

Interactive Focus+Context Visualization with Linked 2D/3D Scatterplots

Harald Piringer, Robert Kosara, Helwig Hauser
VRVis Research Center
Vienna, Austria

<http://www.VRVis.at/>
Piringer@VRVis.at, Kosara@VRVis.at, Hauser@VRVis.at

Abstract

Scatterplots in 2D and 3D are very useful tools, but also suffer from a number of problems. Overplotting hides the true number of points that are displayed, and showing point clouds in 3D is problematic both in terms of perception and interaction.

We propose a combination of 2D and 3D scatterplots, together with some extensions to overcome these problems. By linking 2D and 3D views, it is possible to interact in 2D and get feedback in 3D. That feedback is also enhanced by depth cues (color and point size) such that the user gets a better depth impression. Histograms in 2D and 3D show additional information about point densities and additional context can be displayed. An example application demonstrates the usefulness of the technique.

Keywords—Scatterplots, Linked Views, Brushing

1. Introduction

Simple 2D scatterplots are a very versatile and useful tool, used in many applications. The two-dimensional provides an overview and understanding, and lends itself well to direct interaction with two-dimensional input devices (e.g., the mouse).

This simplicity and ease of use hides a number of problems, however. In a large dataset, many points may be plotted onto the same pixel, without the user being able to tell how many. This makes it impossible to judge the true distribution of data from a scatterplot. It is also hard to find structures that exist in more than two dimensions, which has prompted the development of extensions such as the scatterplot matrix [5].

3D scatterplots [2, 20, 21] solve only a part of this problem. There is an additional dimension in which structures can be separated, and the overplotting problem is also reduced. Interaction is harder in 3D, however, and also the display of points in 3D is quite problematic. Current implementations usually lack sufficient depth cues, so judging the

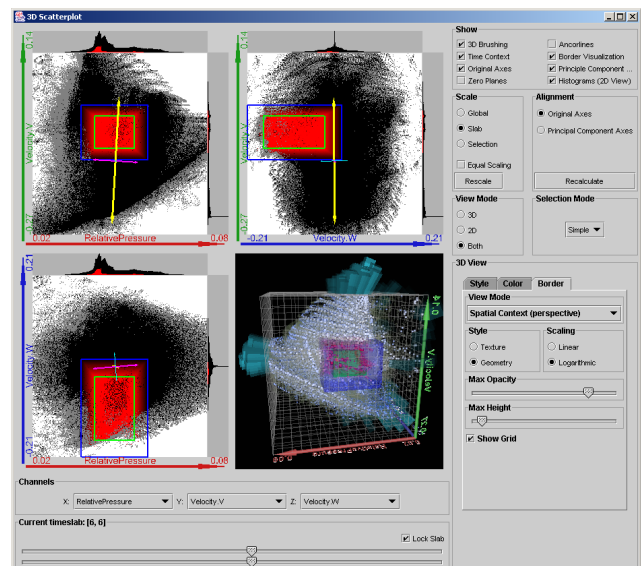


Figure 1. The combined 2D and 3D scatterplot with user interface.

three-dimensional structure of the points is very difficult. Many implementations of 3D scatterplots also use volume rendering or binning of values into larger objects, which makes efficient rendering easier, but also creates a number of problems: Which value is represented when many different values are present in a bin? How do we show the number of values in the bins?

The contribution of this paper is as follows: First, we introduce halos and depth-dependent point size as novel depth cues in the context of 3D scatterplots, and we use histograms as technique to highlight the point distribution and density inside and outside the spatial focus. Second, we propose a linked setting of the 3D scatterplot with three 2D views, which combines the advantages of both: Interaction is done in 2D, with the results shown in 3D. Third, we present a case study that illustrates the usefulness of the combined approach and the novel techniques.

The combined 2D/3D scatterplots were implemented as

part of SimVis [7], a system that uses linked views extensively for the display of high-dimensional data from computational flow dynamics (CFD) simulations. The 3D scatterplot displays the points directly, instead of using binning, and exploits the capabilities of modern graphics hardware to provide interactive frame rates even for large datasets.

2. Related Work

2D scatterplots are a very old and well-known visualization method for unstructured data. They and their 3D counterparts are available in many statistics and maths packages, e.g., SPSS, R, MatLab, etc.

A number of extensions have been developed for 2D scatterplots to deal with higher dimensional data. Prosecution views [10] allow the user to select a range of values on one or more axes that are not shown – only the points that lie in this range are projected. SimVis uses a selection of values on the time axis in a similar way.

Another extension is the scatterplot matrix [5], which consists of many scatterplots that show all combinations of two dimensions from the data space. They are positioned in a matrix so that all plots with the same Y axis are in one row, and all plots with the same X axis are in the same column. This idea is partly used in the layout of the combined 2D/3D scatterplots, for a set of three dimensions.

The combined 2D/3D scatterplot – as well as the whole SimVis system [7] – uses multiple, linked views [1], which is one of the key ideas for working with high-dimensional data. In order to make use of multiple views, the user can interact with them by selecting parts of the data to be investigated further. This technique is called brushing and is usually done directly on the visible structures within the view. The goal of brushing is to assign a scalar value representing its degree of interest (DOI) [11, 23] to each data entry. A classic example for linking and brushing is WEAVE [15], which allows the user to see different views like scatterplots, histograms, and a 3D rendering of an object and brushing in the 2D displays. More recently, Groth and Robertson [16] present an integrated database visualization system that supports 2D and 3D scatterplots. Another example is the XGobi system [24], which provides interactive methods for the manipulation of various data views including high-dimensional scatterplots. If the DOI is not restricted to binary values, brushing is said to be smooth [8].

Several variants of 3D scatterplots have been proposed [2, 20, 21], which all use volume rendering for display. In contrast to the statistics packages mentioned above, volume renderers usually provide fast rendering, but also limit the possible interactions. Volume renderers also usually use parallel projection, which further limits depth perception.

3. Extending 3D Scatterplots

Before we discuss in detail our way of interactively linking 2D and 3D scatterplots (section 4) and then apply them exemplarily in combination with multiple linked views to explore a large dataset (section 5), this section proposes various extensions to standard 3D scatterplots; these extensions address common problems and succeed in overcoming certain known restrictions, like a compromised perception of depth and point density.¹

3.1. Improving Depth Perception

A common problem of conventional 3D scatterplots is the loss of depth information after the projection onto the 2D output device. Interacting with the viewpoint can partly compensate this shortcoming due to the different movement speeds of the displayed points (known as motion parallax); yet if still images are requested, this solution is not applicable. We address this problem by changing two attributes of the data points depending on their distance from the viewpoint – size and color – and propose halos to outline the shape more clearly (see Fig. 2 for an example).

Size is one of the primary natural depth cues [13], because the human brain is used to the fact that distant objects appear smaller, as they occupy less space on the retina of the eye. As an independent visual dimension (like color and position), size is also well-suited for representing depth in 3D scatterplots: Decreasing the point-size with increasing distance from the viewer enhances depth perception considerably. The size can either drop off reciprocally with the distance – mimicking a perspective projection in a mathematically correct way – or decrease linearly from a maximum to a minimum inside the 3D scatterplot, which permits a better discrimination in point size of the distant parts (which we found more useful in the context of our application).

