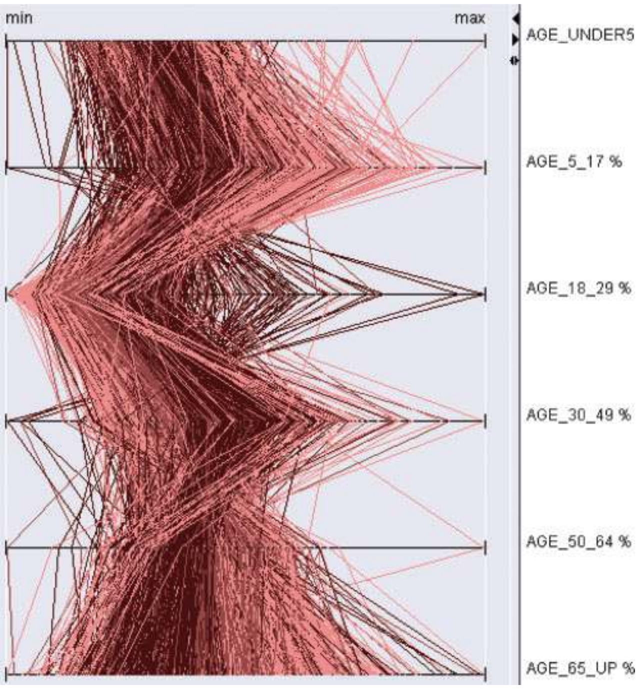


Figure 6. A parallel coordinate plot with the lines coloured according to the classes defined on the basis of population density

classes the respective objects belong to. Rows corresponding to the same class can be grouped together (Figure 5). Within a group, the rows can be sorted according to values of some attribute. However, as we have already mentioned, the technique of table lens is inappropriate for large object sets.

In a parallel coordinate plot (Figure 6), objects are represented by polygonal lines connecting points ('coordinates') on parallel axes, which represent attributes. The lines can be painted according to class colours. In simple cases, this visualization may be helpful for forming a general view of distribution of characteristics within an object subset (class). However, like with the scatterplot, a great problem of this display is overplotting, which in most cases prevents perceiving properties of classes. This is illustrated in Figure 6. Even if some pixels are painted in a colour of a class, this does not guarantee that there are no lines corresponding to objects from other classes passing through the same point but not visible due to the high density of lines.

As it follows from our analysis, none of the visualization techniques properly supports the exploratory analysis of large datasets on all reading levels: overall (the whole dataset), intermediate (arbitrary subsets, or classes), and elementary (individual data items). Hence, there is a need in designing a new technique, for example, by amending one of the existing techniques. As we have discussed, a promising approach is combining data aggregation with data selection, i.e. aggregated representation of the entire set and subsets (classes) with a superimposed display of individual characteristics of user-selected objects. We have indicated two practical approaches to achieve this: either to modify some display technique representing individual data items so that it could also show aggregated information or to extend some aggregated representation technique with a possibility to display individual characteristics. The former approach seems more promising than the latter one. At least, some modifications of scatterplot and parallel coordinate plot techniques towards representing aggregated data already exist (while we could not find a proper way to extend the table lens technique). At the same time, it seems quite difficult to find a way to extend histogram and box plot displays for including individual data.

The modifications of scatterplot are known as binned scatterplot or two-variable histogram (see, for example, [C91] or [W99]). They exploit the idea of binning — dividing the plot area into small cells (bins) and counting the number of data instances fitting into each bin. The counts are then represented by painting the cells into different colours or shades of grey. There is a possibility to represent selected individual instances by symbols (e.g. small hollow circles) superimposed on such a representation, although we did not find in the literature any mentioning of this being actually done. A disadvantage of binning is that it excludes the possibility to represent several classes within a single display: it is difficult to show in a legible way how many items of each class fit into each of the bins. Hence, only display multiplication may be used for this purpose. However, in a case of more than two attributes, multiple scatterplots are needed, and their multiplication will tremendously complicate the overall visualization.

The parallel coordinate plot (PCP) technique seems to be a more suitable candidate for an extension towards the representation of aggregated information. Various attempts to include aggregated information in a parallel coordinate plot have been made by a number of researchers. In next section, we briefly consider their suggestions.

3. ENHANCING PARALLEL COORDINATES

A. Inselberg introduced the parallel coordinate plot in the 1980s as a geometrical abstraction for presenting selected projections of a multidimensional attribute space (see http://www.cs.tau.ac.il/~aiisreal/— 'Home of Parallel Coordinates' — for the history of PCP). At about the same time, D. Schilling introduced a similar to PCP 'value path' technique for multi-criteria optimization problems [SRC83]. In early papers, various mathematical and algorithmic properties of PCP were studied ([I85], [ICR87]). Later, A. Inselberg and E. Wegman in parallel considered PCP as a tool for exploratory data analysis ([I90], [I98], [W90], [MW91]).

It is necessary to stress that the presence of N axes for N attributes does not mean that the plot fully reflects the N-dimensional space without loosing important information. Actually, the plot displays values of N attributes in (N-1) pairwise projections.

