Similarity Brushing for Exploring Multidimensional Relations

Matej Novotný
Comenius University Bratislava
mnovotny@fmph.uniba.sk

Helwig Hauser
VRVis Research Center, Vienna
hauser@vrvis.at

ABSTRACT

Displaying multidimensional information has always been a challenge. Projecting multiple dimensions into a two dimensional display is one of the core tasks of information visualization. The human visual system is limited to a low number of dimensions and therefore the human-oriented projection does not easily combine the whole information contained in the original space. This paper introduces a new interaction tool, that implants the n-dimensional information into a low dimensional view and bridges the projection space with the original space in an intuitive and simple way. In one direction the tool performs n-dimensional data-driven brushing based on screen space interaction. In the opposite direction it allows for interactive visual exploration of the original multidimensional space in an infovis display. The implementation is presented using a standard scatterplot but it can be extended to many other infovis techniques as the concept does not depend on the screen space configuration.

Keywords: Information visualization, brushing, selection, multiple dimensions, interaction, scatterplot.

1 INTRODUCTION

analysis of multidimensional information is a widely spread and important task. Many domains generate and handle data of multiple attributes e.g. physical simulations, biochemical data or stock market information. The raw data themselves contain a lot of knowledge but almost none of it reveals without analysis. Many techniques were developed to support the knowledge discovery and two basic directions of research can be observed. One of them exploits the processing power of computers to work out the knowledge in an automatic way using statistical or data mining methods. The drawbacks of the automatic methods are usually lack of semantics or non-linear logic. Therefore the second approach takes advantage of abstract thinking and domain knowledge of human and often uses visualization-based interfaces to analyze the data. Both human and computer have their own qualities that predetermine them each for specific (and usually different) tasks.

In data analysis domain these two powerful processors are now being used in conjunction and the resulting techniques try to take the best of both worlds to complete their tasks. The power, storage and precise computations of a machine are being combined with the intuition, experience-based judgment and common or domain knowledge.

Journal of WSCG, ISSN 1213-6972, Vol.14, 2006
Plzen, Czech Republic.
Copyright UNION Agency – Science Press

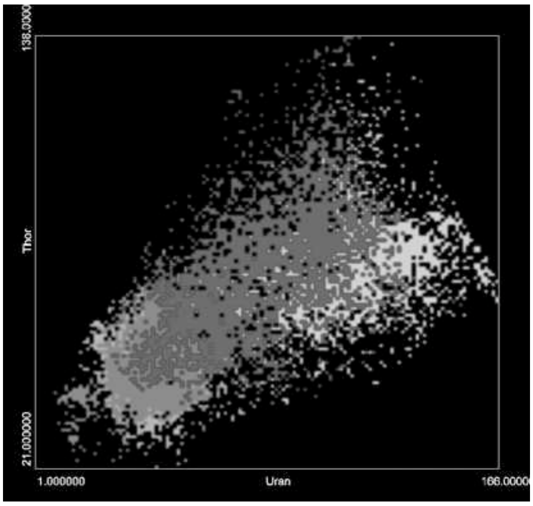

Figure 1: A complex n-dimensional segmentation performed in a simple scatterplot. These overlapping and fuzzy segments would require arduous effort to select if only standard brushing was applied. Please refer to [23] for full color figures.

One of the advantages of computers over humans is the ability to handle high-dimensional information. To compensate for this, the broadest information channel (the human visual system) is popularly used to communicate between computers and humans. Numerous visualization systems operate these days to support this type of information exchange. But when exploring multidimensional information through the means of computer visualization, one usually faces the problem of the low dimensional graphical interface between the human and the computer [20] Our solution presents a way to combine user-driven analysis with the power of automatic processing in order to effectively observe, explore and analyze multidimensional information in any common visualization technique (Figure 1).

1.1 Multidimensional visualization

Numerous solutions for visualization of multivariate data exist but, no matter how precise they are, eventually they bring up the question of how much does a low dimensional projection correspond to its original multidimensional source. The link between the 2D display and the original *n*D data leads through the projection. The action (be it either a selection or an observation) performed in the display extrapolates to the data space in order to match the original multidimensional context. Even though the re-projection extends the 2D action into *n*D, its nature remains two-dimensional. Another 2d action (usually using a different view) has to be presented to refine the action and such a refinement often has to be performed several times to satisfyingly approximate the desired *n*D action through a combination of multiple 2D actions [22], [2].

