Brushing of Attribute Clouds for the Visualization of Multivariate Data

Heike Jänicke, Michael Böttinger, and Gerik Scheuermann, *Member, IEEE*

Abstract—The visualization and exploration of multivariate data is still a challenging task. Methods either try to visualize all variables simultaneously at each position using glyph-based approaches or use linked views for the interaction between attribute space and physical domain such as brushing of scatterplots. Most visualizations of the attribute space are either difficult to understand or suffer from visual clutter. We propose a transformation of the high-dimensional data in attribute space to 2D that results in a point cloud, called attribute cloud, such that points with similar multivariate attributes are located close to each other. The transformation is based on ideas from multivariate density estimation and manifold learning. The resulting attribute cloud is an easy to understand visualization of multivariate data in two dimensions. We explain several techniques to incorporate additional information into the attribute cloud, that help the user get a better understanding of multivariate data. Using different examples from fluid dynamics and climate simulation, we show how brushing can be used to explore the attribute cloud and find interesting structures in physical space.

Index Terms—Multivariate data, brushing, data transformation, manifold learning, linked views.

1 Introduction

Although many scientific data-sets comprise multiple variables, their visualization is still a challenging task. Multivariate data occurs for example in fluid dynamics, where simulations feature velocity, pressure, temperature and density values. Even more variables are used in climate simulations where additional information on precipitation, evaporation, and cloud cover is important. As usually more variables than spatial dimensions are available, techniques are required that combine the different variables in a more compressed visualization. Two different strategies for the visualization of multivariate/multifield data can be distinguished: On the one hand, glyphs are used to represent the individual values at each position. An example is the use of polygons with n vertices, where n is the number of variables, e.g., starplots [19]. The distance between the center of the polygon and the vertex depicts the value of the corresponding variable. On the other hand, the multivariate data can be visualized in attribute space. Scatterplots/scatterplot matrices [19] are a common example for this type of analysis. Here, interesting parts of the scatterplot can be selected and highlighted in the physical data-set. Glyph-based techniques are well suited if the user wants to analyze the values at certain positions. However, if the goal of the visualization is a general exploration of the data-set, methods in attribute-space are more appropriate. These techniques can cope with more variables, as they provide a higher level of abstraction. Moreover, the user can more easily search for certain combinations of values or detect correlations. As the focus of this paper is on data-exploration, we will follow the second paradigm.

Different techniques have been proposed for the visualization of data in attribute-space. If more than two variables are to be analyzed, scatterplots are extended to scatterplot matrices [4]. At each matrix entry, the scatterplot of the two corresponding variables is displayed. This visualization technique can be thought of as an ordered collection of two dimensional projections of the high-dimensional space. A different approach is taken by parallel coordinates [10]. In this approach, the different variables are displayed on parallel axes. Value combinations that occur in the data-set are linked by poly-lines across the different axes. Moreover, methods are available that reduce the dimensionality of the data, such as principal components analysis (PCA) [13] where the data is projected to the axes of largest variance.

After the visualization of data in attribute space, methods are needed that allow the user to link the physical and the attribute domain. A well established technique is brushing as introduced by Becker et al. [1]. The basic idea of brushing is that the user can select a subset of the data in attribute space by brushing corresponding points in the scatterplot(s) and that positions in the physical domain holding these values are highlighted. Martin and Ward [15] improved the performance of the initial brushing. They provide brushes with the same dimensionality as the attribute space (N-D instead of 2D) and allow for different operations on the brush, e.g., interactive manipulation. A multi-resolution approach for brushing was introduced by Wong and Bergeron [24]. To cope with larger data-sets, Fua et al. [7] used structure-based brushes on clustered data. Henze [9] first applied brushing to CFD data using multiple metric spaces and linked views. Hauser et al. [8] applied brushing to parallel coordinates and Doleisch and Hauser [5] to flow simulation data. Tricoche et al. [22] used multi-dimensional transfer functions, which act like brushing, for the visualization of flow derived scalar quantities. Operations on multiple brushes such as joints and subtractions were proposed by Chen [3]. Roberts and Wright [17] incorporate additional items such as menus or axes into the brushing process.

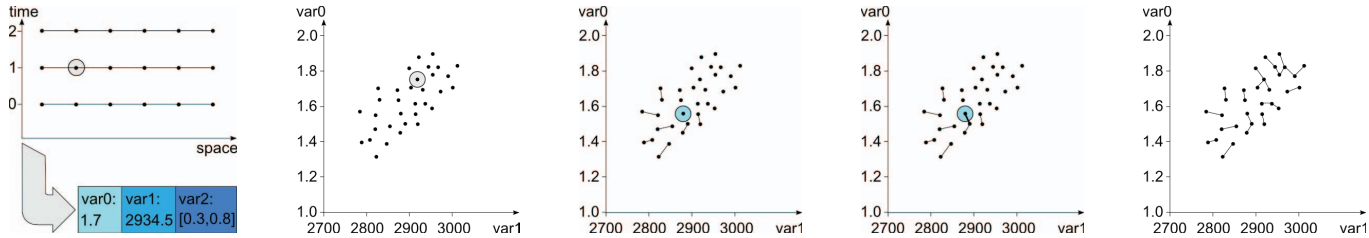
Except for the method proposed in [7], all brushing techniques operate on the two basic multivariate visualization techniques in attribute space, scatterplots and parallel coordinates. Scatterplot matrices have the disadvantage that the amount of plots increases quadratically with the number of variables. Thus, data with many variables is hard to analyze. Parallel coordinates allow for the visualization of many variables in a single image. However, this kind of visualization suffers often from visual clutter. In this paper we propose a brushing technique that operates on preprocessed data. We adopt a technique from multivariate statistics to provide a two-dimensional visualization of the attribute space without clutter that still allows for the analysis of similarities and correlations in the data-set. A combination of brushing and boxplots is used to interactively explore the multivariate data.

2 Transformation of Multivariate Data

A large variety of statistics is available to analyze multivariate data [14]. Many of these techniques aim at the identification of interesting formations in the data, such as clusters or linear correlations. In our data-sets from scientific visualization, however, we observed no such prominent structures. The high-dimensional scatterplot rather resembles a single point cloud that features changes in density that are not isolated but form a branching skeleton (cf. Fig. 7). Thus, we aim at an interactive exploration of the data. The user is presented a

- *Heike Jänicke and Gerik Scheuermann are with Universität Leipzig, E-mail: {jaenicke,scheuermann}@informatik.uni-leipzig.de.*
- *Michael Böttinger is with German Climate Computing Centre (DKRZ), E-mail: boettinger@dkrz.de.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.