The technique of PCP attracted attention of many researchers, who suggested various extensions and modifications. A complete overview of the related work is beyond the scope of this paper. We shall only consider the suggestions that introduce display of aggregated data into PCP and thereby make it more appropriate for large datasets. This includes drawing coloured bands along the axes to represent the attribute ranges for the whole data set and for classes ([S00]), putting box plots on the axes ([T02]), and even drawing histograms along axes ([OL96], [PT03]). Such enhancements can be called 'parallel box plots' and 'parallel histograms' [PT03]. These additions are very useful for the understanding of properties of subsets and their comparison. However, such additional graphics inside a PCP overlap with lines, which complicates the perception. Besides, each graph represents only a single attribute ignoring its relationships to other attributes.

Another way of introducing summary information is showing average or median lines for the whole set and for classes ([S00]). This is done by connecting positions on neighbouring axes corresponding to the mean or median values of the respective attributes. However, since the mean or median values are counted for each attribute independently, one needs to be cautious and avoid interpreting the resulting lines as the most typical profiles (i.e. combinations of characteristics) for the sets of objects.

In [MW91], it is proposed to draw a 'line density plot' instead of individual lines. For this purpose, the area between axes is divided into zones (pixels), and the number of lines passing through each zone is counted. The so obtained counts are normalized and shown by painting the zones into different colours or colour grades. This technique, which is analogous to 'binning' on a scatterplot, supports quite well the overall analysis level and is suitable for large and very large datasets. Although this was not suggested in [MW91], it can be easily imagined how to superimpose drawing of selected individual lines upon the density-based background painting of the plot area. However, this technique does not allow us to represent object classes on the same display. Hence, the intermediate analysis level can only be supported by means of display multiplication.

Drawing bands, or envelopes, around selected subsets of lines provides yet another variety of aggregation. Extending PCP by showing envelopes of subsets of lines was proposed already as early as in [I85] and [ICR87]. Later, A.Inselberg proposed an iterative procedure of visual data mining [I98] that consists of sequential narrowing of envelopes by means of selecting subintervals of attribute values. In this procedure, subsets are defined by means of interaction with the PCP display. It is not relevant to investigation of arbitrary classes.

A combination of hierarchical clustering with PCP is proposed in [FWR99]. The authors represent clusters of objects by variable-width opacity bands rather than individual lines. Only cluster centres are shown on top of the bands by lines. Several clusters can be shown and analysed simultaneously. The outlines of the bands correspond to Inselberg's envelopes, and the opacity gradually decreased from cluster centres to the boundaries without taking into account line densities.

A similar approach was proposed in [HC00], where PCP was enhanced by rule induction methods. Discovered rules are visualized using semi-transparent overlapping bands.

This method was further developed in [BH03] where a fuzzy membership function for object subsets resulting from hierarchical clustering was represented by varying opacity. The authors introduced special drawing hints to be used instead of alpha-blending — a rather slow technique of graphical output typically used for transparent drawing and painting on the computer screen.

Building envelopes around lines representing selected subsets of objects is a convenient tool for comparing characteristics of a subset (specifically, ranges of attribute values) to those of the whole data set. However, the applicability of this method for comparison of subsets is limited because overlapping of their envelopes complicates the analysis (see Figure 7).

As a conclusion from this overview, we find the following PCP modifications to be supportive for exploration of properties of object classes:

1. Replacing individual lines on a PCP by class envelopes.
2. Putting additional graphics on axes to represent the distribution of attribute values in the whole data set and in subsets.

These ideas were used in our own implementation of PCP-based technique for class investigation described in the next section.

4. OUR APPROACH

In our implementation, it is possible to replace drawing of individual lines by either class envelopes or special graphics on the axes representing summary information about the distribution of attribute values in the whole object set and in different classes. Lines for user-selected objects can be overlaid on such an aggregated representation.

Let us now consider both aggregation modes in more detail, starting with class envelopes. Actually, envelopes in their 'pure' form are not especially informative: they only

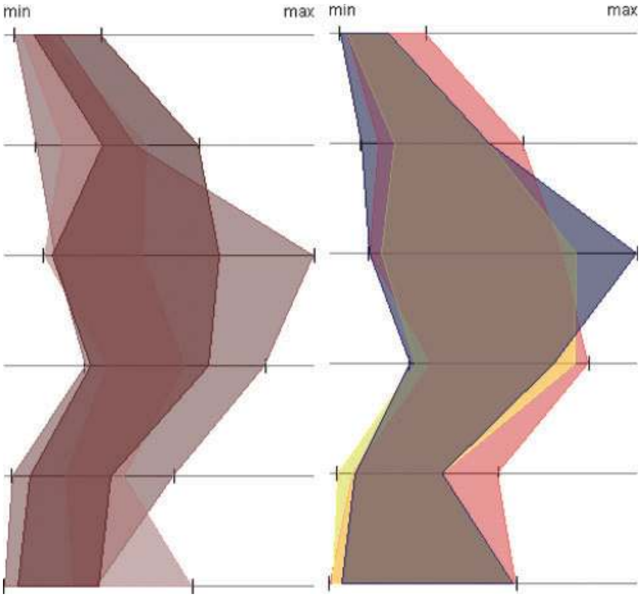


Figure 7. Transparent colour bands, or envelopes, represent the ranges of object characteristics for the classes of counties according to the population density (left) and the clusters according to the age structure (right)

show ranges of attribute values regardless of the distribution of the values within the ranges. Hence, just a single outlier can significantly increase the width of a band representing some class. Therefore, in most real-world cases, envelopes of different classes greatly overlap and do not help to reveal substantial differences in class characteristics. It would be good to build 'smart envelopes' ignoring outliers. Alternatively, an envelope could be drawn so as to provide more information concerning value distribution of each attribute.