The tool presented in this paper, called the similarity brush, combines automatic and human-based processing in a way that overcomes the dimensionality bottleneck of a computer display. This is feasible using the presumption that the samples similar one to another are often parts of the same structure regardless of their dimensionality. This enables to bridge the screen space and the data space through similarity information that captures the *n*D structures inside the data. Thus high-dimensional relations can be explored by user in a single 2D display and the interaction with the 2D display connects directly to the original data space where automatic techniques take place. The similarity brush provides a new means to focus user attention and to steer the exploratory process inside a multidimensional environment.

The fact that various similarity measures used to abstract multidimensional information have been heavily investigated in the data mining society [15], [16] creates a reliable theoretical background for abstracting the *n*D information and makes the presented concept a promising framework for visual exploration of multidimensional data.

The tool and the idea behind it are further explained in Sections 3 and 4. Examples of using the similarity brush together with comments on them can be found in Section 5. The related work is addressed in Section 2.

2 RELATED WORK

The need for an accurate display of multivariate data is one of the most motivating stimuli for information visualization. The techniques that visualize multivariate information are basically twofold. Either they reduce the number of dimensions (by dimension subsetting or dimension reduction) so that intuitive visualization methods can be used or they display all dimensions using various sophisticated designs [9] (dimensional stacking, dimension embedding or axis reconfiguration.) Dimension reduction techniques such as principal component analysis [10], self organizing maps [11] or multidimensional scaling [13] produce a low-dimensional representation of the data while trying to preserve most of the multidimensional information. In our approach this information is condensed in a function that describes similarity between two data entries.

The similarity brush uses this function to produce a data-based selection that is derived from a user specified screen-based brush. The idea of data-driven brushes was successfully implemented in the structure based brushes [9] to perform selection in data space, but a hierarchical structure for the data had to be provided beforehand. An attempt to perform data-driven brushing was presented by Martin and Ward [14]. Their solution operates only on the two-dimensional data subspace identical to the screen space and the brush is eventually ruined by being transformed to a combination of regular one-dimensional value-based queries, which naturally includes many undesired entries into the result.

Our approach protects the multidimensional nature of a data-driven brush and works without any a priori given hierarchy. Moreover it stores separate information about the screen-based brush and the derived data-driven brush to enable further refinement in both the data space and the screen space. These two brushes are combined using a framework described in Section 3.2. The framework extends and formalizes various previous approaches to brush combination [5], [21] and balances the combination of two different information spaces.

2.1 Interaction

One way to deal with the limitations of an infovis display leads through changing the parameters of the visualization or performing user-driven operations on the data. Manipulation with the display is a crucial part of the visual exploration. Especially if the data are multivariate and the user has to change his focus, operate with different views, refine his actions or adjust the display to fit his/hers needs. As described in [7], through a realtime interaction with the display the user immerses himself in the data and if the connection between his actions and the reaction of the display is appropriate, even complex structures can be perceived in a 2D display [12].

An important part in interactive exploration is defining the area of interest. This paper addresses this problem by combining *n*D brushes and 2D interaction, which allows to perform multidimensional selection

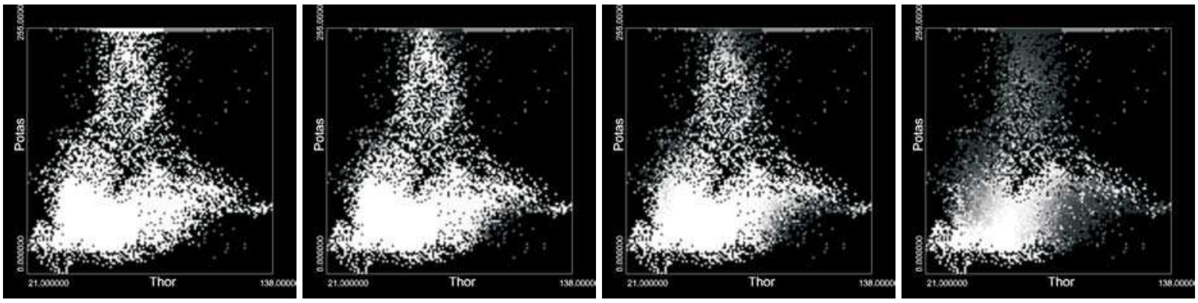

Figure 2: The basics of similarity brushing – primary selection inside the remote sensing data set is conducted on screen as a combination of different local brushes (marked yellow.) The data-driven selection (marked red) is derived from it by decreasing the similarity threshold.

operation using only standard interaction metaphors like brushing or dynamic queries [1], [24].