(a) Multivariate data of a single position. (b) Attribute space of the first two variables. (c) Edge for the outlined point required. (d) Link to other fragment. (e) After linking all size 1 fragments.

Fig. 1. Computation of the Euclidean Minimal Spanning Tree: (a) For each position the attribute vector is determined and (b) corresponds to a point in attribute space. (c) The fragment outline in blue is to be linked to the closest non-fragment point (d). (e) Intermediate step after linking all size one fragments.

two-dimensional representation of the multivariate data, that is easy to understand and can be further investigated. Our work was inspired by the method proposed by Stuetzle [20]. Before extracting the clusters in a data-set, they compute an Euclidean minimal spanning tree that represents the skeleton of the high-dimensional density. This skeleton forms the basis of our representation of the data.

2.1 Euclidean Minimal Spanning Trees

In order to compute the Euclidean Minimal Spanning Tree (EMST), we need the high-dimensional point cloud in attribute space spanned by the entire set of attributes (simplified illustration in Fig. 1(b)). Therefore, we compute for each position in the data-set an attribute vector that consists of the individual multivariate values at this position (Fig. 1(a)). A spanning tree connects all points in attribute space with line segments, such that the resulting graph is connected and has no cycles. Hence, we need $n-1$ edges to connect n nodes. A spanning tree is a EMST if the sum of the Euclidean distances between connected points is minimal. For the computation of the EMST we used the algorithm proposed by Nevalainen et al. [16]. The method iteratively joins the fragment (set of linked nodes) with fewest nodes to the closest point that is not included in the fragment itself. After $n-1$ iterations the EMST is finished. The algorithm consists of three steps that we will review briefly.

- Store the points in attribute space in a kd-tree [2] to allow for efficient search of neighbors.
- Store each single point in a separate fragment and initialize the queue of fragments' sizes. Fragments of the same size can be inserted arbitrarily.
- Choose the fragment with minimal size and add the shortest edge between a fragment and a non fragment point. Continue this step until the number of fragments is one (and the EMST is finished).

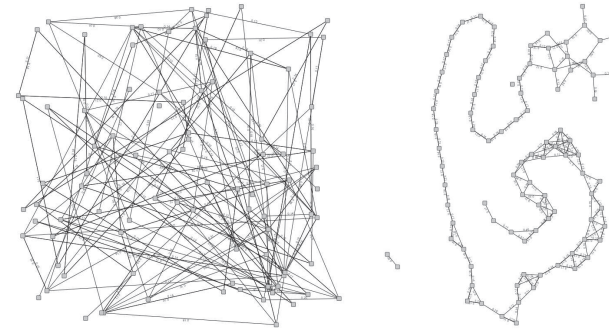
Figures 1(c-e) display intermediate steps of the linking procedure. In Figure 1(c) the first seven fragments of size one have been linked to the closest non-fragment point. The point outlined in blue is the next fragment to be processed. Searching for the closest point, we find the edge added in Figure 1(d). Here we see, that linking to fragments of arbitrary size is a valid step. After linking all fragments of size one, we obtain the graph displayed in Figure 1(e). The final EMST is given in Figure 3(a).

Conceptually, this technique is similar to the Isomap method [21] from manifold learning, where the neighborhood graph (similar to the EMST) is projected to lower dimensions using multidimensional scaling.

2.2 Layout in 2D – The Attribute Cloud

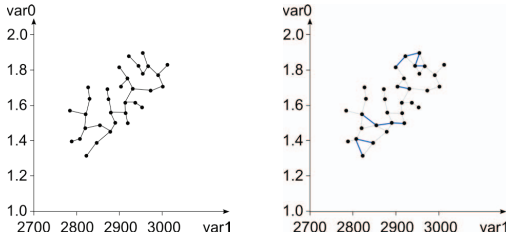
For clarity of presentation we chose an example in 2D. Common computations are in much higher space and therefore the resulting EMST cannot be visualized easily. We need a transformation to 2D. As mentioned earlier, the point clouds we observe in attribute space are compact and extend more or less uniform in all direction. Thus, scaling approaches such as PCA or multidimensional scaling are less suited. Our approach is based on graph drawing, where the EMST is first projected to 2D by assigning each node an arbitrary position. Afterwards the graph is laid out to achieve appropriate edge lengths and few edge intersections. From the large variety of existing layout algorithms we chose the force-directed layout algorithm by Fruchterman and Reingold [6]. The basic idea of this algorithm is to model the graph as a system of springs. Each edge in the graph is represented by a spring. All springs have the same spring rate and thus try to have the same length l. In an iterative optimization process the nodes are moved such that all springs are as close to length l as possible. In the original algorithm the user has to specify the size of the bounding rectangle of the graph. We replaced this restriction by the definition of desired edge length l. Thus, the graph can extend arbitrarily in x- and y-direction and take the shape that is best suited for its structure. A second change that we incorporated is the possibility to have individual optimal edge lengths l_i for each edge taking into account the actual length in high-dimensional space. Figure 2 gives a layout example using the modified Fruchterman-Reingold algorithm.

More efficient layout algorithms are available for trees. However, using solely the EMST for the two-dimensional layout, destroys local correlations of the high-dimensional point cloud easily, as only few points are linked and neighboring branches may diverge. Consider the two branches in the lower left part of the graph in Figure 3(a). Although all points are rather close, no pair is linked and in a layout process these branches might easily drift apart. To prevent this behavior, we add additional edges as illustrated in Figure 3(b). For each node $node$ we add all edges that have maximally size $\alpha \times shortestEdge$, where α is a constant parameter and $shortestEdge$ is the Euclidean distance to the closest point. In all our examples we set α to 1.05 and resolved good results. As this approach introduces cycles, tree layout algorithms are no longer applicable and we have to use a more general layout algorithm as the Fruchterman-Reingold algorithm. For α we



(a) no layout (b) layout

Fig. 2. Fruchterman-Reingold graph layout: The randomly initialized graph before (a) and after the layout process (b).



(a) Final Euclidean minimal spanning tree. (b) Additional edges.

Fig. 3. Euclidean minimal spanning tree (EMST): (a) Final EMST. (b) Additional edges (blue) required for the layout of the EMST in 2D.

tried different values. In the range between 1.00 and 1.03 we observed that many loose branches at the boundaries occurred. For α values greater than 1.1 the attribute cloud becomes a single compact structure as too many different parts of the EMST are linked.