We have chosen the second option: we divide each value range into subintervals containing approximately equal number of lines and connect corresponding breaks (quantiles) on adjacent axes. As a result, an envelope is divided into stripes. For better visibility, stripes can be painted slightly differently, as is shown in Figure 8. The user can choose the desired number of subintervals. Thus, in Figure 8, division into 10 subintervals is represented.

It may be seen from Figure 8 that a divided envelope gives much better understanding of characteristics of a class that just its outline. The main impression from the outline is that characteristics within the class greatly vary. However, the partition of the envelope shows us that 80% of values of each attribute lie within quite a narrow interval. It can be seen, for example, that the class is characterized by mostly low values of the third attribute (this is true at least for 90% of class members) and mostly medium values of the second attribute (again, at least 90% of class members have medium values of this attribute). Unfortunately, it is hard to say anything concerning how the first 90% are related to the second 90%. In general, these subsets are different, and discarding the leftmost and the rightmost stripes will not give us a band containing 80% of all lines. This is a consequence of the independent partition of the range of each attribute, which must be borne in mind to avoid misinterpretation of the display.

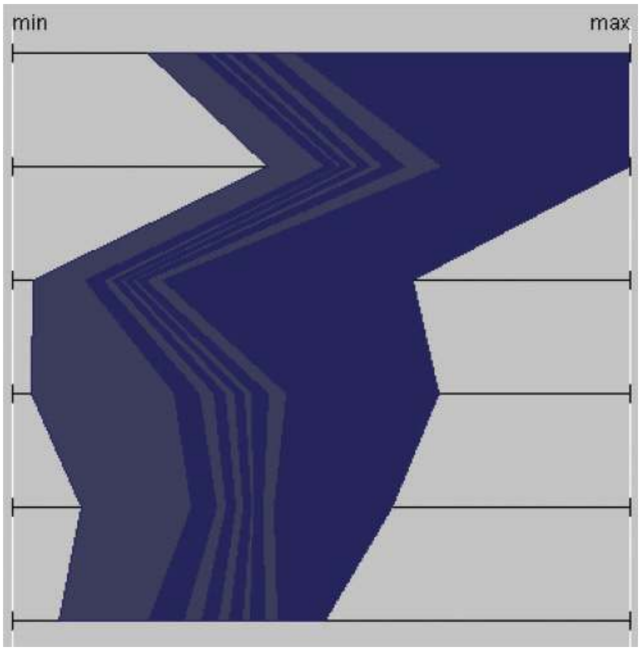


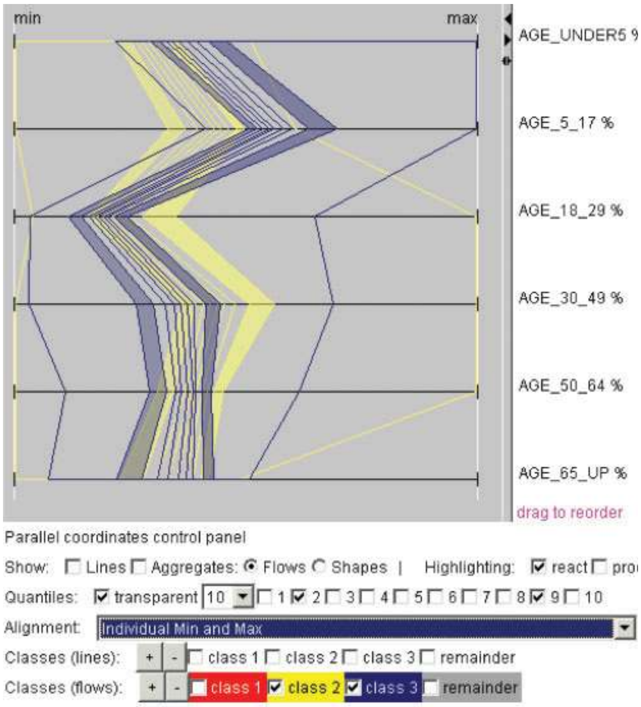
Figure 8. The distribution of object characteristics within a class is shown by dividing the value range of each attribute into 10 equal-frequency subintervals. Hence, each subinterval corresponds to 10% of objects of the class

Despite of this problem, the stripe display can still be useful for investigation of class properties and comparison of classes. In our implementation, the user may switch on and off the representation of any class. This helps the user to focus on a particular class and to make pairwise comparisons between classes, which is easier than analysing three or more classes simultaneously. Class envelopes can be painted in a transparent or opaque mode. Besides, the user can choose which stripes must be painted (filled) and which only shown by bounding lines. Thus, the display in Figure 9 represents two classes out of three: class 2 (lighter colour) and class 3 (darker colour), which is also shown in Figure 8. The class envelopes are divided into 10 stripes. The user has chosen the second and the ninth stripes to be painted in the transparent mode. For the remaining stripes, only outlines are shown.

The display in Figure 9 shows us that class 2 and class 3 differ most significantly by values of the second attribute, proportion of the age group from five to 17 years. Quite substantial distinction exists also with respect to the proportion of children under five years old. There is no difference in proportions of people of the age 65 years or more.