In a wide area of data analysis tasks the area of interest is not known beforehand and the exploratory process involves looking for interesting structures or patterns in the data. The hereby presented approach takes advantage of another well-known metaphor – the Magic Lens [4] – and uses it to integrate information of *n*D nature into the screen space. The examples demonstrating the advantages of these new interaction options can be found in Section 5.

2.2 Scatterplot

We chose Scatterplot to illustrate the benefits of the similarity brush, as it is a popular and very powerful visualization technique. Scatterplot [3] is plainly an orthographic projection of *n*-dimensional data space into a two-dimensional subspace determined by particular two of the original dimensions. The greatest advantage of the scatterplot lies in the ability to show two-dimensional relations in an instant thanks to the projection that preserves the basic spatial relations. The drawback of the simplicity of the scatterplot are the limitations of the displayed dimensions. Structures exceeding the two specified dimensions or those that are overplotted might get lost in the scatterplot. A different view is usually necessary to improve the visualization. The similarity brush overcomes this limitation and provides valuable information from "behind the scenes". With the help of the similarity brush many new structures that could not be seen before are revealed, mainly those of higher dimensionality or with unsharp and overlapping borders.

3 SIMILARITY BRUSHING

In this section we describe our new approach to jointly operate in visualization space as well as also in data space when interacting with the data, e.g., while selecting data subsets of special interest or during interactive data exploration. Below, we first describe the basic idea of *Similarity Brushing* before we go into details.

3.1 Similarity Brushing – The Basic Idea

For similarity brushing we consider a visualization scenario in which an n-dimensional dataset D (with n usually being around 5 to 50) is visualized in an m-dimensional visualization space V (with $m < n$ and m usually being 2 or 3), i.e., a scenario in which the visualization transform $\mathbf{p} : D \rightarrow V$ introduces a loss of dimensionality. As well known also from a lot of related work, it is difficult (or sometimes even impossible) to properly represent the n-dimensional relations between the data items of D in the m-dimensional visualization. Through $\mathbf{p}$, it is usually well possible that data items, which are far apart from each other in the n-space, lie near to each other in the mD visualization. Accordingly, it easily can happen that data substructures (like data clusters), which clearly are delimited in n-space, show up intermingled in the visualization and therefore cannot be visually differentiated apart from each other.

Similarity brushing now enables the user to jointly address structures in the visualization, i.e., data structures which are preserved by transform $\mathbf{p}$, as well as also structures, which only show up in the original nD data space. The basic idea of similarity brushing is as follows:

Working in visualization space: First, the user interactively marks a certain structure in the visualization (like in standard brushing) to select some data items for further investigation. The prime example here is that the user marks the core of a data structure of interest in the visualization. This can be one data point only, an entire subset of the data, or even a larger part of the visualized data items.

Working in data space: Next, the user extends this first brush to also include further data items which are similar to the already selected data items. The important thing here is that now a distance metric for the nD data space is used (instead of measuring distances in visualization space). Thereby, only those data items are added to the original brush which also are near to the previously selected ones in the n-space. To continue our prime example, the user would thus extend his/her first (quite conservative) selection to also include all other data items of the spotted data structure (but without touching all those data items which only seem to be part of the structure, but not really are – at least in terms of distances in *n*-space). An example is depicted in Figure 3.

The advantage of similarity brushing is that we can exploit the advantages of the visualization as well as of a data-centered approach (such as data mining): (1) The *m*D data visualization usually provides the user with a very intuitive interface to the data – the user literally sees the data in front of him/her. The human visual system is very powerful in detecting interesting structures/subsets in such a visualization. Accordingly, it is very useful to allow the visualization-based selection of data items (as long used under the term brushing). (2) Our approach to extend (not substitute) this interaction also to data space allows to overcome situations where disadvantages of the visualization become apparent such as the loss of dimensionality that leads to ambiguities in the visualization.