Although varying the point-size obtains a convincing depth-impression for sparse areas, this effect levels off as soon as clusters of similarly colored points cover large, almost monochromatic spots on the output device, where single data entries can not be distinguished any more. A solution is to outline the shape of single points by drawing halos [17]: As a technique known from painting, accentuating the outline can help to intuitively indicate the presence of depth discontinuities between contiguous elements in a projection. This comes at the cost of a slightly increased amount of occlusion, which we found hardly disturbing, since only the foremost front of a big cluster is visible anyway. Technically, each point is surrounded by a thin, semi-transparent circle of the same hue as the original point-color

¹ For a fully colored electronical version of this paper and a video demonstrating the combined scatterplot, see: <http://www.vrvis.at/vis/research/2d3dscatterplots/>

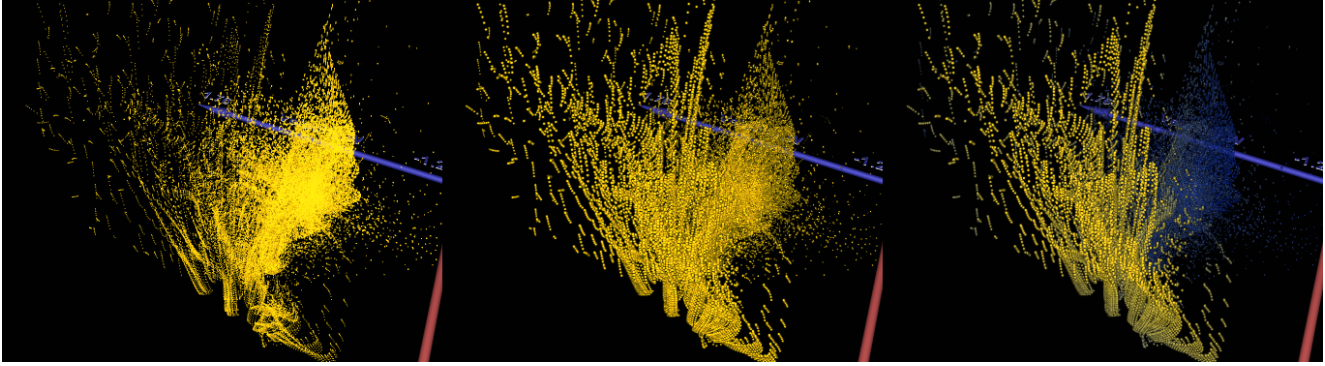


Figure 2. Improving depth perception. Left: No depth cues are used; Middle: Depth is indicated with point size and halos are used to ease the discrimination of single points; Right: Depth cueing using both color and point size, as well as halos.

but with a much lower brightness. As illustrated in the middle of Fig. 2, varying the point-size in combination with halos conveys a three-dimensional impression and permits to make out single entries even in dense areas – unlike in the left image, where neither depth cues nor halos are used.

Apart from size, depth can also be mapped to color: Altering the hue and contrast of scene elements that are farther away from the viewer is a technique known from painting and rendering and is a frequently used sort of depth cueing [12]. Due to characteristics of the human visual system, bright and warm colors (like red or yellow) are suited to indicate proximity to the viewer, while dark and cold colors (blue or grey) are intuitively associated with depth. This optical phenomenon is referred to as chromo-stereoscopy [26]. Of course, color can be used to encode information other than depth as well: Our implementation for example permits mapping an arbitrary dimension of the dataset to color using transfer functions, which is a common and essential functionality for 3D scatterplots. However, mixing various coloring schemes – although technically possible – is most often confusing and usually not recommended.

Depth cueing with point size, color and halos can be done independently or in combination, the latter yielding the most convincing depth perception (see Fig. 2, right). In order to achieve interactive frame rates for a large number of points, modern graphics hardware should be used for modifying point size and color as well as for generating halos. Vertex programs have proven useful for these tasks if the 3D scatterplot is point-based (as in our case), stored as a set of independent vertices. Hardware-based fogging is another option, yet applies to color only and provides less flexibility than vertex programs; Our implementation uses one vertex program for all tasks, since the same interpolation factor can be applied to both point size and color, which is efficient and guarantees consistency.

3.2. Representing Point Density

The approximate point density at a certain position is important information when analyzing characteristics of a dataset. However, scatterplots (2D and 3D) represent density in a satisfactory way only as long as hardly any points are ultimately projected to the same pixels (either due to the limited screen resolution or because the considered data is identical for several entries). Adapting the scaling does not solve the problem of identical data and comes at the cost of losing the overview. Our approach is to incorporate two-dimensional histograms at each border plane of the cube where the 3D scatterplot is drawn. Each rendered data point (lying inside the cube) is orthographically projected to each border plane. The 2D distributions are discretized by equally sized squares (called bins) in accordance with the desired resolution. A user-definable scaling is applied to the bin-counts before any further visualization: Linear scaling highlights peak densities while logarithmic scaling permits to make out slight differences in sparse areas.

As actual depiction of the 2D histograms, we propose either a geometric representation using one axis-aligned cuboid per bin (Fig. 3, right cube), where the height reflects the respectively scaled bin-count, or a texture-based flat visualization using one quadratic texture per border plane matching the resolution of the binning with nearest-neighbour filtering (Fig. 3, middle cube). In both cases, the scaled bin values are mapped to the opacities of the cuboids or texels, respectively, with a user-definable maximum opacity. Additional color coding is possible, but may cause problems due to too much visual complexity. Concrete implementations must take into account that transparent geometry requires a view-dependent back-to-front rendering order.

Although both visualization methods can not solve the problem of unrecognizable point densities inside the scatterplot itself and allow for rather approximate assessments only, they provide very useful information and contribute to a better understanding of the data. Chapter 4.2 adapts this

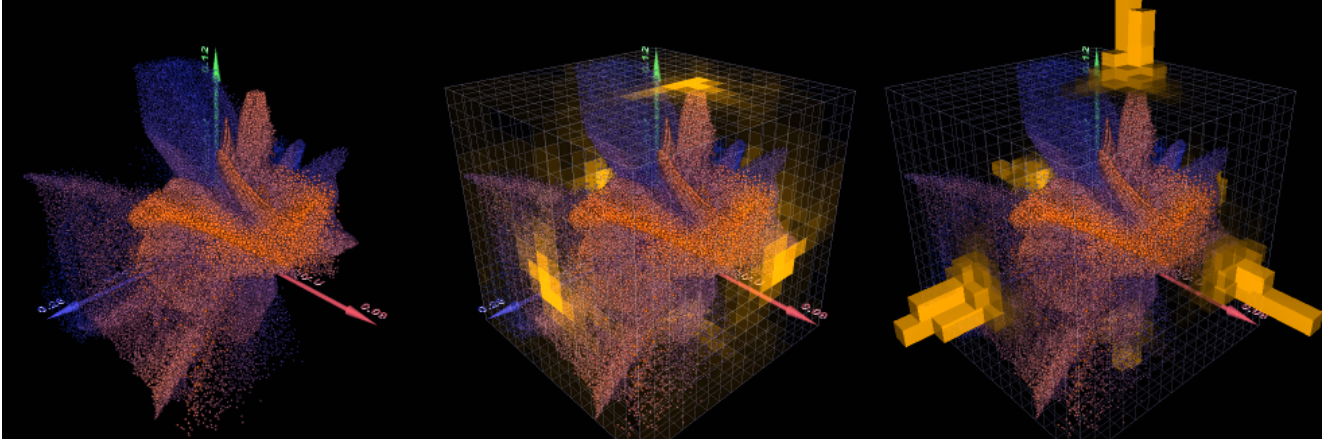


Figure 3. Representing the point density with transparent 2D histograms: The density is not clearly recognizable in the left third. In the middle, a texture-based representation uses opacity only, while the right third illustrates mapping density to opacity as well as the height of axis-aligned bins.