One problem that occurs using the Fruchterman-Reingold layout algorithm is the fact that it starts with a random layout and therefore is not deterministic. In our applications we observed that the layout algorithm only altered the orientation of the attribute cloud and not the shape. Two strategies can be followed to overcome these changes. A simple and straightforward one is to store the layout of the graph. This approach has the advantage that the layout has to be computed only once. However, this strategy is not well suited if for example the attribute cloud of two different data-sets are to be compared. Here, deterministic graph layout algorithms can help. In our applications we stored the layout for successive analysis.

2.3 Discretization

The algorithm explained so far transforms the multi-dimensional point-cloud in attribute space to 2D. If we incorporate the exact numerical values of all positions in physical space individually, the image easily suffers from occlusion and visual clutter. Therefore, we discretize the values before the transformation process. The data range is divided into a number of subranges (e.g. 5 subranges) with $value_{min} \leq r_1 \leq r_2 \leq \ldots \leq r_n \leq value_{max}$. Only the different discretized attribute vectors are used for the computation of the EMST and the layout. Color coding and scaling of the points of the attribute cloud are used to indicate the number of positions that belong to a representing point.

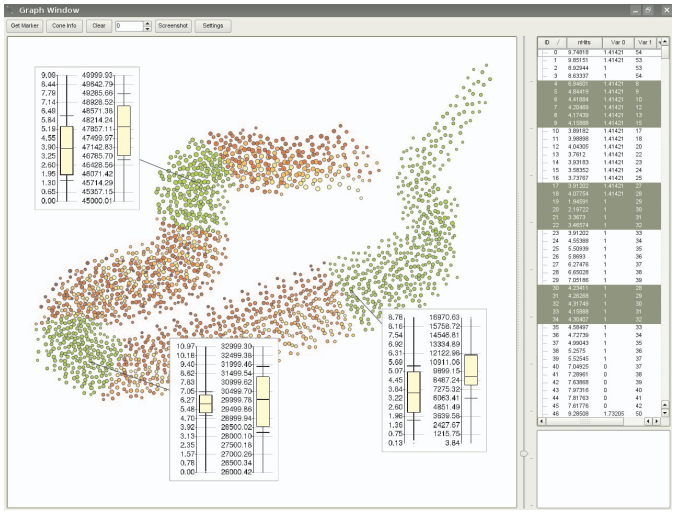


Fig. 4. The brushing window consists of two parts: The brushing area on the left and a column for information on the points (right).

3 Visualization

3.1 Incorporating Information into the Attribute Cloud

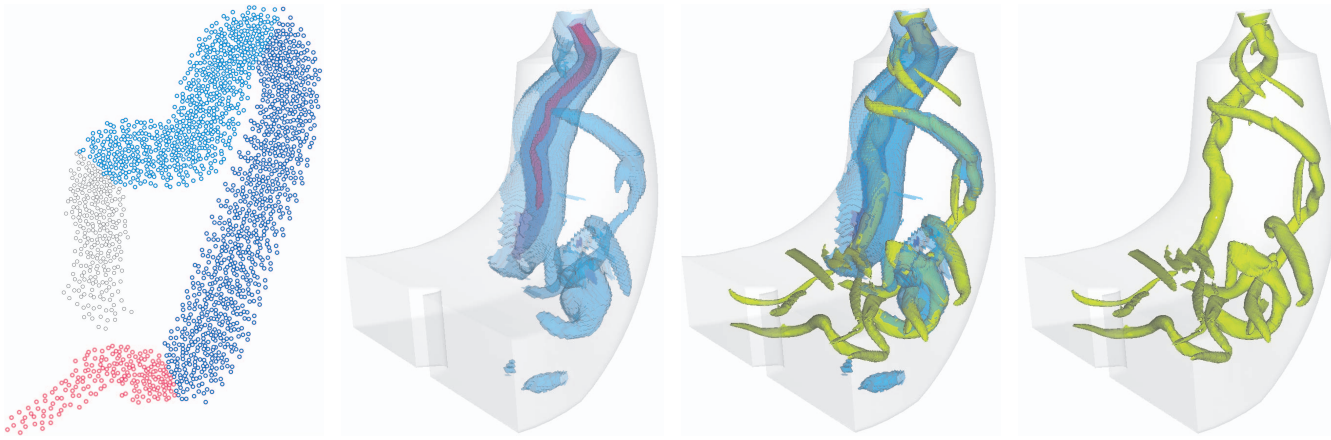
As the attribute cloud cannot be annotated with axes we need different means to enable the user to find certain values and get a better understanding of the point cloud. For this purpose we use color coding and scaling of the points. The user can interactively switch between the different variables, and the color coding of the points change to the desired quantity. Figure 6 shows the color coding for the z component of velocity, the norm of velocity and pressure of a fluid dynamics data-set. Incorporating three-dimensional vector information into a 2D point cloud is rather difficult. Therefore we chose the annotation using the different components of the vector. Additionally, the cloud can be colored using the norm of the velocity to give the user an idea where areas of strong current are located. Alternative annotations of the attribute cloud are color coding of the norm of the deviation from the average or potential flow. Investigating the different variables, we see that the norm of the velocity changes across the cloud (that rather resembles a snake) and pressure changes along it. Thus, the user can easily select regions that feature a combination of norm of velocity and pressure that he/she is interested in. A histogram can be used to select a certain range of the individual variables. In the given attribute clouds, the points are scaled according to the number of points in physical space they represent. Thus, the user gets quantitative information on the distribution without visual clutter.

Conceptually this approach is dual to brushing of scatterplot matrices (cf. Fig. 7). While we keep the positions of points fixed and change values, the points "move" to appropriate values in scatterplots. Both approaches have advantages and disadvantages. In a scatterplot matrix the user can easily search for values in a well-known coordinate system. In the attribute cloud the user has to interact with the visualization to get this information. As changing the color coding is interactive, we think this is a minor disadvantage. The profit gained by this aspect is the fact that in our visualization positions of subsets are easier to remember. In a scatterplot representation the user has to remember a contour in each scatterplot. With the attribute cloud, the user operates on a single image and can remember information like "The major vortex is located at the tail of the snake.", which is easier to remember and requires less mental reorganization of the data. A further advantage of attribute clouds is its intuitive use. While it is difficult to predict the final brushing when interacting with scatterplot matrices, the user has to interact with only a single structure in the attribute cloud approach. Comparing images Fig. 6 and Fig. 7, which depict the same attribute space of four variables (three components of the velocity and pressure), we think it is easier to get an idea of the distribution of the variables in high-dimensional space using attribute clouds rather than scatterplot matrices. Again, if the focus is more on quantitative results, scatterplot matrices are to be preferred.