We see several key applications of this concept of similarity brushing to ease the interactive visual analysis of *n*-dimensional data as described below:

***n*D substructure brushing:** The most straight-forward application of similarity brushing, as already addressed in the example above, is the interactive selection of *n*D data substructures, which are nicely delimited in *n*-space (but not in *m*-space, i.e., in the visualization space). As described above, the procedure is to (1) select a visually well-separated core subset of the structure under investigation and then (2) extend this brush to also include the other (visually not so well-delimited) data items of the respective data substructure. The result of such an action can be seen in Figure 3.

$(n-k)$D subspace brushing: The substructure brushing described before does not necessarily have to consider the full dimensionality of the data set. Often there are features that reside inside a certain $(n-k)$D subspace of the original data domain but are lost if the whole set of dimensions is considered. The similarity brush allows for user-driven selection of dimensions to use to evaluate the similarity.

Interactive *n*D exploration: Another very useful application of similarity brushing is the interactive visual analysis of high-dimensional properties of the *n*D data. This can be achieved by interactively moving the visualization-based *m*D brush over the visualization and at the same time watching what data items get selected through the brush extension based on the *n*D distance metric. Examples of the application used to discover hidden relations are presented in Figure 4.

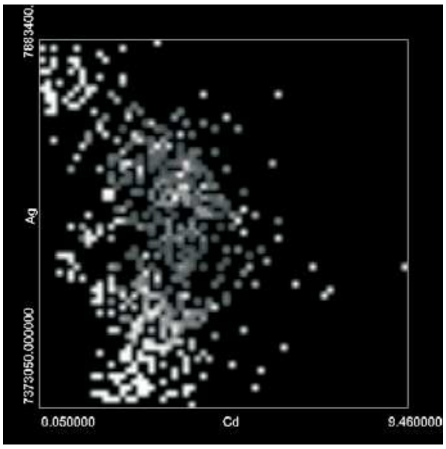

Figure 3: An *n*D similarity-based selection in the geochemical data [8] renders as sparse and scattered in 2D. It is even overlapped by different unselected items. Obviously it would be very hard and too laborious to select this structure using only conventional brushing.

Iterative brush refinement: The fourth interesting application of similarity brushing is the option for iterative brush refinements. In this application, the two-step process of similarity brushing are extended to form a process of alternately working in visualization and data space. For example, a brush can be started in visualization space as described above, then the brush can be extended to *n*D (again as above). But instead of stopping here, the user could go back to visualization space, e.g., alter the visualization setup by choosing a different visualization mapping **p** first and then again restrict the brush to only contain a subset of the currently selected data items (an AND operation with a second brush, for example).

Below, we now present a formal framework of how to integrate the selections in visualization space and those in data space.

3.2 Similarity Brushing – A Unified Framework

First we recall that we assume an *n*D data space D and an *m*D visualization space V, as well as a visualization transform $\mathbf{p} : D \rightarrow V$. In the following, we will now consider the two parts of similarity brushing, i.e., the visualization-based brushing as well as the data space based consideration of distances between data items.

For *m*D brushing (part 1), we assume that brushing interactions result in the assignment of a so-called degree-of-interest (DOI) function b_V to all the data items – $b_V(\mathbf{d}_i)$ is 1 if data item $\mathbf{d}_i$ is brushed, i.e., selected, and 0 if not. Often, b_V will be such a function to either map to 1 or 0, but nothing else (either a data item is brushed, or not). However, in many applications, it also makes sense to allow b_V to map to the entire interval $[0,1]$ – called smooth brushing in the work of Doleisch et al. [6]. Even though we will in the meantime assume the b_V is either 0 or 1, we will further below demonstrate that all the here presented framework also works fine with a smooth brush b_V.

For nD extensions to our mD brushes, we assume to have a nD×nD metric $< .,. >_D \in \mathrm{R}^+$ available in data space to compute distances between nD data items (with $< \mathbf{d}_i, \mathbf{d}_j >_D = 0 \Leftrightarrow \mathbf{d}_i = \mathbf{d}_j$). In a first approach, we will consider the nD extension of an mD brush b_V to be defined as follows: all data items $\mathbf{d}_i$, which not yet are brushed by b_V, i.e., with $b_V(\mathbf{d}_i) = 0$, are checked whether there exists any other (brushed) data item $\mathbf{d}_j$, i.e., with $b_V(\mathbf{d}_j) = 1$, which is near enough, i.e., with $< \mathbf{d}_i, \mathbf{d}_j >_D < d_{\max}$. If such a near and brushed data item $\mathbf{d}_j$ can be found, then $\mathbf{d}_i$ is added to the brush.