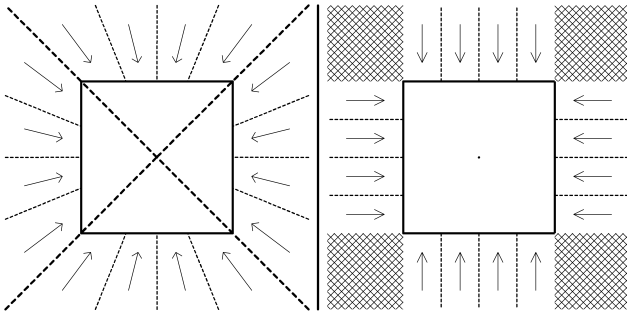


Figure 4. Projecting the spatial context perspectively (left) or orthographically (right). The hatched space is not captured by the orthographic projection. Note the unequal binning when equally subdividing the angle instead of the plane with the perspective projection.

technique for 2D scatterplots where we face similar problems in representing the point density.

3.3. Spatial Context Information

When analyzing large datasets with many thousands of points, it is sensible to allow for zooming into the data by showing only a cubic cutout of the whole scatterplot (the spatial focus). The points outside this cube (the spatial context) are not rendered by default. In order to avoid deteriorating the overview, we discuss a visualization of this context similar to the way as we deal with point densities in chapter 3.2: After scaling the data as currently chosen, all points of the spatial context are projected onto the border planes of the cube as described below and binned according to the desired resolution. We perform either linear or logarithmic scaling before the results are mapped to opacities and displayed using a texture or geometry-based transparent representation.

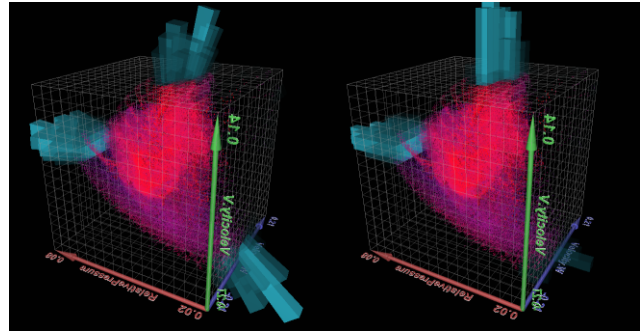


Figure 5. Representing the spatial context: The perspective projection (left, drawn using projection-aligned transparent bars) captures the whole spatial context, while the orthographic projection (right) omits parts of it (the lobe in the bottom right corner for example).

We propose a perspective projection towards the center of the cube of all points, which are outside the spatial focus (see Fig. 4 left for a 2D sketch): Each border plane serves as view plane for all points of the spatial context seen from the cube center using a quadratic view frustum with a field of view of 90 degrees. This approach strongly resembles the way cube maps [14] are generated in the field of real-time rendering (for instance for environment mapping) and it serves a similar purpose.

For binning, the angle of the field of view (rather than the view plane) should equally be subdivided in order to make sure that each bin covers an equal amount of space: Each bin represents the number of points of the spatial context in a certain direction. In order to intuitively indicate the direction captured by one bin, we propose to use projection-aligned bars as geometric representation (Fig. 5, left). Using non axis-aligned geometry for a perspective projection also renders it easier for the user to visually distinguish be-

tween the various visualization modes. The scaled bin values are again mapped to the heights of the bars. It may possibly seem more intuitive to place the bars of the perspective projection on a sphere, but we have decided against this representation as it differs from the shape of the actual spatial focus and leaves a gap in between. A texture-based representation is somewhat problematic, since the unequally sized bins mismatch the equally sized texels. A simple solution is to draw one correctly sized flat quad per bin instead.

Another option is to use an orthographic instead of a perspective projection. However, without extensions, this is applicable only to such data entries that are outside the spatial focus in not more than one considered dimension; These points are projected onto the nearest border plane as illustrated in the right of Fig. 4. The result can be equally binned and visualized as texture or using axis-aligned cuboids (Fig. 5, right). Although this method is simpler, we still favor the perspective projection, since it captures the whole spatial context: Neglecting the parts of the context which are outside the focus in two or three dimensions may be very misleading and is not a satisfactory visualization. However, this flaw could possibly be overcome by adding a separate context representation for the non-captured space.

3.4. Temporal Focus–Context Discrimination

The SimVis system [9], which the 3D scatterplot is part of, is designed for analyzing flow simulation results over time, thus the dimension *time* is essential in all views and should always remain within the attention of the user. In addition to mapping the dimension *time* to the axes of the scatterplot just like any other, which virtually decreases the number of freely assignable dimensions, the view provides a range slider (somewhat similar to projection views [10]) as a means of specifying a certain time span of interest – called temporal focus – within the overall duration of the simulation: Only this focus is subject to any kind of interaction and spatial context visualization. We refer to this kind of focus–context discrimination as temporal focus–context discrimination, though this does not restrict the generality of the principle visualization, since applying it to arbitrary dimensions is imaginable (e.g., ‘gender’ when dealing with demographic datasets).

The temporal context is optionally rendered as transparent grey points – analogous to the visualization in other SimVis views [7] and clearly discernible from the opaquely colored points of the temporal focus (see Fig. 6). Since the rendering order is important for all transparent objects, we suggest drawing the entire temporal focus before the temporal context with the depth-test enabled: This ensures that the focus remains visible even behind the context. The user can define the point sizes separately for focus and con-

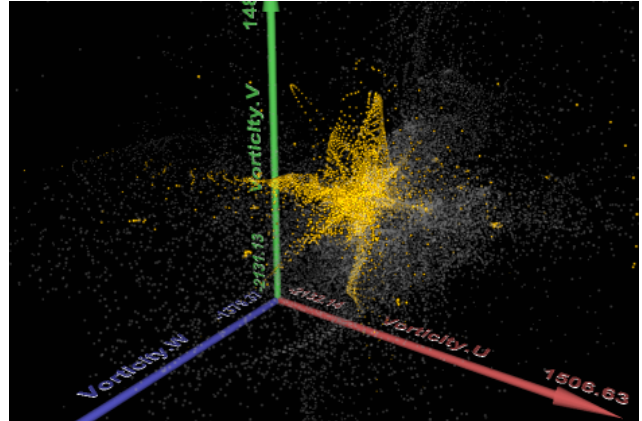


Figure 6. Displaying the temporal context as grey points.

text permitting to highlight one or the other. Halos and depth-dependent point sizes as techniques for depth cueing (see 3.1) are applicable to the context as well.