In order to compare the distribution of characteristics within a class to that in the entire set of objects, it is possible to display analogous stripes for the whole set. An alternative way is statistics-based scaling of the axes, which is described in our earlier paper [AA01]. The technique is illustrated in Figure 10. The same information as in Figure 9 is represented, but the axes are distorted and aligned so that the centre of each axis corresponds to the median value of the respective attribute and the middle positions between the centre and the ends correspond to the quartiles.

From Figure 10, it can be noticed that class 3 (shown in the darker colour) substantially deviates from the whole set



Show: □ Lines □ Aggregates: ● Flows ○ Shapes | Highlighting: ☑ react □ pro

Quantiles: ☑ transparent 10 □ 1 ☑ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 ☑ 9 □ 10

Alignment: Individual Min and Max

Classes (lines): + - □ class 1 □ class 2 □ class 3 □ remainder

Classes (flows): + - □ class 1 ☑ class 2 ☑ class 3 □ remainder

Figure 9. Comparison of distributions of attribute values in two classes. Class envelopes are partitioned into 10 stripes, of which only two (2nd and 9th) are painted, according to user's choice

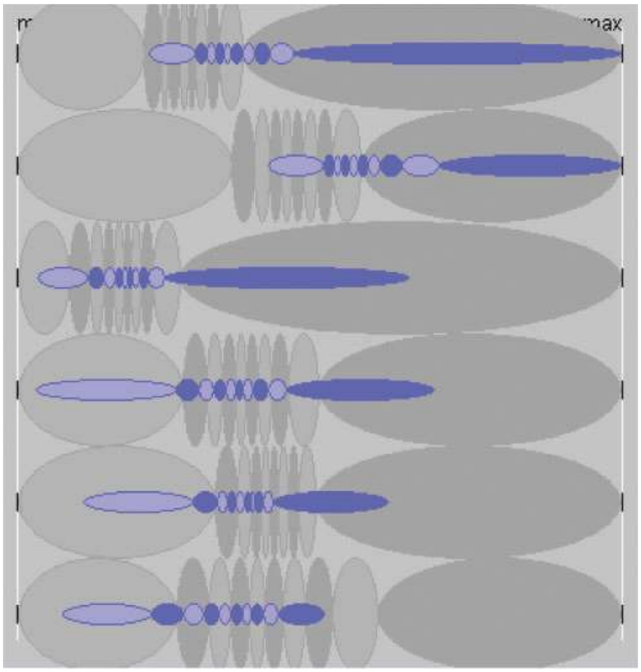
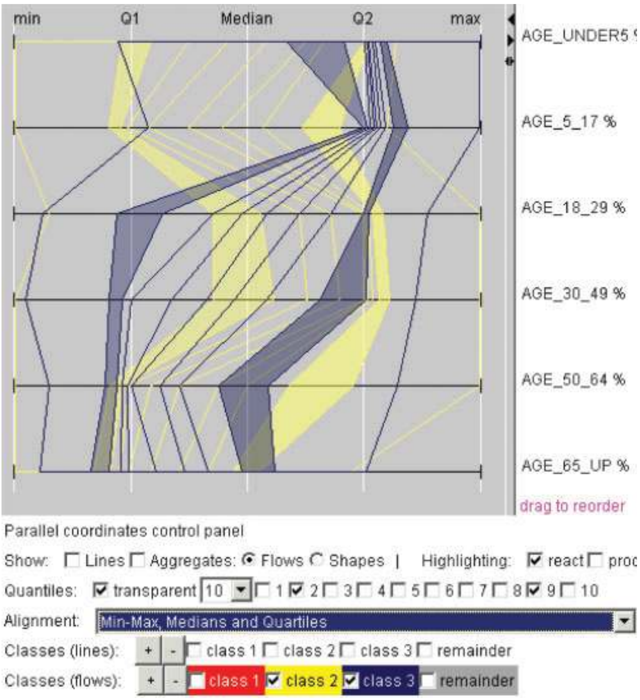


Figure 11. The same class as in Figure 8 (class 3) is represented by ellipse plots. For building the ellipses, the value ranges of the attributes are partitioned into 10 equal-frequency intervals. In the background, ellipse plots for the entire set of counties are shown

with respect to the proportions of the age groups under five years and from five to 17 years. Class 2 has more or less 'standard' proportions of these age groups as well as the group 50–64 years: the bulk of values lie between the first and the last quartiles of the entire dataset. Concerning the age groups 18–29 years and 30–49 years, the proportions in class 3 are about 'standard' while class 2 tends to have higher proportions than in class 3 and in the entire set in general. Both class 2 and class 3 have smaller proportions of the age group 65 and more years in comparison to the distribution for the whole set of counties. Class 3 is also characterized by smaller proportions of the age group 50–64 years.



Show: □ Lines □ Aggregates: ● Flows ○ Shapes | Highlighting: ☑ react □ proc

Quantiles: ☑ transparent 10 □ 1 ☑ 2 □ 3 □ 4 □ 5 □ 6 □ 7 □ 8 ☑ 9 □ 10

Alignment: Min-Max, Medians and Quartiles

Classes (lines): + - □ class 1 □ class 2 □ class 3 □ remainder

Classes (flows): + - □ class 1 ☑ class 2 ☑ class 3 □ remainder

Figure 10. Comparison of value distributions in two classes to that in the entire dataset by means of statistics-based scaling of the axes

The modification of the 'enveloping' technique by partitioning envelopes into stripes can be viewed at the same time as a generalization of putting box plots on plot axes. Thus, when the user chooses to divide the envelopes into four stripes, the division represents the medians and quartiles of the attributes, analogously to box plots. Finer partitions provide more detailed information about value distributions.