In our unified framework, we formulate mD brushing and nD extensions of mD brushes as follows. In addition to brush b_V we assume a non-visual "brush" b_D to map nD distances to 1 (or 0), depending on whether the distance yields an inclusion within the extended brush (or not, respectively). We now integrate b_V and b_D to yield a combined brush b for all data items $\mathbf{d}_i$, depending on whether they are part of the extended similarity brush:

$$b(\mathbf{d}_i) = 1 - \min_j((1 - b_D(< \mathbf{d}_i, \mathbf{d}_j >)) + (1 - b_V(\mathbf{d}_j))) \tag{1}$$

In other words, to evaluate whether a data item $\mathbf{d}_i$ is part of the extended similarity brush b, all data items $\mathbf{d}_j$ are checked (at least in principle; in practice it is sufficient to check only those with $b_V(\mathbf{d}_j) > 0$ – all the others cannot generate a $b > 0$): If there is at least one data item $\mathbf{d}_j$ which (1) lies in the original brush, i.e., $b_V(\mathbf{d}_j) = 1$, and which (2) is near enough to data item $\mathbf{d}_i$, i.e., $b_D(\mathbf{d}_i, \mathbf{d}_j) = 1$, then also b is 1. This, of course, also holds if $\mathbf{d}_i$ itself lies in the original brush b_V. There are a number of nice properties of this integration to be mentioned:

Boundedness of b – The $((1 - b_D(.)) + (1 - b_V(.)))$-argument of the min is bounded (for an arbitrary j) between 0 and 2 (which potentially could lead to negative bs). But for $j = i$, $b_D(< \mathbf{d}_i, \mathbf{d}_j >) = b_D(0) = 1$. This yields that $((1 - b_D) + (1 - b_V))$ is bounded between 0 & 1 for $j = i$. Accordingly, the entire min-expression cannot become more than 1 which consequently yields that $0 < b < 1$.

Preservation of b_V – With the same line of argumentation as above we can show that $b(\mathbf{d}_i) \geq b_V(\mathbf{d}_i)$, i.e., for data items which already lie within the original brush, the extended brush cannot exclude them anymore.

Smooth brushing compliance – Equation (1) also holds for smooth brushes, i.e., $b_V \in [0,1]$ and $b_D \in [0,1]$. The $((1 - b_D) + (1 - b_V))$-expression can also be interpreted as a sum of two distances, one measured in visualization space $(1 - b_V)$ and one measured in data space $(1 - b_D)$. If the sum is small enough, then the resulting b can become greater than 0 which means that the respective point is included within the extended similarity brush b.

For the implementation, a number of optimizations can be realized, of course, to speed up the calculation of b. First, only those data items $\mathbf{d}_j$ need to be checked with a non-zero b_V (this most oftenly is a comparably small number). Second, data items $\mathbf{d}_i$, which are too far away from brush b_V after projection $\mathbf{p}$, do not need to be evaluated since they can never generate a $b > 0$. Practice shows that only a relatively small number of data items actually have to be checked to compute b.

4 SIMILARITY BRUSH WORKFLOW

The process of data exploration using the similarity brush is user-driven and constructed in a way that the user can take advantage of the automatic methods during the whole process. In the first stage a primary selection (depicted in yellow) is performed in the screen space using usual brushes and their various combinations (AND, OR, NOT). The secondary selection is a data-driven brush derived from the primary selection and is depicted in red. The last parameter is the similarity threshold that, basically, determines the extent of the selection.

The selection process can be iterated or refined to support complex data exploration tasks. Let's consider real world data (Figure 2). These data contain many outliers that for many reasons are usually considered an undesired feature of the data. To remove the outliers, we select several among them on screen and then extend this selection in the data space to include all the outliers. This selection can than be refined in other dimensions to include more outliers or remove entries that are of interest with respect to a different projection. After that the outliers, which are now selected using the similarity brush, can be removed and the data analysis can continue.