3.5. Displaying Principle Component Axes

An important reason for drawing scatterplots is to relate two or three dimensions in order to visually detect potential correlations. A mathematical way to deal with correlations between dimensions is to perform a principle component analysis [18]: Transforming (possibly) correlated variables of an n -dimensional data space into n uncorrelated variables with decreasing variability yields an orthogonal basis of the data space. The axes of this basis are ordered by the amount of variance the data shows in the respective direction. This is valuable information when exploring the characteristics of the dataset and is thus (optionally) visualized: The principle components are calculated for the spatial and temporal focus of the three dimensions mapped to the axes of the scatterplot in scatterplot space and displayed as 3D arrows (see Fig. 9 for an example). We use the mean values of all considered data entries as origin of the obtained coordinate system and scale the displayed arrows in accordance to the absolute values of the respective eigenvalues, whose computation is an intermediate step in the overall calculation of the principle components.

4. Interactively Linking 2D and 3D Scatterplots

The main advantage of 3D scatterplots is that one more dimension is simultaneously displayed. However, 2D scatterplots are much more widely used and thus much more familiar to the majority of users. Besides, the two dimensions of the mouse as standard input device match the dimensionality of 2D views, which makes interaction with the data much easier and more intuitive. This section proposes a

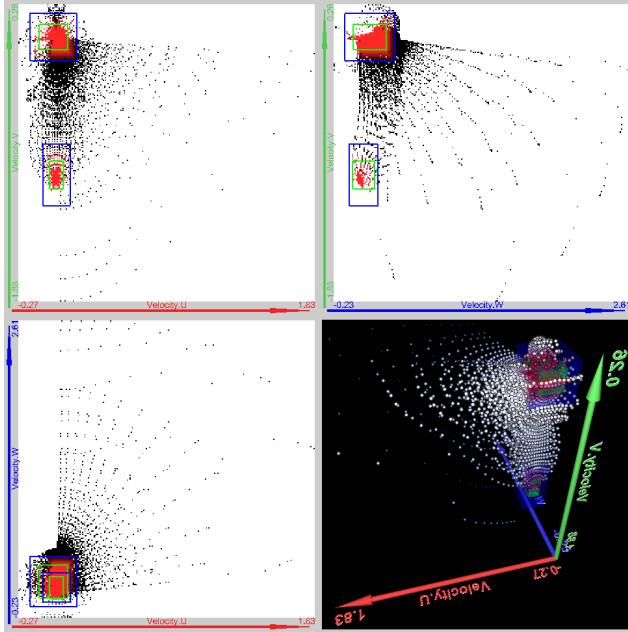


Figure 7. Composite smooth brushing in the combined 2D/3D View: Two cuboid-shaped basic brushes are logically OR-combined. DOI coloring is applied in all views. The basic brushes are drawn as rectangles in 2D and as transparent boxes in 3D. Note the arrangement of the views recognizable by the color-coding of the axes.

combination of 2D and 3D scatterplots and discusses linking and brushing [3] in this context.

4.1. Assisting 3D Viewing with 2D Scatterplots

A common way to complement 3D viewing with 2D views in commercial modelling applications is to add three 2D views, each showing an orthographic projection of the scenery for the X-, Y- and Z-axis, respectively. Applying this approach to 3D scatterplots yields a spreadsheet forming a simple 2D scatterplot matrix [5]. This matrix displays every combination of two data dimensions mapped to the three axes of the 3D scatterplot (see Fig. 7). The arrangement is important, as neighboring views assign the same dimension to the common edge to ease comparisons. In order to attain a consistent multi-viewing, changes to any relevant parameter (e.g. axis-mapping, scaling and so on) take immediate effect in all views. An in-depth discussion about using linked multiple views in a spreadsheet setting is provided by Chi et al. [4].

Apart from providing more familiar 2D scatterplots as a purpose on its own, a significant advantage of the combined approach is the ability to easily define axis-aligned 3D brushes in a 2D environment. Although conceptually identical to 2D brushes, it is important that the brush con-

cept matches the view layout: Using the feature specification as proposed by Doleisch, Gasser and Hauser [7], a brush defined by the combined scatterplot is a logical AND-combination of selections on all three dimensions of the dataset which are currently mapped to the axes. Such a brush (referred to as simple brush) represents an axis-aligned cuboid in 3D and a rectangle in 2D. Due to the view layout, the boundaries of a brush are collinear in neighbouring 2D views. Simple brushes are created by dragging the mouse in any 2D view, where the two dimensions mapped by this view are constrained according to the user input, while the third (hidden) dimension is initially defined by the size of its spatial focus. After creation, simple brushes can be moved and resized in any 2D view which takes immediate effect in all views. We only support axis-aligned brushes, because brushes of this shape do not implicate any correlation between the dimensions [7]. However, the user can define arbitrarily complex composite brushes by combining simple (and composite) brushes using logical AND or OR combinations [7].

We also support smooth brushing [8]: The user can split the border of a simple brush into independently modifiable interior and exterior boundaries, which causes the according degree of interest (DOI) function to drop off linearly from 1.0 inside the interior boundary to 0.0 at the outer boundary. A visualization of the DOI in 2D scatterplots is essential: Brushes defined by the combined scatterplot constrain three dimensions and thus one more than can be shown by a single 2D view; However, 2D rectangles as 2D outlines of brushes do not provide any information concerning the depth-validity of the brush, and even points inside a rectangle in 2D can lie outside the brush in 3D; therefore the coloring of the points should reflect their DOI in order to facilitate a correct understanding of the current brushing situation. Among possible visualization options for the DOI are mapping the minimum, maximum or average of the DOI values of all data entries, which project to a certain pixel, to color. Additionally, providing coloring based on the DOI in the 3D view can further enhance the comprehension of the current brushing. An example for a composite smooth brush with DOI coloring in the combined 2D and 3D scatterplot is shown in Fig. 7.

4.2. Adapting 3D Extensions for 2D Scatterplots

Some extensions of 3D scatterplots as presented in chapter 3 are also applicable to 2D scatterplots in a slightly modified version. This is because 2D and 3D scatterplots have some common drawbacks: Due to the loss of one dimension compared to 3D scatterplots, the problem of unrecognizable point densities is usually even worse in 2D. Analogically to the 3D case (see section 3.2), we propose depicting the spatial focus with two histograms per view –

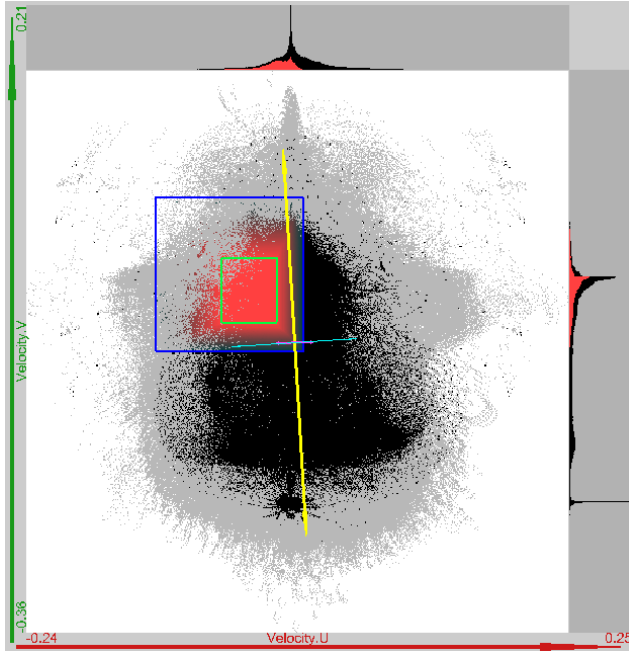


Figure 8. Extensions for 2D scatterplots: Histograms indicate the point density of the whole focus and the brushed focus and grey areas represent temporal context.