3.2 Brushing

A snapshot of the brushing window is given in Figure 4. The window consists of two parts: The brushing area on the left and the point information section on the right. The brushing area is used to display the attribute cloud and additional statistics. The point information contains an entry for each point in the cloud and lists information on the point, such as the values of the individual variables. Thus, the user can follow two strategies for the selection of points. Either a subset in the attribute cloud can be marked or entries in the point information list can be selected. Both views are linked and if a point is selected in either of the two views, it is selected in the other one as well.

The brushing process of the attribute cloud resembles brushing of scatterplots. We provide three different brushes: a rectangular brush, a circular brush, and a lasso (polygon-strip drawn with the mouse). Using these brushes, the user can select a subset of the points of the attribute cloud. Points inside the selected region are highlighted in a different color (green points in Figure 4). The way selected regions can be manipulated adopts the style of painting programs. Pressing the control key and brushing a different region adds an additional selection. Control key combined with a mouse click deselects an already



(a) Brushing. (b) Brushed positions. (c) Brushed positions and λ_2. (d) $\lambda_2 < -3000$

Fig. 5. Draft tube: (a) Brushed attribute cloud in 2D and (b) corresponding positions in the physical domain. (c) Overlay of brushed positions and λ_2 structures. (d) Largest connected components with $\lambda_2 < -3000$.

selected point. Pressing the right mouse button over a selected point opens an additional menu that allows the user to choose a color for all selected nodes or add annotations to the selection.

3.3 Boxplots

Annotations are used to incorporate additional statistics about the selected points into the visualization. We found boxplots [23] as shown in the sample window (Figure 4) very helpful. A boxplot is a visualization of the five-number summary of a sorted set of values. The five relevant numbers are: the smallest observation, lower quartile (Q_1), median, upper quartile (Q_3), and largest observation. The quartiles and the median give values such that a certain fraction of the sorted observations is below the value and the rest is above. The partition for the lower quartile is 1/4 below and 3/4 above. For the upper quartile the fractions are exchanged. Hence, 50% of the observations lie within the range $[Q_1;Q_3]$. The median is a value, such that 50% of the observations are above and 50% are below. Before the visualization, outliers have to be computed and extracted from the boxplot. Outliers are defined as points whose distance from the mean is larger than two-third of the interquartile distance, i.e., $2/3(Q_3 - Q_1)$. The boxplots provide a quick overview over the distribution of the individual variables in the brushed subset of the attribute cloud.

3.4 Visualization of Physical Positions

After the brushing of an interesting subset of points in the attribute cloud, the corresponding positions in physical space have to be visualized. We chose two different strategies to cope with different data. If only few positions are to be visualized or additional information is to be incorporated we use a glyph-based approach. Therefore, all positions featuring values that were brushed in the attribute cloud are marked using a cube or a sphere. If larger three dimensional structures are of interest we follow a different strategy. Here, a new scalar field is created that encodes the assigned color in the attribute cloud. Afterwards, volume rendering or isosurfacing is used to visualize corresponding structures. All pictures to follow were computed using isosurfaces.

4 Results

To show the ability of the new method, we will investigate three data-sets. The draft tube and the delta wing are flow simulations. Using the draft tube, we will show how brushing of different regions of the attribute cloud is used for a fundamental investigation of the data-set. The delta wing contains different features that will be extracted in attribute space. The last example stems from climate research. We will investigate the changes in precipitation for each month of the year between two means of thirty years ([2071;2100] and [1961;1990]). Thus, we have multivariate data with twelve variables that are commonly hard to visualize. As outlook we provide a data-set from outer-space simulation which consists of 10 variables.

4.1 Draft Tube

The first data-set represents the draft tube of a Francis turbine as illustrated in Figure 5. The water enters the turbine from top and acts on the runner, causing it to spin. The runner is not illustrated in the image and sits horizontally on top of the circular inlet.

Fig. 6. Attribute cloud of the draft tube highlighted using three different variables. From left to right: -z component of velocity, norm of velocity, pressure.

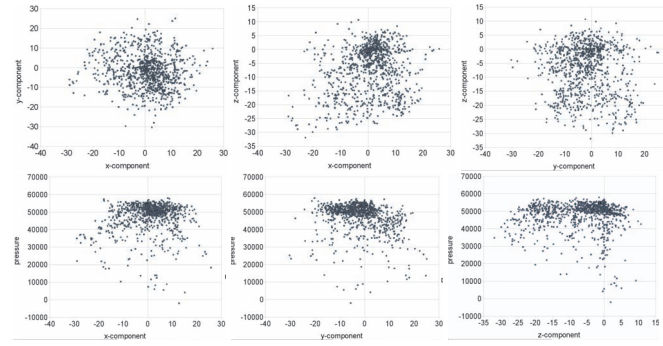


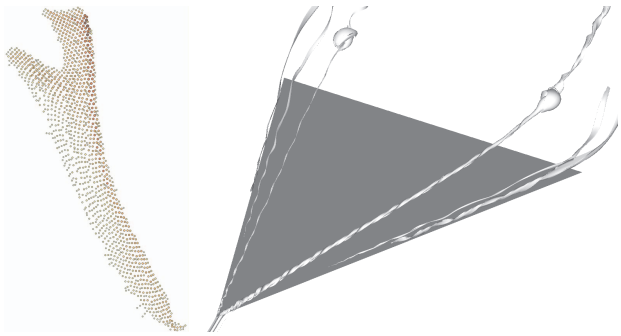
Fig. 7. Charts of the scatterplot matrix of the draft tube consisting of the following variables: (a) velocity(x), (b) velocity(y), (c) velocity(z), and (d) pressure. The scatterplots feature the following combinations: (top) a/b, a/c, b/c. (bottom) a/d, b/d, c/d

After passing the runner, the flow enters the draft tube, where it is decelerated. Thereby the kinetic energy is converted into static pressure. The flow leaves the tube through the rectangular regions in the lower left part. The data-set consists of 980,000 positions and two variables (velocity and pressure). Although this might seem rather simple, remember that velocity consists of three components (x, y, and z). Therefore, we are actually dealing with data in four-dimensional space.