4.1 Advanced Interaction

To support the visual exploration tasks such as segmentation or classification a number of other functionalities is present. Any performed selection can be stored as a segment which excludes its entries from further brushing and is marked by a different color. In addition, every segment can be broken apart which reverses the segmentation process and returns its entries back to the data domain. With the support for unsharp and overlapping features provided by the similarity brush this allows for efficient user-based data-driven classification or segmentation of multidimensional data. (see example)

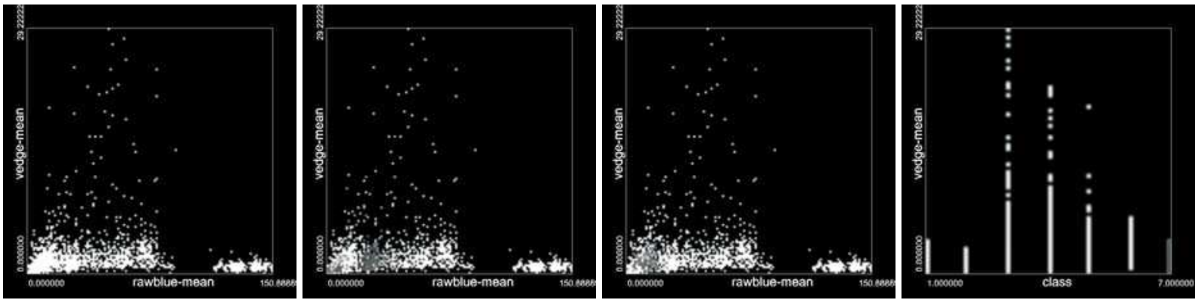

Figure 4: Major changes in the underlying structure discovered using the realtime exploration feature. The assumed indifferent region (first picture) in the bottom left of the scatterplot is evidently compiled of two separate structures (second and third picture.) The fourth picture proves the smaller structure being identical with the class number seven.

5 EXAMPLES

In the following sections, several examples illustrate the using of similarity brush to discover interesting multidimensional behavior or to easily perform complicated brushes. The data sets for these examples are the remote sensing data [19] obtained from SPOT satellites [17]. It contains 5 distinct channels (SPOT, magnetics, 3 bands of radiometrics) combined for a particular region in Western Australia. The second data set contains geochemical data [8] of concentration of multiple elements in a series of observed samples. The last one is one of the Statlog datasets [18] and contains samples produced by image processing together. It is a data set that is usually used for training automated techniques and thus it also includes classification information (brickface, sky, foliage etc.) We used this classification to partially evaluate relevance of our exploration.

5.1 Separate structures

Interesting topology can be discovered in a 2D display using the similarity brush. For example sudden changes in the brushes generated by two areas imply that these areas are separate structures in the original *n*-dimensional space. This can be explored using the realtime exploration feature of the similarity brush. The user moves the primary brush (often only a point selection) over the display and observes the changes of the secondary brush. An area in the image processing data, that was previously considered homogeneous, turned out to consist of two separate structures (Figure 4.) To illustrate the relevance of the data-driven brush we compared this new information to the classification provided with the data set. The entries encompassed in the brush fully correspond to those segments of the source images that depict grass.

5.2 Subspace relations

Unlike the previous example, the structures don't only have to be separate in the full dimensionality of the original data domain. Two different three-way combinations (SPOT, Magnetics, Uranium and SPOT, Thorium, Uranium) of dimensions were used to create two different similarity measures. Given the same screen-based primary selection (samples with very high potassium values) two data-driven selections were derived from that (Figure 5) and we observe differences between them. The most significant difference is that a change in the set of considered dimensions splits the dense U-shaped cluster into two, revealing its two-fold intrinsic nature. The left part (with low thorium values, evaluated using SPOT, Magnetics, Uranium) and the right part (with high thorium values, evaluated using SPOT, Thorium, Uranium). The left part is much more similar to the samples with high potassium values with respect to the magnetics characteristics. Unlike that, the right part is more similar to the high potassium samples with respect to the concentration of thorium.

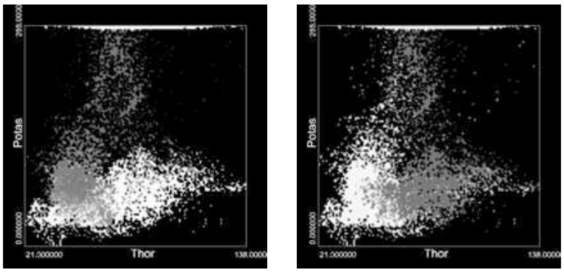

Figure 5: Two different subsets of dimensions were used to compute similarity information. This resulted in two different data-driven brushes (red) given the same primary selection (yellow).