for the X- and Y-axis, respectively (see Fig. 8): Located at the margin of each view, they share its resolution. Since the amount of values actually brushed is often hard to make out in 2D, brushing can additionally be considered by drawing a co-located and equally scaled histogram in a different color, counting each entry according to its current DOI value (which fits well with smooth brushing). It should be mentioned that point densities in 2D scatterplots could also be visualized using alpha-channel transparency [25, 6], but this has not been implemented here.

The principle component axes can be shown in 2D views as well (see section 3.5): Either an independent 2D analysis can be performed for each view, obtaining three different 2D coordinate systems (one for each 2D view), or the 3D axes computed by the analysis in 3D space can be projected orthographically to 2D space. The former approach provides more information for the 2D views themselves, while the latter is more consistent with 3D viewing.

Analogous to the 3D case, 2D scatterplots also permit visualizing the temporal context as described in chapter 3.4. Since the perception of the respective focus must not be compromised, we suggest drawing the focus on top of the context, hiding it in places where both focus and context can be found and using clearly discernable colors for both. Fig. 8 illustrates a combined application of the proposed 2D extensions.

4.3. Linking External Views

The SimVis system provides linked views of different kinds in general, and multiple instances of combined scatterplots in particular. This general linking between views differs from the way, how the views of one instance of the combined scatterplot are bidirectionally linked, as described in chapter 4.1: Brushes which are defined in one view can not be modified in an external view, since any interaction depends very much on the concrete visualization as well as the subset of displayed dimensions, which are both likely to mismatch in general. However, it is possible to import brushes defined in external views for visualization purposes only, which is crucial as it allows users to identify brushed features with different visualization techniques and axis-mappings. Furthermore, composite brushes can be defined by logically combining brushes of different views [8, 7]: It is therefore helpful to distinguish between local brushing (the combination of all brushes defined in a single view) and global brushing (actually brushed data points considering the overall composite brush). To allow for this distinction, different colors are assigned to locally and globally brushed data points (see Fig. 13 and Fig. 14 for examples).

5. Application Scenario

This chapter demonstrates the application of combined 2D and 3D scatterplots and other linked views for interactively exploring and analyzing a large dataset. It puts special emphasis on using the extensions proposed by this paper as well as the aspect of linking with other views of the SimVis software. Although this case study investigates time-dependent simulation results, the extensions and linking techniques are also applicable for general information visualization (InfoVis) datasets, as illustrated in Fig. 9. It demonstrates the evolution of the attributes weight, miles per gallons and horsepower of cars in the years 1970 to 1982, showing a clear tendency towards lighter and weaker cars with less gas consumption. For a fully colored version of this paper and animation sequences, please refer to <http://www.vrvis.at/vis/research/2d3dscatterplots/>.

5.1. The Application Setting

Our collaboration partner belongs to the field of the automotive engineering industry, where results from computational flow simulation are analyzed, which is a challenging task. Simulations are time-consuming and typically many simulation-cycles are required to optimize the performance of a specific system. In order to speed up these simulation-cycles and in turn also shorten development times, interactive visualization is crucial to achieve fast and

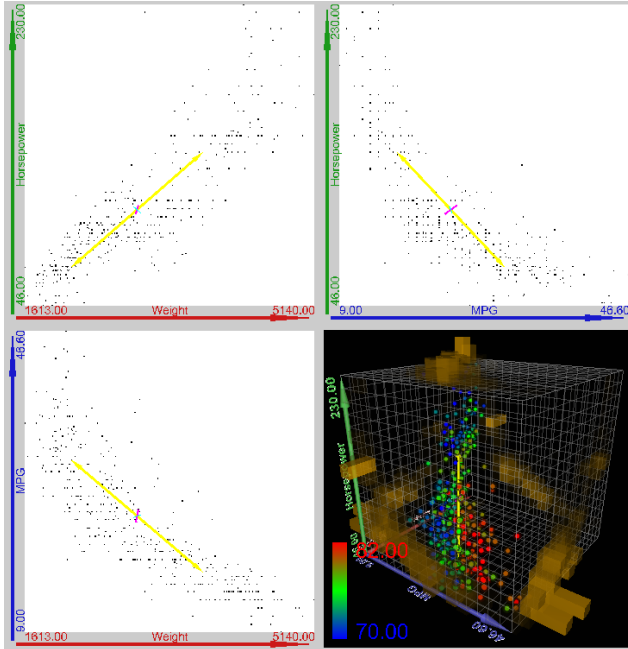


Figure 9. An InfoVis dataset: The attribute 'year' is mapped to color, the attributes 'weight', 'horsepower' and 'miles per gallon' are mapped to the axes. The yellow line is the first principle component axis, the other two axes are hardly visible as correlation occurs mostly in one direction.

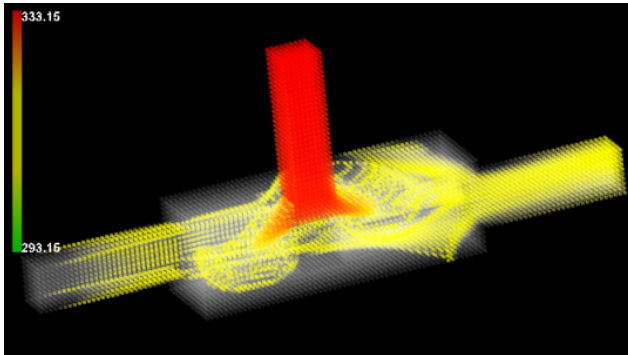


Figure 10. The T-junction: Warm liquid is entering from one inlet (right), hot liquid expands from a second inlet (above) and splits due to an obstacle.

successful analysis of the data. Visualization is often the means to understand complex relationships between different data items. However, the visualization has to cope with an amount of data which is usually vast due to detailed geometrical models (in terms of numbers of cells), the number of attributes computed for each cell and the time-dependent aspect (number of timesteps) of the simulation.

The case presented here is a T-junction with an extended chamber around the junction (see Fig. 10) and an obstacle below the secondary inlet (red in Fig. 10). Warm liquid starts to float in from the main inlet with the beginning

of the simulation (at the right of Fig. 10). After the first third of the time-span of the simulation, a hot liquid enters the junction from the second inlet. In this example the user is interested in the mixing behaviour and particularly in the existence of vortices and eddies, which are hard to detect by purely mathematical means. The dataset is considerably large (approximately 32000 cells, 18 data attributes, 100 timesteps), thus a thorough analysis represents a challenge for the software (on PC hardware) as well as for the user.