Drawing graphics on the axes is an alternative method provided in our implementation for displaying information concerning value distributions. Instead of box plots, which show only medians and quartiles, we use graphs that may be called 'ellipse plots'. Ellipse plots can represent arbitrary partitions, analogously to 'striped' envelopes.

Figure 11 illustrates how ellipse plots are built. Analogously to partitioning envelopes, the value range of each attribute (pertinent to a class or to the entire dataset) is divided into a desired number of equal-frequency subintervals. Then, instead of connecting corresponding quantiles on adjacent axes, as we do when striping envelopes, we draw ellipses around the subintervals, i.e. the horizontal diameters of the ellipses are proportional to the lengths of the subintervals. With this technique, we can also use the vertical diameters of the ellipses to convey

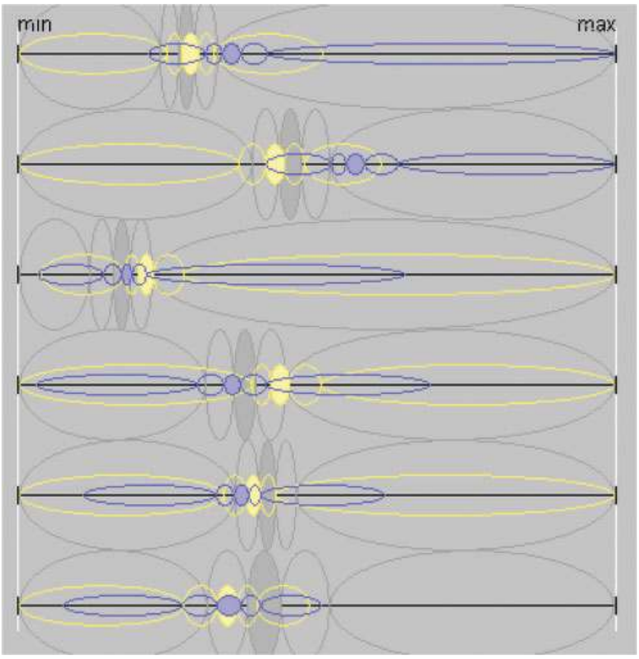


Figure 12. Value distributions in two classes (class 2 and class 3) and the entire dataset are represented by ellipse plots on the basis of partitioning into 5 equal-frequency intervals. Each ellipse stands for 20% of objects of the respective set. The vertical diameters of the ellipses are proportional to the sizes of the sets

additional information. We make them proportional to the sizes of the subsets represented by the ellipses.

In Figure 11, the smaller ellipses show the distribution of attribute values in the same class as is shown in Figure 8 by a 'striped' envelope (specifically, class 3 resulting from the clustering). Like in Figure 8, division into 10 subintervals is applied, i.e. each ellipse represents 10% of the class members. Hence, all information available in Figure 8 can also be seen in Figure 11. Additionally, the bigger ellipses in the background represent the distribution of attribute values in the entire dataset, so that class 3 can be conveniently compared with the whole set of counties. Analogously to the smaller ellipses, each big ellipse stands for 10% of all counties. Thus, we can see on the upper axis that 10% of the whole set of counties that have the highest proportions of the age group below five years contain 40% members of class 3, and top 20% of the whole set contain almost 70% of the class members. A similar observation can be drawn from the second axis corresponding to the age group from five to 17 years: top 20% of the whole set contain about 80% of the class.

Besides comparing the distributions, we can also estimate the size of class 3 in relation to the size of the whole set: the proportions between the heights of the ellipses suggest that the class contains approximately one-fifth of all counties.

Ellipse plots can also be used for comparing two or more classes. Thus, in Figure 12 class 2, class 3 and the entire set are represented simultaneously. For easier comprehension, the attribute ranges are divided this time into five subintervals instead of 10. As can be seen from the figure, the user can choose which of the ellipses will be filled and which remain hollow. In this example, filling is used for the central ellipse in each ellipse plot (i.e. middle 20% of objects of the respective sets). The filling can be opaque, as in the figure, or transparent.

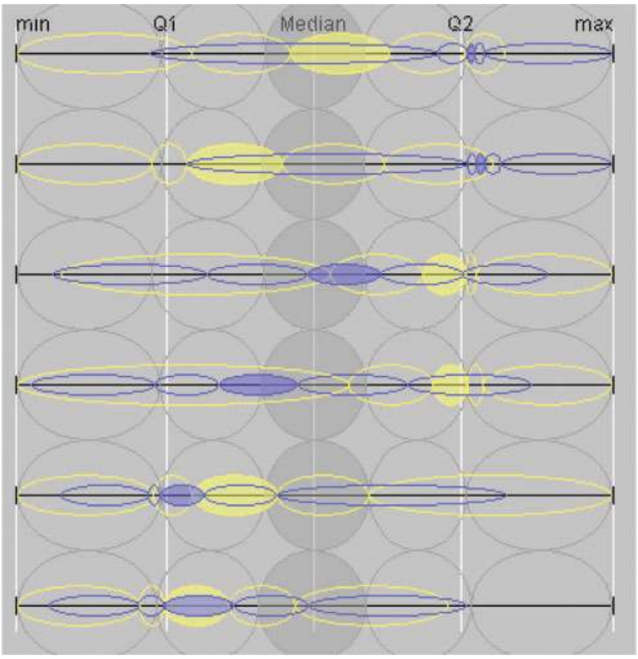


Figure 13. Statistics-based scaling of axes has been applied to the display in Figure 12

With Figure 12, we can make similar comparison of class 2 and class 3 as with Figure 9. Additionally, we can relate our observations concerning these two classes to the distributions of attribute values in the entire dataset. We can also estimate the relative sizes of the classes: class 2 is nearly two times bigger than class 3 and constitutes about 40% of the entire set of counties.