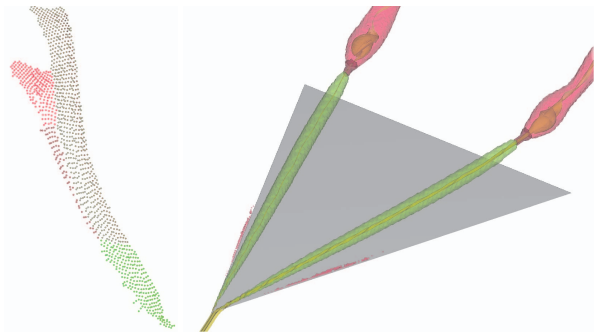
An interesting feature of the data-set is the formation of vortices. The λ_2-criterion [12] is a standard technique in vortex detection and identifies regions where the second eigenvalue of the Jacobian is smaller than zero. Visualizing an isosurface with value -0.01 covers almost the entire data-set with different intricate structures and barely anything is visible. Therefore we chose $\lambda_2 < -3000$ and visualized connected components containing more than 1000 cells in Figure 5(d) to give an impression of the data-set.

The attribute cloud of this data-set is given in Figures 5(a) and 6 – the snake we have already seen. Figure 6 illustrates different variables contained in the attribute cloud. We see that the pressure decreases along the snake and that the norm of the velocity changes transversely. Thus, there is no strict correspondence between pressure and norm of velocity. Moreover, the velocity is not as well separated as pressure indicating the turbulent structure of the data-set.

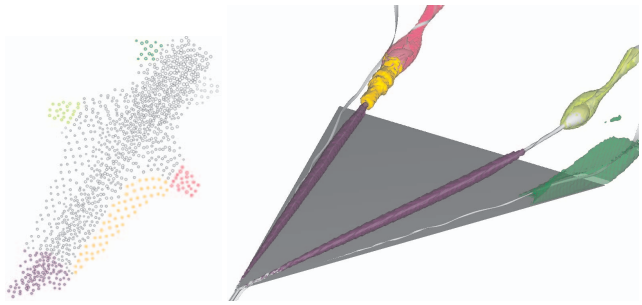
In a next step we brush several sections of the snake as illustrated in Figure 5(a). We can distinguish four major regions: the tail in magenta corresponds to a thin tube around the major vortex. The region in darker blue extends this tube. The section behind the head in light blue additionally represents several other vortex structures and finally the head, which comprises most positions of the data-set, forms the surrounding flow. Here we see that already a naive brushing of different subsets reveals the most important structures and helps the user to get a first impression of a new and maybe little understood data-set. Moreover, the distribution of the multivariate data can be investigated easily in both spaces – attribute cloud and physical domain.



(a) Original dataset: (left) Attribute cloud color-coded using number of points. (right) Delta-wing with major vortices.



(b) Variables: Norm of velocity, pressure, and density.



(c) Variables: Norm of velocity, pressure, density, x-, y-, and z-coordinate of position.

Fig. 8. Delta wing – comparison of different multivariate sets. (a) Input data-set. (b) Attribute cloud and corresponding brushed regions with three variables. (c) Analogue brushing with six variables.

4.2 Delta Wing

The EDELTA data-set represents the airflow around a delta wing at low speeds with an increasing angle of attack. Multiple vortex structures form on the wing due to the rolling-up of the viscous shear layers that separate from the upper surface. These formations of three vortices can be observed on either side of the wing (Fig. 8(a)). With increasing angle of attack the intensity of the primary vortices (vortices nearest the symmetry axis) increases until in time-step 700 a vortex breakdown occurs (bubbles at the end of the vortices). The analysis of vortex breakdown is highly interesting, as it is one of the limiting factors of extreme flight maneuvers. The grid consists of approximately 4.1 million positions. For our analysis we chose time-step 700 and the variables norm of velocity, pressure, and density and in a second visualization additionally the physical position of each point in the data-set.

Figure 8(b)(left) shows the attribute cloud of the delta wing data-set which resembles a Concorde in approach for landing. Basically the shape is similar to the one of the attribute cloud of the draft tube (snake), except for the bump/airfoil on the left. Brushing different regions of the Concorde, we get the image depicted in Figure 8(b)(right). Here we see, that the nose of the Concorde contains positions belonging to the major vortices. The airfoil of the Concorde, highlighted in pink, comprises the area around the recirculating bubbles. Points located at the tip of the airfoil correspond to the inner regions of the recirculating bubble. The isosurface of the lower fuselage (purple) isolates the region, where the major vortex merges with the recirculating bubble. Although, this part is close to the upper fuselage, we can observe a clear gap between these two central parts of the attribute cloud. This means that some multivariate configurations are very similar, but in general there are more similar configurations in other parts of the data-set and both structures are related but form separate clusters.

Figure 8(b) showed that the different structures on the left and the right hand-side feature similar values in the multivariate data. In order to separate structures at different positions in physical space, we added the position of each point in physical space to its list of multivariate data. Thus, each data point holds six variables: norm of velocity, pressure, density, and x-, y-, and z-position. The attribute cloud of this data-set is illustrated in Figure 8(c)(left). Though we newly computed the layout of this second attribute cloud, there is a strong resemblance to the previous one. Basically, the concorde was mirrored along its roof and now looks like a flying ghost. Each half of the ghost corresponds to a half of the delta wing. Brushing regions analogue to the previous example, we are now able to extract relevant structures on either side of the wing (Fig. 8(c)(right)).

This example shows that even complicated and intricate structures can be separated in attribute space. Obviously more sophisticated and specialized approaches are needed to get a fully-fledged analysis of the different "features". However, the brushing of attribute clouds is a versatile low-cost approach that can help in getting an idea, what might be present in the data-set and which special algorithms might be worth testing. Moreover, the analysis of the attribute cloud can assist in identifying new features or structures that cannot be found by standard feature extraction algorithms so far.

4.3 Climate Change

Simulations with weather and climate models produce large time dependent 3D multivariate data sets. Nevertheless, in most cases only very small subsets of the data are regularly analyzed, as the amount of information produced by such a simulations is too large to visually analyze the entire data. Commonly the analysis concentrates on visualizations of 2D slices. 2D data visualization, however, offers quite limited capabilities for the analysis of multivariate data. Only 2 or 3 scalar quantities can be visualized concurrently, e.g. by the use of overlay graphics. When more fields need to be analyzed concurrently, a visual comparison of images of the different parameters is often the only solution, although this is only a quite qualitative approach.