5.2. Initial Exploration

One can roughly distinguish between two phases during the investigation of the dataset: exploration and analysis [22]. The exploration is often unfocused and aims at becoming familiar with the specific characteristics of the dataset, studying approximate developments over time, identifying correlations between dimensions, and has the goal to formulate certain hypotheses. The subsequent analysis focuses on these hypotheses and confirms or refutes them. Both phases can be supported by interactively linked visualization tools (such as SimVis) in general and the combined scatterplots in particular.

The number of dimensions obtained from computational simulation (18 in the T-junction example) typically exceeds by far, what can simultaneously be displayed in a scatterplot. However, users usually consider a much smaller set of dimensions for most questions of analysis: Referred to as preferred dimensions, this set is normally sufficient to comprehend the most important features of the dataset (at least with respect to certain questions). In the case of the T-junction, these dimensions consist of velocity, pressure, turbulence kinetic energy (TKE), temperature and time, which is quite typical for computational flow simulations. Especially important in our context is that this set nicely matches what simultaneously can be displayed by our scatterplot (and preserving a certain visual coherence): Velocity, pressure, and TKE are mapped to the three axes of the 3D scatterplot, temperature is mapped to color and time is considered as temporal focus and context, defined with a range slider (see chapter 3.4).

In order to get a first approximate idea of the data, the exploration starts with moving a narrow temporal focus (time-slab) back and forth in time (temporal context and histograms for the spatial focus are enabled to permit a correct assessment of the relative position in time and the current point distribution, respectively, see Fig. 11). This basically reveals four temporal periods: Expansion of the liquid from the first inlet towards the outlet, a first consolidation phase, expansion and mixture from the liquid of the second inlet, and the second consolidation phase. The two expansion phases are characterized by approximately linear

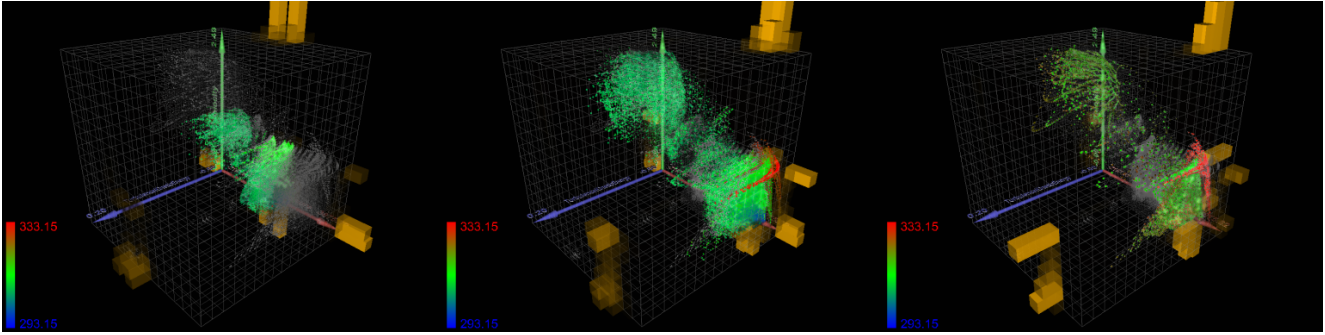


Figure 11. The evolution of velocity, pressure and turbulence over time: The first consolidation phase (left), the second expansion period (center) and the state at the end of the second consolidation phase (right).

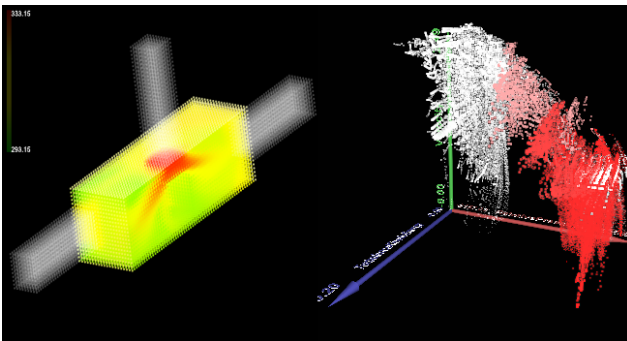


Figure 12. Mapping brushed physical space to attribute space: The main chamber of the T-junction (shown in the linked 3D View of SimVis) closely corresponds to the lobe at high pressure in the linked 3D scatterplot.

changes in velocity, quick rising in pressure and high values of TKE. During the consolidation phases, velocity and pressure remain approximately the same, while the TKE slowly decreases with increasing duration of the phases; the simulation shows stabilization at the end of the last phase. A particularly strong increase in the overall size of the point cloud with respect to the displayed dimensions can be observed during the second expansion phase, which suggests distinct vortices. This is identified as interesting for further examination – together with the beginning of the second consolidation phase in order to check, which turbulences are non-recurring and which are persistent.

5.3. Information Drill-Down and Analysis

Unlike the first phase of exploration as described above, where specifying the temporal focus has been the only interaction with the data, more advanced investigations require extended brushing facilities and linking of different views (see chapter 4.3). Relating physical locations to features in attribute space and vice versa is an essential part of the exploration with brushing. Three settings are reasonable in general for linked views:

- Brushing in space and visualizing the according attributes (Which characteristics can we find in certain parts of the geometric model?)
- Brushing attributes and visualizing the related positions in space (Where can we find certain characteristics?)
- Brushing attributes and visualizing other attributes (How are certain dimensions related to each other?)

We focus the further exploration on the main chamber (by brushing in space), since the situation within the inlets and outlet (being significantly influenced by the boundary conditions of the simulation) is rather known and therefore less interesting. Brushing in 3D space can easily be done with the combined scatterplot by mapping the X-, Y- and Z-coordinates of the cell centers to the three axes. As can be seen in Fig. 12, the main chamber (brushed in space and visualized with the linked 3D view of SimVis) largely coincides with the lobe at high pressure of the linked combined scatterplot showing pressure versus velocity versus TKE (Fig. 12 is an example for linking multiple views of different kinds). In order to make efficient use of the available space and resolution, we adapt the scaling of both combined scatterplots (space and attributes) so that it contains only the brushed part and consider the rest as spatial context.

Locally investigating the main chamber with a spatial brush which is refined to 5x5x5 cell centers (plus one surrounding slice of cells smoothly brushed) reveals high peaks of TKE in some parts next to the obstacle in the direction to the main inlet: Comparing spatial and attribute domain, Fig. 13 illustrates very different turbulence-conditions across the width of the T-junction over time. High turbulences often indicate the presence of eddies; therefore it could be that some of the hot water entering the main chamber from the secondary inlet forms an eddy in this place.