Like with envelopes, we can also apply statistics-based scaling of the axes. After applying this transformation, the display from Figure 12 looks as is shown in Figure 13. Now, the area around the median line can be viewed as a sort of 'standard' range of characteristics, and we can investigate the deviations of the classes from this standard range. The information perceived from this picture is similar to what is conveyed by the 'striped' envelopes in Figure 10.

In comparison to a 'striped' envelope, a collection of ellipse plots representing the same information does not produce an integral image. This is a weakness and an advantage at the same time: a weakness because a single image is easier perceived (and, hence, easier compared with an analogous image of another set of objects) and an advantage since the misleading interpretation of stripes as line containers (i.e. that each line is fully contained in a single stripe) is precluded.

As compared to traditional box-and-whiskers plots, ellipse plots are more general: they can represent not only medians and quartiles but any number of quantiles (in our implementation, from 2 to 10). We use ellipses rather than boxes because two adjacent ellipses touch just in a single point and, hence, can be easier distinguished visually.

Display of aggregated information can be combined with drawing lines for selected individual objects. Object selection can be done through any display. In particular,

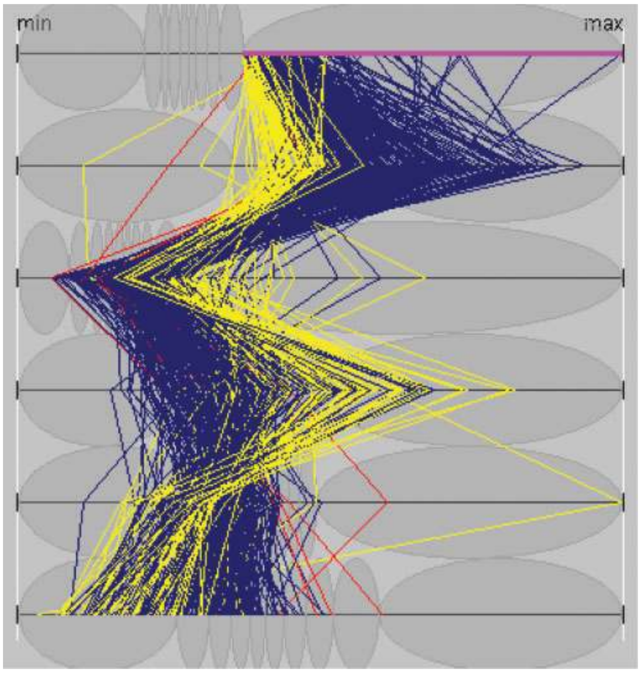


Figure 14. Clicking on an ellipse results in the corresponding objects being selected and their lines appearing on the plot. Here, clicking on the rightmost ellipse on the top axis has selected the top 10% of counties with respect to the proportions of children under 5 years old

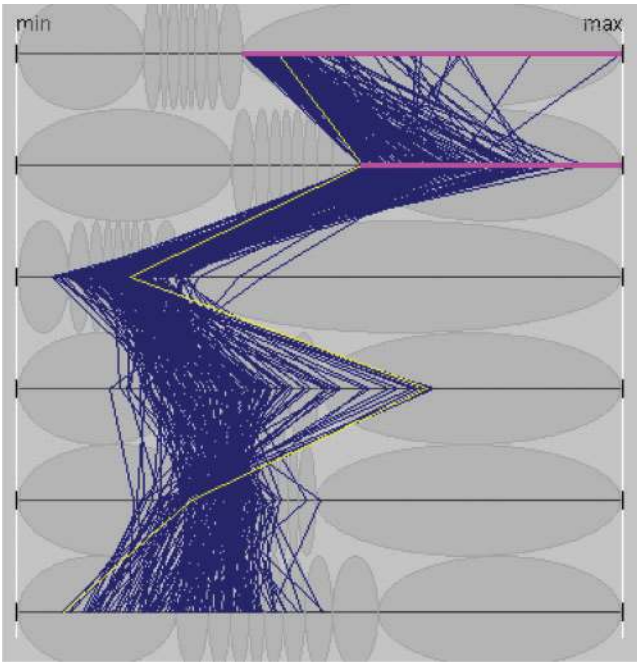


Figure 15. Clicking on the rightmost ellipse on the second axis has modified the selection from Figure 14. Now, selected are the lines of the counties with proportions of the age groups less than 5 years old and from 5 to 17 years old being among the top 10% over the country

one can select objects by clicking or dragging on the parallel coordinate plot. The typical reaction is selection of the objects the lines of which come near the mouse position (in the case of clicking) or cross the rectangular area specified by mouse dragging. When some object classes are propagated to a PCP display, the lines of selected objects are coloured according to the classes the objects belong to (see Figure 14), otherwise the lines are shown in black.