Using only usual visualization the cluster would probably be considered homogeneous. The real nature of the cluster could be discovered using automatic data mining, but without user interaction the analysis of such a knowledge would require additional human-based effort.

5.3 Anomalies

The geochemical data contains an interesting entry that was discovered using the similarity brush. When investigating the data set using the realtime similarity brush, one entry was found to generate no secondary brush. Even though this entry is not depicted as an outlier (in

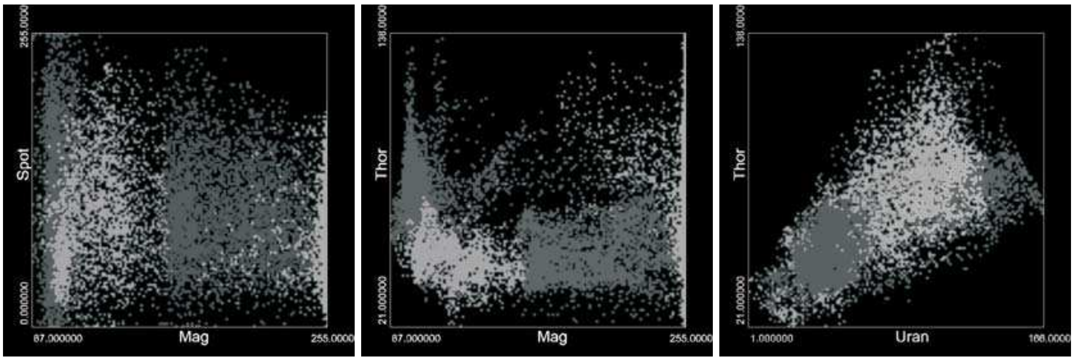

Figure 6: Complex, unsharp and overlapping segments are seldom feasible when conventional brushing techniques are applied. With similarity brush the segmentation of the remote sensing data set took less than a minute and required only simple interaction. Please refer to [23] for full color pictures.

any projection of the geochemical data) it is not similar to any of the remaining entries unless a very loose threshold is chosen. It is a hot candidate for a multidimensional outlier – a sample that lies within a reasonable range of neighbors in every projection, but the sets of the neighbors change over the dimensions (imagine a point in the centre of a hollow sphere-shaped shell.) This makes it isolated if the full dimensionality of the space is taken into account.

5.4 Interactive segmentation

Automated techniques are often used to perform segmentation tasks. But the automated techniques in many ways benefit from the domain knowledge, intuition and abstract thinking of the human user. In conjunction with the similarity brush, the user has the ability to incorporate his capabilities into the segmentation process by specifying interactively the core of the segments and the difference tolerance level within a segment. This gives him a two-fold advantage over the automated techniques. First, the user can specify complex and sophisticated starting points for the segmentation process via creating the primary selection. Second, the user can at any time refine his selections, backtrack the steps and change the decisions he/she made. These actions are rarely performed in an automatic segmentation process.

In addition, the similarity brush interleaves the semantic identification process with the segmentation process. If a computer performs data segmentation automatically, it often produces segments without an actual meaning and additional human-based processing has to take place in order to identify the semantics of the segments. In interactive segmentation provided by similarity brushing the segment starts as a specific core that is user-specified and thus correspond to some real world knowledge provided by the human.

The Figure 6 shows the results of user-based segmentation on a dense multidimensional data set. The resulting segments are consistent and prove to be compact in all views (only three are depicted here though.) The segments have sparse boundaries and overlap in most of the views, which is a common property among real world multivariate data, and would be difficult to mark out using only conventional screen-based brushes.

6 EXTENDING THE CONCEPT

The similarity information computed from all the dimensions of a data set offers hints about the *n*D nature of the information. As a concept, this can be easily incorporated into other popular displaying techniques, such as the parallel coordinates or the histograms. Also possible extensions of the concept could be used in scientific or flow visualization.

The design allows for arbitrary similarity functions to be used. Among the most popular ones are the spatial distance measures (Euclid, Chebychev, Manhattan, Mahalanobis.) Another option is to use information gained from e.g. fuzzy clustering or other automated data mining techniques. Such techniques detect items of similar properties in the set and group them together. The similarity of two samples could thus be evaluated using this information.