We try to verify or refute this assumption by brushing in attribute space. If an eddy exists, it most probably contains some of the entering hot liquid due to the proximity to

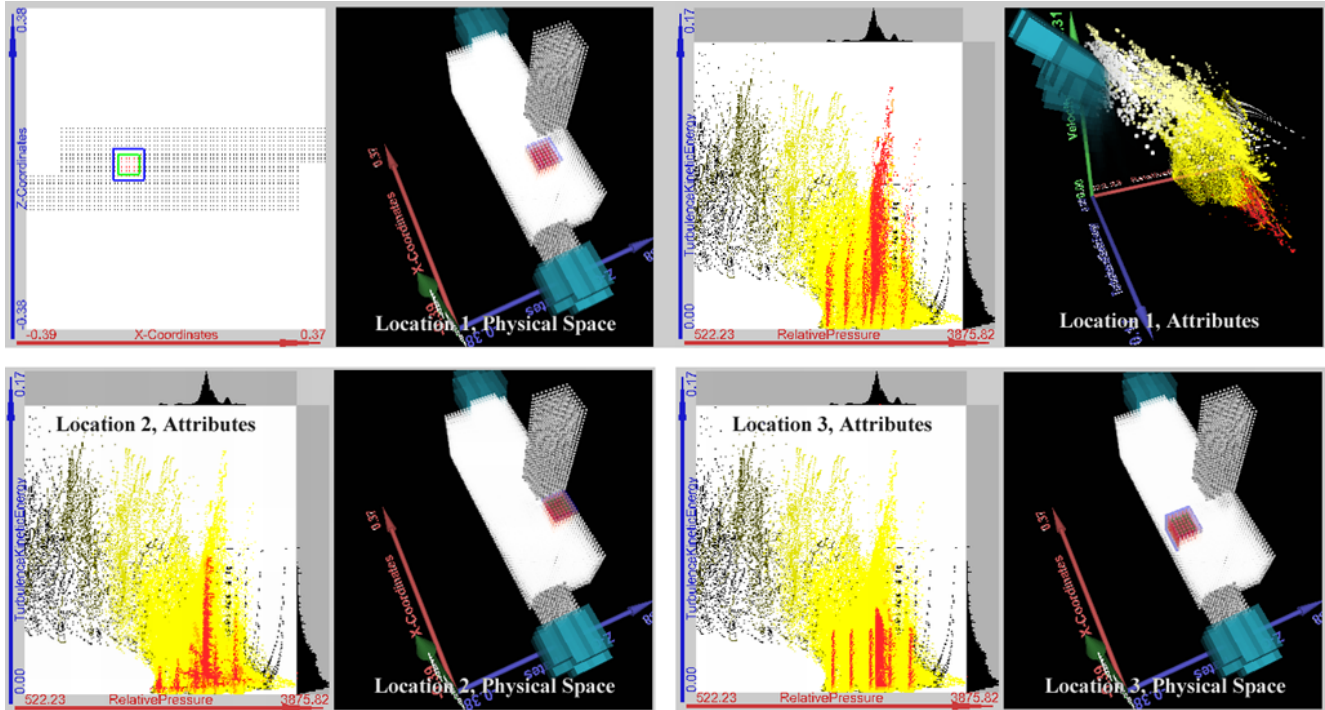


Figure 13. Relating locations inside the main chamber to their attributes: The turbulence kinetic energy is very high for location 1 (top), considerable for location 2 (bottom, left) and only medium for location 3 (bottom right). Note the representation of spatial context in the 3D views. Not shown by this figure: The smaller lobes in TKE originate from timesteps before the hot liquid has fully entered the main chamber and vanish afterwards.

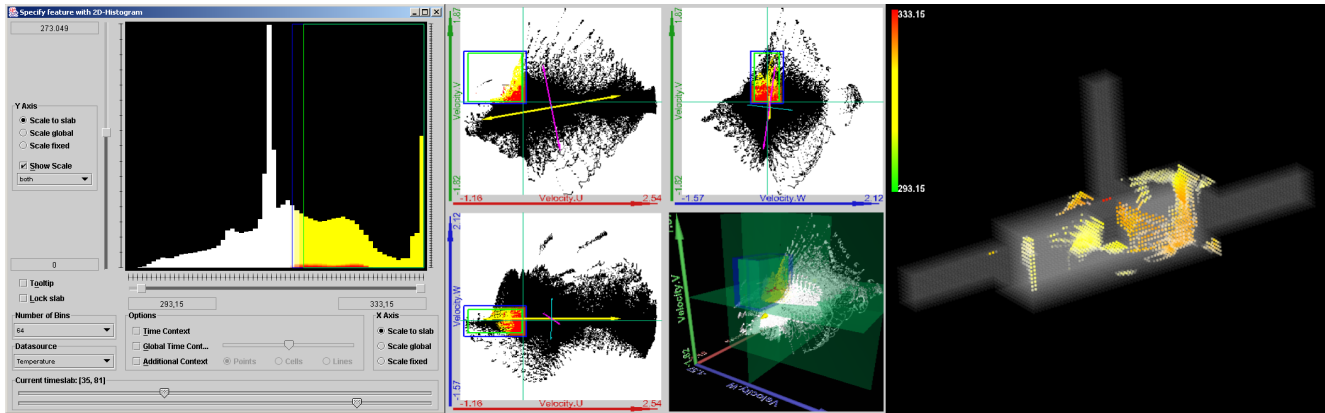


Figure 14. Combining several brushes to extract the eddy: Temperature is brushed using a histogram (left), the flow direction is defined with the combined scatterplot (middle) and the result is visualized in the 3D view (right).

the second inlet. We address this consideration by brushing high temperatures in the histogram view of SimVis [19], as illustrated in the left of Fig. 14. Furthermore, we are especially interested in areas where the flow direction is different from the main direction, because swirling liquids usually exhibit velocities from a wide range of flow directions. In order to address this, we refine the brush on temperature with a brush defined in a linked combined scatterplot where the X-, Y- and Z-components of velocity are mapped

to the three axes: The principle component analysis of these three dimensions reveals that the main flow can be found in positive X-direction with an additional flow in negative Y-direction – it is also helpful for this task to display the zero crossings of each dimension (as thin lines in 2D and transparent planes in 3D). Smoothly brushing all opposite flows by selecting approximately the negative X- and the positive Y-direction as well as a certain band around zero in the Z-direction (Fig. 14 middle) outlines a rotating flow, located

in the investigated area, in the linked 3D View of SimVis (Fig. 14 right). This is a clear indication of an eddy, which cools down during the upward flow, as can be seen when mapping temperature to color. However, the visualization shows that the defined properties can be found on the other side of the obstacle as well: This secondary result could be subject of further investigations.

Concluding, we have demonstrated in short, how interactively linking the combined scatterplot with other views of SimVis has been a successful way to explore and analyze complex, multi-dimensional and time-dependent flow data. Apart from linking and brushing, a task-centered application of the proposed extensions to 2D and 3D scatterplots has proven useful in gaining an in-depth understanding of the dataset.

6. Discussion and Future Work

In this paper, we have presented several extensions to 2D and 3D scatterplots in order to overcome some known restrictions (as a compromised 3D impression and limited interaction facilities) and have proposed a way of augmenting a 3D scatterplot with three 2D views in order to combine the advantages of both. The introduced extensions improve the depth perception in 3D, address the problem of overlapping point densities in both 2D and 3D, and help the user to keep an overview in space as well as in time when focussing on certain temporal and spatial parts of the plot. Linking 2D scatterplots, which are a widely used visualization technique, successfully complements the 3D scatterplot by providing both an alternative visualization and a convenient way for defining brushes which are consistent with the 3D setting. Since many questions occurring during the interactive exploration of datasets can only be answered by linking multiple views of various kinds (also different from scatterplots), we described the way, how the combined scatterplot is integrated in our SimVis system as an example for interactive, multiple-view exploratory software. Finally, we demonstrated the application of the proposed scatterplot-extensions as well as linking with external views by means of an exemplary investigation of large computational flow simulation results: Repeatedly interacting with the data (for example by moving around brushes) and visualizing related data proved a successful and quick way to detect interesting features and answer questions of the exploration.