A similar problem arises, when spatiotemporal patterns of cyclic processes have to be visualized: most meteorological parameters show pronounced seasonal variations. In climatology, multiyear monthly means of the respective quantities are computed in order to eliminate noise caused by short term variability and hence to study the mean spatial patterns in the term of the year. Standard data analysis and visualization tool allow to interactively explore the resulting mean monthly data. Qualitatively, this can be done with the help of time animations or by a visual comparison of the 12 single visualizations. By combining groups of months to "seasons", the amount of information is reduced, but even with this simplification, the images of at least 4 fields (4 "seasons") will have to be compared visually. Until now there is no standard technique available which visually summarizes the main features in such data in one single display.

In contrast to the applications described in Sections 4.1 and 4.2, where we used our brushing technique in the parameter space, we now apply the method to only one attribute, but in the temporal space, with the aim to eventually create one single visualization which summarizes the most prominent features within the whole annual cycle.

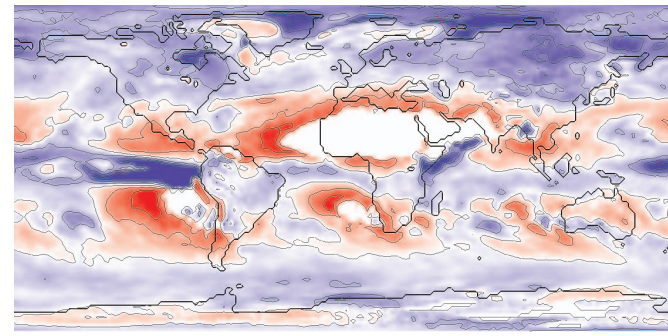
We have selected an example from a global climate simulation, more specifically the projected percental change in precipitation by the end of this century relative to the simulated precipitation for the end of the last century for IPCC scenario A1B. The data set (Figure 9) is based on a global climate simulation, a German contribution to the latest IPCC Assessment Report [25] with the coupled atmosphere-ocean-model ECHAM / MPI-OM [18] carried out by the Max Planck-Institute for Meteorology (MPI-M) at the German Climate Computing Centre (DKRZ). The data was acquired from the "World Data Centre for Climate" (WDCC) database. The spatial horizontal resolution of the 192 x 96 grid is approximately 200 km.

For the A1B scenario, the ECHAM/ MPI-OM model projects a global mean temperature increase of 3.8°C for 2071 - 2100 compared with 1961 - 1990. In this changed climate, also the precipitation will change. Precipitation and hence the availability of water is a key factor for the ecosystems as well as for our living conditions.

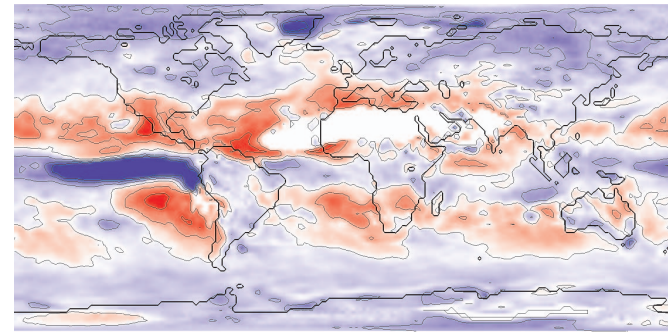
The spatio-temporal analysis of the projected precipitation changes can help us to better adapt our socio-economic system to unavoidable climate changes. Also, a better understanding and communication of the possible future changes in the climate system is a good motivation to reduce future emissions of greenhouse gases.

According to the description given above, multi-year monthly means of the precipitation were computed for the period 1961-1990 and for 2071-2100. A threshold of 3 mm/month (τ-threshold) was used in order to avoid a division by 0 when computing the percental change for each grid point and each month. Thus, we can interpret the resulting 12 2D fields as a multivariate data-set with 12 variables at each position. The mean changes for January, April, July, and October are visualized in Figure 9. Comparing these images, we see that some basic patterns can be traced, but it is hard to find a general trend or identify similar regions. If we had to compare all twelve images, the task would be even more challenging.

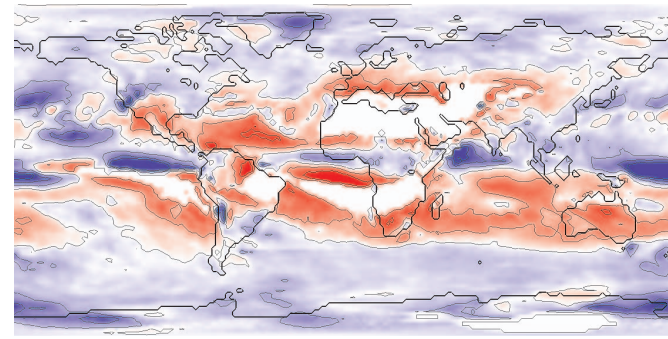
Figure 10(bottom) shows the corresponding attribute cloud for all twelve variables. The changes in precipitation range from -100% to +440%. As the 12-dimensional configurations vary a lot, we used a rather coarse discretization of 60% to receive a reasonable amount of points in the attribute cloud. The attribute cloud consists of a dense center (potato) and a boundary region with fewer points. An inves-



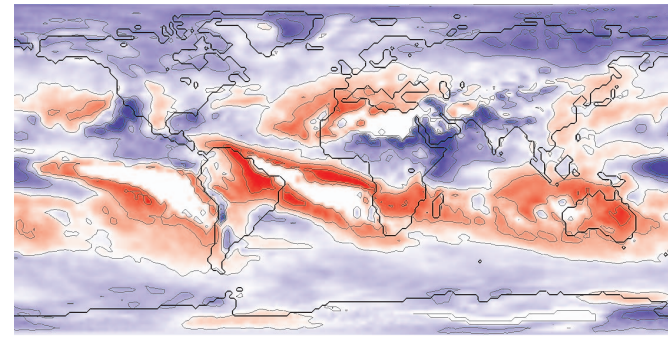
(a) January



(b) April



(c) July



(d) October

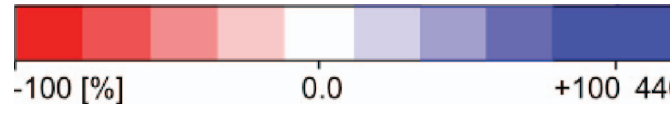


Fig. 9. Mean percental changes in precipitation for 2071-2100 compared to 1961-1990. Values < 3 mm/month have been set to 0 (τ-threshold). (a) Changes between both January means. (b) Changes for April. (c) Changes for July. (d) Changes for October.