A special type of interaction is possible for a PCP display with ellipse plots. Clicking on an ellipse selects the objects from the corresponding subset. Thus, Figure 14 shows the result of clicking on the rightmost ellipse on the top axis. This ellipse represents the top 10% of the whole set of counties with respect to the proportions of children under five years old. The click on the ellipse has selected this group of objects. It can be seen that the group consists mostly from members of class 3 (dark lines) with a smaller fraction of objects from class 2 (light lines) and only a few members of class 1. Unfortunately, the lines belonging to class 1 (red) cannot be distinguished from the lines of class 3 (blue) in a greyscale reproduction.

Clicking on another ellipse modifies the selection. Thus, Figure 15 shows how the plot from Figure 14 changes after clicking on the rightmost ellipse on the axis second from top, which corresponds to the proportions of the age group from five to 17 years old. The result is selection of the counties the lines of which belong to both ellipses clicked. These are the counties with proportions of the age groups less than five years old and five to 17 years old lying among the top 10% over the country. This subset of counties includes only one member of class 2; the remaining counties belong to class 3.

The general discipline for the interaction with ellipse plots is following: clicking on two or more ellipses on the same axis results in adding new selections to the previously made ones (logical 'OR' operation); clicking on an ellipse on another axis results in intersecting the previous selection with the new one (logical 'AND').

5. DISCUSSION AND CONCLUSION

Our work on PCP modification has been incited by the observation that this technique as well as other traditional methods for data visualization and display coordination cannot properly support analysis of large datasets. Our special interest is analysis of object classes, in particular, results of applying clustering algorithms of data mining (in general, outputs of data mining procedures are often quite difficult to interpret; therefore, a proper visualization support is required). A traditional approach for representing object classes on data displays is so-called multi-coloured brushing, i.e. painting display elements (dots on a scatterplot, lines on a parallel coordinate plot, etc.) in the colours of corresponding classes (see, for example, [HT98]). However, due to overplotting, brushing often fails to convey correct information concerning the classes. Hence, for large datasets, brushing should be substituted by other methods of representing class-relevant information. We believe that the most appropriate approach is to provide such information in an aggregated form.

Our modifications of the basic parallel coordinate plot technique increase its appropriateness for large datasets at the cost of replacing representation of individual instances by the display of aggregated information concerning a dataset as a whole and its subsets (object classes). This allows one to apply parallel coordinate plots for data analysis on the overall and intermediate levels. The elementary level of analysis is supported by the possibility to show individual characteristics of selected objects on top of the aggregate representation.

We have suggested two alternative methods for representing aggregated information: 'striped' envelopes and ellipse plots. Both are based on partitioning the value ranges of the attribute into equal frequency intervals. The envelope representation conveys information concerning value distribution in an object set through a single image whereas ellipse plots do not promote such an integral perception. While ellipse plots are perceptually more complex than 'striped' envelopes, the latter may induce the misleading interpretation of lines being fully enclosed in the stripes without intersecting their boundaries. Ellipse plots can be better combined with drawing individual lines than envelopes.

A limitation of the suggested approach is that it is suitable only for a relatively small number of classes. It is very difficult to compare class characteristics when many classes are simultaneously shown in the display. Although the user may select the classes to view and perform pairwise comparisons, this procedure may become impractical with dozens or hundreds of classes.

The aggregate representations can gain from a proper scaling of PCP. Thus, statistics-based scaling decreases the influence of outliers and therefore can make the visualization more effective. Additionally, it can convey overall level information concerning the distribution of characteristics in the entire dataset. At this cost, the envelope or ellipse plots corresponding to the whole dataset may be omitted from the display thus making it simpler and more legible.

An important feature of the proposed approach is its scalability. Overplotting is reduced because mostly aggregated characteristics are shown and, optionally, just a subset of lines. The computational complexity is M*N*log(N), where N is a number of instances, and M is a number of their attributes. Therefore, the applicability is limited only by the amount of RAM on user's computer.

This restriction can be removed by implementing the visualization in a client-server mode and using a powerful database system on the server side. Thus, the database can compute and provide aggregated characteristics of data (particularly, value ranges, quantiles and other necessary statistics). This is sufficient for the visualization and interactive manipulation. Instance data can be transferred only on demand, for a selected area of interest.

However, a weakness of both 'striped' envelopes and ellipse plots is that they convey summary information for each attribute independently of others. One can compare values distributions of different attributes in classes and entire dataset but cannot investigate relationships between the attributes and cannot explore the distribution of value combinations. Thus, neither envelopes nor ellipse plots give an idea concerning the 'typical profiles' of class members.

This weakness can be partly compensated by the possibility to represent individual characteristics for selected object subsets. Appropriate selections can be made by means of interacting with the parallel coordinate plot. For example, the user may set a kind of 'trajectory mask' by clicking on ellipses on different axes and see how many objects have their characteristics fitting in this mask, where these objects are on a map or other displays, and what classes they belong to. In principle, this is equivalent to applying a dynamic query tool [AWS92], but in the case of ellipse plots information about value distributions can be conveniently used. However, in this interactive way, the user cannot explore all possible masks (characteristic profiles) but only a few. Hence, additional tools are needed to properly support profile analysis.

We plan to implement a computational tool that would count all existing combinations of characteristics for a user-specified partitioning of value ranges of attributes. This may be a partitioning by quantiles, as in ellipse plots and 'striped' envelopes, or a partitioning into a desired number of equal intervals. While the computation itself is rather simple, the combination analysis tool requires a convenient user interface for selecting and partitioning attributes and an effective visualization of the results obtained. Another idea is to combine user-defined partitioning with an automated discovery of association rules by means of data mining followed by interactive visualization and analysis of the results.