Another promising extension is to allow new samples to "join" the screen-based brush if they are close enough or follow other given criteria. This would allow for the chaining effect known from data mining and structures of even more complex shapes could be addressed.

7 CONCLUSION

The tool presented in this paper uses a combination of visual and automatic data-mining to introduces a new way to integrate *n*-dimensional information into a low dimensional display. By interaction with the similarity brush, the user gets to directly touch the multidimensional structures in their original space instead of having to only approximate this by numerous low-dimensional actions. This interaction technique can be used to enhance visual exploration of multidimensional data. As shown by the examples, complex multidimensional topology can be observed even in a simple scatterplot by using the similarity brush. With the use of the similarity brush for visual exploration, extra information can be provided that might help the user to steer his precious attention in further visual exploration actions.

This intuitive tool does not encumber the user's perception by generating visual overload and can be successfully used in many displays. We believe that the similarity brush may well become a useful interaction tool for exploring multidimensional data in many future applications.

8 ACKNOWLEDGMENTS

The research is a part of the basic research on visualization conducted in cooperation between the Comenius University, Bratislava and the VRVis Research Center in Vienna (http://www.vrvis.at). VRVis is funded by an Austrian research program called Kplus. The project is also partly funded by the Slovak government grant VEGA 1/3083/06.
The authors would also like to thank Prof. Peter Filzmoser from Vienna University of Technology for his kind help with the data sets.

REFERENCES

[1] C. Ahlberg, C. Williamson, and B. Shneiderman. Dynamic queries for information exploration: an implementation and evaluation. In *CHI '92: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 619–626, New York, NY, USA, 1992. ACM Press.

[2] M.Q.W. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Proceedings of the working conference on Advanced visual interfaces*, pages 110–119. ACM Press, 2000.

[3] R.A. Becker and W.S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.

[4] E.A. Bier, M.C. Stone, K. Pier, W. Buxton, and T.D. DeRose. Toolglass and magic lenses: the see-through interface. In *SIGGRAPH '93: Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, pages 73–80, New York, NY, USA, 1993. ACM Press.

[5] H. Chen. Compound brushing. In *Proceedings of the IEEE Symposium on Information Visualization 2003 (INFOVIS'03)*, 2003.

[6] H. Doleisch and H. Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. *Journal of WSCG*, 10(1):147–154, 2002.

[7] S. Eick and G. Wills. *High Interaction Graphics*. 1995.

[8] C.Reimann et al. *Environmental Geochemical Atlas of the Central Barents Region*. 1998.

[9] Y.H. Fua, M.O. Ward, and E.A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *INFOVIS*, pages 58–64, 1999.

[10] I. T. Jolliffe. *Principal Component Analysis*. Series in Statistics. Springer-Verlag, 1986.

[11] T. Kohonen. *Self organizing maps*. Springer, New York, 2000.

[12] R. Kosara, H. Hauser, and D. Gresh. An interaction view on information visualization, star. In *EUROGRAPHICS 2003*, 2003.

[13] J. Kruskal and M. Wish. *Multidimensional Scaling*. Sage Publications, 1978.

[14] A.R. Martin and M.O. Ward. High dimensional brushing for interactive exploration of multivariate data. In *VIS '95: Proceedings of the 6th conference on Visualization '95*, page 271, Washington, DC, USA, 1995. IEEE Computer Society.

[15] L. Parsons, E. Haque, and H. Liu. Subspace clustering for high dimensional data: a review. *SIGKDD Explor. Newsl.*, 6(1):90–105, 2004.

[16] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest neighbor queries. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 71–79, New York, NY, USA, 1995. ACM Press.

[17] http://www.spot.com.

[18] http://www.liacc.up.pt/ml/statlog/datasets.html.

[19] http://davis.wpi.edu/˜xmdv/datasets.html.

[20] E.R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[21] M. O. Ward. Creating and manipulating n-dimensional brushes. In *Proceedings of Joint Statistical Meeting*, pages 6–14, August.

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

[23] http://www.vrvis.at/via/research/simi-brush/.

[24] G. J. Wills. Selection: 524,288 ways to say "this is interesting". In *INFOVIS '96: Proceedings of the 1996 IEEE Symposium on Information Visualization (INFOVIS '96)*, page 54, Washington, DC, USA, 1996. IEEE Computer Society.