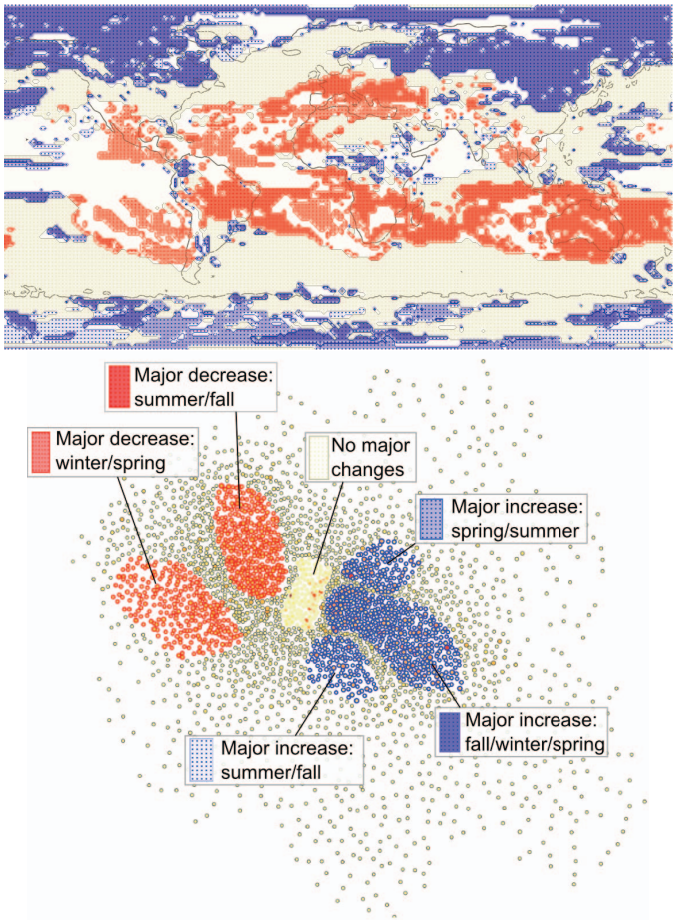
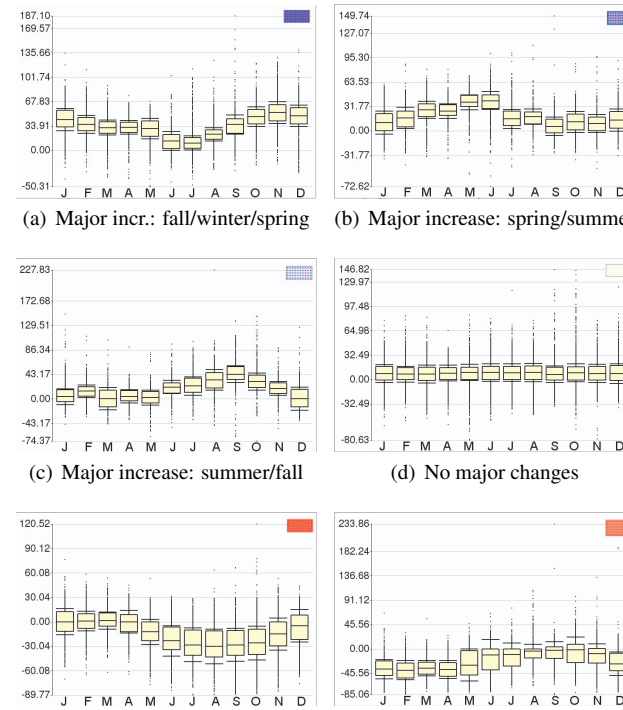


Fig. 10. Classes of change in precipitation: (bottom) Attribute cloud highlighting six subclasses. (top) Corresponding areas in the data-set.



(a) Major incr.: fall/winter/spring (b) Major increase: spring/summer

(c) Major increase: summer/fall (d) No major changes

(e) Major decrease: summer/fall (f) Major decrease: winter/spring

Fig. 11. Boxplots of different classes of average change in precipitation in percent [(2071-2100) - (1961-1990)]. Values < 3 mm/month have been set to 0 (τ-threshold). (top) Major increase in precipitation in different seasons (seasons on northern hemisphere). (d,e) Major decrease in precipitation in summer/fall and winter/spring. (f) Throughout the year no major changes can be observed.

tigation of different areas in the cloud using boxplots revealed that the data-set consists of many different combinations of the multivariate data. For example we can find areas that feature a constant decrease or increase in precipitation throughout the year, areas that face changes only during certain months, and areas where the precipitation decreases in one season and increases in another one. Many configurations in-between exist as well and result in the compact shape of the attribute cloud. In our investigation we found six major structures, which are highlighted in Figure 10(top). The left half of the potato contains points that represent positions where the precipitation mainly decreases, and the right one positions with increasing precipitation. In the center of the potato the changes in precipitation are rather low compared to the entire data ($< 15\%$). The decreasing half (red) is divided into two regions: decreasing precipitation from December until May (winter/spring) and decrease from June until November (summer/fall). The patterns for increasing precipitation (blue) are more difficult and we distinguish three categories: increase from August until May (fall/winter/spring), increase from February until June (spring/summer), and increase from June until October (summer/fall).

The corresponding boxplots are given in Figure 11. The large numbers of outliers show that clustering climate data is rather difficult and most points included in a cluster contain one or more months that do not match perfectly. Nevertheless, a clustering is necessary to cope with the giant amount of information and to get an overview over basic trends. Brushing the attribute cloud easily revealed different spatiotemporal patterns that occur in the data-set. Although the brushing process is not fully automatic and requires some guidance by the domain expert, the method easily yields a nice compression of the most important information contained in the data as shown in the final visualization in Figure 10(top).

4.4 Outlook – Star Formation

As outlook we chose a large-scale data-set from an outer-space simulation investigating the formation of ionization front instabilities, which are responsible for a variety of phenomena in interstellar medium, e.g. the formation of stars. The data-set is one time-step (100) of the data provided for the IEEE Visualization 2008 contest and consists of 36 Mio. cells and 10 variables. Figure 12 shows the corresponding attribute cloud color coded using 5 of the 10 variables. Brushing the region featuring highest density ($log(density) > 3.2$) (Fig. 13(left)) reveals the shadow instability shown in Figure 13(right). Now the user can interactively explore the effect of the different variables on various regions of the instability and get an idea how certain combinations of variables interact.