Similar ideas can be applied and further developed for the analysis of time-series data. In this case, we can assume that values for consecutive time moments are auto-correlated. This assumption may allow us to develop methods for more sophisticated visual analysis.

ACKNOWLEDGEMENT

This study was partly supported by the European Commission in project GIMMI (IST-2001-34245).

REFERENCES

Andrienko, G. and Andrienko, N. (2001). 'Constructing Parallel Coordinates Plot for Problem Solving', in **Proceedings Smart Graphics**, pp. 9–14, ACM Press, New York [AA01].

Andrienko, N. and Andrienko, G. (2003). 'Informed Spatial Decisions through Coordinated Views', **Information Visualization**, 2, 270–85 [AA03].

Ahlberg, C., Williamson, C. and Shneiderman, B. (1992). 'Dynamic queries for information exploration: an implementation and evaluation', in **Proceedings ACM CHI'92**, pp. 619–26, ACM Press, New York [AWS92].

Bertin, J. (1983). **Semiology of Graphics. Diagrams, Networks, Maps**, The University of Wisconsin Press, Madison [B83].

Berthold, M. R. and Hall, L. O. (2003). 'Visualizing Fuzzy Points in Parallel Coordinates', **IEEE Transactions on Fuzzy Systems**, 11, 369–74 [BH03].

Carr, D. B. (1991). 'Looking at Large Data Sets Using Binned Data Plots', in **Computing and Graphics in Statistics**, ed. by Buja, A., Tukey, P. A., pp. 7–39, Springer-Verlag, New York [C91].

Fua, Y.-H., Ward, M. O. and Rundensteiner, E. A. (1999). 'Hierarchical parallel coordinates for exploration of large datasets', in **Proceedings IEEE Visualization**, pp. 43–50, San Francisco CA, October 1999, IEEE Computer Society Press, Washington [FWR99].

Han, J. and Cercone, N. (2000). 'RuleViz: a model for visualizing knowledge discovery process', in **Proceedings KDD 2000**, pp. 244–53, ACM Press, Boston MA [HC00].

Hoffmann, H. and Theus, M. (1998). 'Selection sequences in Manet', **Computational Statistics**, 13, 77–87 [HT98].

Inselberg, A. (1985). 'The plane with parallel coordinates', **The Visual Computer**, 1, 69–91 [I85].

Inselberg, A. (1990). ‘Parallel Coordinates: A Tool for Visualizing Multi-Dimensional Geometry’, in **Proceedings IEEE Visualization**, pp. 361–78, IEEE Computer Society Press, Washington [I90].

Inselberg, A. (1998). ‘Visual Data Mining with Parallel Coordinates’, **Computational Statistics**, 13, 47–63 [I98].

Inselberg, A., Chomut, T. and Reif, M. (1987). ‘Convexity Algorithms in Parallel Coordinates’, **Journal of the ACM**, 34, 765–801 [ICR87].

Miller, J. J. and Wegman, E. J. (1991). ‘Construction of line densities for parallel coordinate plots’, in **Computing and Graphics in Statistics**, ed. by Buja, A. and Tukey, P. A., pp. 107–23, Springer-Verlag [MW91].

Ong, H.-L. and Lee, H.-Y. (1996). ‘Software report: WinViz — a visual data analysis tool’, **Computers & Graphics**, 20, 83–84 [OL96].

Pratt, K. B. and Tschapek, G. (2003). ‘Visualizing concept drift’, in **Proceedings 9th ACM SIGKDD Conference**, Washington DC, August 2003, pp. 735–40, ACM Press, New York, NY [PT03].

Rao, R. and Card, S. (1994). ‘The Table Lens: Merging graphical and symbolic representations in an interactive Focus+Context visualization for tabular data’, in **Proceedings of ACM CHI’94 Conference on Human Factors in Computing Systems**, Boston MA, April 1994, pp. 318–22, ACM Press, New York [RC94].

Schilling, D. A., Revelle, C. and Cohon, J. (1983). ‘An approach to the display and analysis of multiobjective problems’, **Socio-Economic Planning Sciences**, 17, 57–63 [SRC83].

Siirtola, H. (2000). ‘Direct Manipulation of Parallel Coordinates’, in **Proceedings Information Visualization 2000**, London UK, pp. 373–78, IEEE Computer Society Press,Washington [S00].

Shneiderman, B. (1992), ‘Tree Visualization With Treemaps: A 2-D Space-Filling Approach’, **ACM Transactions on Graphics**, 11, 92–99 [Sh92].

Tukey, J. W. (1977). **Exploratory Data Analysis**, Addison-Wesley, Reading [T77].

Theus, M. (2002). ‘Interactive data visualization using Mondrian’, **Journal of Statistical Software**, 7 [T02].

Tweedie, L. and Spence, R. (1998). ‘The prosection matrix: a tool to support the interactive exploration of statistical models and data’, **Computational Statistics**, 13, 65–76 [TS98].

Wegman, E. J. (1990). ‘Hyperdimensional data analysis using parallel coordinates’, **Journal of the American Statistical Association**, 85, 664–75 [W90].

Wilkinson, L. (1999). **The Grammar of Graphics**, Springer-Verlag, New York [W99].

Witten, I. H. and Frank, E. (1999). **Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations**, Morgan Kaufmann [WF99].