4.5 Timings

All timings are given in the form minutes:seconds. Preprocessing comprises the discretization and the computation of the EMST. Currently, the layout takes longest, which can be easily improved using a GPU-based implementation of the algorithm. The brushing of the attribute clouds is interactive and the timing depends only on the visualization technique. Using GPU-based volume rendering, we achieve an interactive presentation of the isosurfaces. System information: operating system - Linux, language - C++, processor - 1 CPU of an AMD Opteron Quad-Core (2,110 MHz), RAM - 31.5 GB. The memory overhead for the algorithm is rather small, as the attribute clouds usually contain less than 5000 points to prevent occlusion.

Data-set	nPositions	nVar	Preprocessing	Layout
Draft tube	0.98 Mio	4	0:04	2:36
Delta wing	4.1 Mio	6	0:05	3:02
Climate	18,000	12	0:11	2:58
Star	36 Mio	11	1:53	3:14

(a) log(Density) (b) log(Gas temperature) (c) H mass abundance (d) He mass abundance (e) H_2 mass abundance

Fig. 12. Attribute cloud of the ionization front colored according to 5 of the 10 different variables.

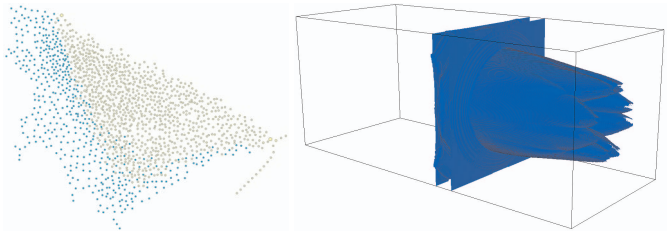


Fig. 13. Brushing of the star formation data-set using highest density.

5 Conclusion

In this paper we proposed brushing of attribute clouds. This technique transforms multivariate data to 2D resulting in a planar point cloud, called attribute cloud. Our method builds upon solid and well established techniques such as scatterplots, linked views and brushing. However, the transformation of the multivariate attribute is a new approach that allows for an intuitive analysis of many variables. The transformation is based on ideas from multivariate statistics and manifold learning and therefore has a sound theoretical basis. Using multivariate and, to some extent, unsteady data, we showed that both types of data can be easily investigated. Complex correlations and coherencies in the high-dimensional multivariate attribute space can now be explored by the user in 2D.

Although we were able to extract relevant structures in all data-sets that we investigated, there are still a few aspects that might be improved. For the layout of the attribute cloud, we chose the Fruchterman-Reingold algorithm. More advanced graph drawing procedures exist, which might help with data that is very dynamic (e.g. climate data) to reveal more meaningful structures. Moreover, the layout is currently the slowest part of the algorithm ($\sim$ 3 min). This could be improved using a parallel or GPU-based implementation. The climate data-set revealed two further aspects that might be further improved. If there are plenty of different combinations of variables, the discretization needs to be rather coarse to get a reasonable amount of points in the attribute cloud. Density-driven Voronoi tessellation as proposed in [11] might be a good choice to find representatives. Furthermore, the application of attribute cloud visualization to unsteady data is an interesting topic that needs further research.

6 Acknowledgments

The authors would like to thank Christoph Garth for helpful discussions, VA Tech Hydro and Ronny Peikert for providing the draft tube data-set, and Markus Rütten (DLR) for the delta wing data-set. The graph library which was of great help was implemented by Christian Heine.

References

[1] R. A. Becker and W. S. Cleveland. Brushing scatterplots. *Technometrics*, 29(2):127–142, 1987.
[2] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.
[3] H. Chen. Compound brushing explained. *Information Visualization*, 3(2):96–108, 2004.
[4] W. S. Cleveland. *Visualizing Data*. Hobart Press, Summit, New Jersey, 1993.
[5] H. Doleisch and H. Hauser. Smooth Brushing for Focus+Context Visualization of Simulation Data in 3D. In *WSCG*, pages 147–154, 2002.
[6] T. M. J. Fruchterman and E. M. Reingold. Graph Drawing by Force-directed Placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.
[7] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Navigating hierarchies with structure-based brushes. In *INFOVIS '99: Proceedings of the 1999 IEEE Symposium on Information Visualization*, page 58, Washington, DC, USA, 1999. IEEE Computer Society.
[8] H. Hauser, F. Ledermann, and H. Doleisch. Angular brushing for extended parallel coordinates, 2002.
[9] C. Henze. Feature detection in linked derived spaces. In *VIS '98: Proceedings of the conference on Visualization '98*, pages 87–94, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.
[10] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–92, 1985.
[11] H. Jänicke, M. Böttinger, X. Tricoche, and G. Scheuermann. Automatic Detection and Visualization of Distinctive Structures in 3D Unsteady Multi-Fields. *Computer Graphics Forum*, 27(3):767–774, 2008.
[12] J. Jeong and F. Hussain. On the identification of a vortex. *J. Fluid Mech.*, 285:69–94, 1995.
[13] I. T. Jolliffe. *Principal components analysis*. Springer, 2002.
[14] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, 1979.
[15] 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.
[16] O. Nevalainen, J. Ernvall, and J. Katajainen. Finding minimal spanning trees in a euclidean coordinate space. *BIT Num. Math.*, 21(1), 1981.
[17] J. C. Roberts and M. A. E. Wright. Towards ubiquitous brushing for information visualization. In *IV '06: Proceedings of the conference on Information Visualization*, pages 151–156, Washington, DC, USA, 2006. IEEE Computer Society.
[18] E. Roeckner et al. *The atmospheric general circulation model ECHAM 5. PART I: Model description*. Tech. Rep. 349, Max Planck Inst. for Meteorol., Hamburg, Germany, 2003.
[19] R. Spence. *Information Visualization*. Addison Wesley, 2000.
[20] W. Stuetzle. Estimating the cluster tree of a density by analyzing the minimal spanning tree of a sample. *Journal of Classification*, 20(5):25–47, 2003.
[21] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.
[22] X. Tricoche, C. Garth, G. Kindlmann, E. Deines, G. Scheuermann, M. Ruetten, and C. Hansen. Visualization of intricate flow structures for vortex breakdown analysis. In *VIS '04: Proceedings of the conference on Visualization '04*, pages 187–194, Washington, DC, USA, 2004. IEEE Computer Society.
[23] J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, 1977.
[24] P. C. Wong and R. D. Bergeron. Multiresolution multidimensional wavelet brushing. In R. Yagel and G. M. Nielson, editors, *IEEE Visualization '96*, pages 141–148, 1996.
[25] Working Group I contribution to the Fourth Assessment Report of the IPCC. IPCC AR4: Climate Change 2007 - The Physical Science Basis. Technical report, IPCC, 2007.