Although much less common than 2D scatterplots, 3D scatterplots have proven advantageous along with the presented extensions: Most obviously, the additional dimension is valuable when simultaneously relating as many attributes as needed, and being able to do so in a single view emphasises visual coherence. Furthermore, its dimensionality matches the three dimensions of physical space, which permits intuitive visualization and interaction, like brush-

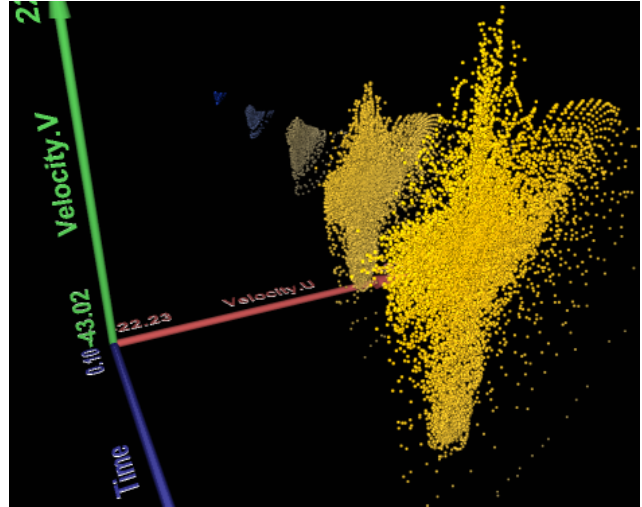


Figure 15. Assigning time to one axis of the 3D scatterplot allows to compare the 2D scatterplots over time.

ing certain areas of a physical model as well as the three components of velocity, or any kind of three-dimensional gradients. Third, special axes-mappings provide interesting views: For instance assigning time to one axis yields a set of 2D scatterplots as a stack of slices (Fig. 15). The major drawbacks of 3D scatterplots are occlusion, comprehension and interaction difficulties due to the mismatch in dimensionality of three shown by the scatterplot and two used by standard input and output devices, yet these can be mitigated, as shown in this paper.

As mentioned in the introduction (Section 1), we apply point-based rendering for displaying the 3D scatterplot – unlike much former work [2, 20, 21], which is based on volume rendering – allowing a continuous mapping from data space to scatterplot space. At the cost of a linear increase of the rendering complexity with the number of points, we avoid aliasing-problems caused by any kind of binning: Especially when displaying categorical dimensions, potential aliasing-artefacts may render the results of discretized visualization useless. Besides, the point-based approach allows assigning colors from channels or the DOI to points in an unambiguous fashion.

As future work, we plan to extend the integration of the principle component analysis as a means to introduce a data-driven coordinate system: Being able to define principle-component aligned brushes could ease the specification of features. Moreover, our future research will focus on brushing directly in the 3D view of the scatterplot.

Acknowledgements

This work was done in the scope of the basic research on visualization (<http://www.VRVis.at/vis/>) at the

VRVis Research Center in Vienna, Austria, which is funded by the Austrian research program Kplus. The example dataset is courtesy of AVL List GmbH, Graz, Austria.

References

- [1] M. Baldonado, A. Woodruff, and A. Kuchinsky. Guidelines for using multiple views in information visualization. In *Advanced Visual Interfaces*, pages 110–119, 2000.
- [2] B. Becker. Volume rendering for relational data. In *IEEE Symposium on Information Visualization (InfoVis '97)*, pages 87–91. IEEE, 1997.
- [3] A. Buja, J. McDonald, J. Michalak, and W. Stuetzle. Interactive data visualization using focusing and linking. In *IEEE Visualization '91*, pages 156–163, 1991.
- [4] E. H. Chi, J. A. Konstan, P. Barry, and J. Riedl. A spreadsheet approach to information visualization. In *ACM Symposium on User Interface Software and Technology*, pages 79–80, 1997.
- [5] W. Cleveland. *The Elements of Graphing Data*. Wadsworth Inc, 1985.
- [6] B. Cook, D. Asimov, and D. Hurley. Theory and computational methods for dynamic projections in high-dimensional data visualization, 1999.
- [7] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proc. of the 5th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*, pages 239–248, 2003.
- [8] H. Doleisch and H. Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. In *10th International Conference in Central Europe on Computer Graphics, Visualization, and Computer Vision (WSCG 2002)*, pages 147–155, 2002.
- [9] H. Doleisch, M. Mayer, M. Gasser, R. Wanker, and H. Hauser. Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In *Proc. of the 6th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2004)*, May 2004.
- [10] G. Furnas and A. Buja. Prosection views: Dimensional inference through sections and projections. *Journal of Computational and Graphical Statistics*, 3(4):323–385, 1994.
- [11] G. W. Furnas. Generalized fisheye views. In *Proc. of the ACM CHI '86 Conf. on Human Factors in Computing Systems*, pages 16–23, 1986.
- [12] A. Glassner. *Principles of Digital Image Synthesis*. Morgan Kaufmann Publishers, Inc, San Francisco, 1995.
- [13] B. Goldstein. *Sensation and Perception*. Wadsworth, Thomson Learning, 2002.
- [14] N. Greene. Environment mapping and other applications of world projections. *IEEE Computer Graphics and Applications*, 6(11):21–29, Nov. 1986.
- [15] D. Gresh, B. Rogowitz, R. Winslow, D. Scollan, and C. Yung. WEAVE: A system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *Proceedings Visualization 2000*, pages 489–492. IEEE, Oct. 2000.
- [16] D. Groth and E. Robertson. An integrated system for database visualization. In *Proc. of the 6th International Conference on Information Visualisation (IV'02)*, pages 462–467, 2002.
- [17] V. Interrante and C. Grosch. Strategies for effectively visualizing 3d flow with volume lic. In *Proceedings of the 8th conference on Visualization '97*, pages 421–ff. IEEE Computer Society Press, 1997.
- [18] I. Jolliffe. *Principle Component Analysis*. Springer-Verlag, New York, 1986.
- [19] R. Kosara, F. Bendix, and H. Hauser. Timehistograms for large, time-dependent data. In *Joint Eurographics – IEEE TCVG Symposium on Visualization*, 2004.
- [20] R. Kosara, G. Sahling, and H. Hauser. Linking scientific and information visualization with interactive 3D scatterplots. In *Proceedings of the 12th International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision (WSCG)*, pages 133–140, 2004.
- [21] G. Reina and T. Ertl. Volume visualization and visual queries for large high-dimensional datasets. In *Joint Eurographics – IEEE TCVG Symposium on Visualization*, 2004.
- [22] B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, page 336. IEEE Computer Society, 1996.
- [23] R. Spence and M. Apperley. Data base navigation: An office environment for the professional. *Behaviour and Information Technology*, 1(1):43–54, 1982.
- [24] D. Swayne, D. Cook, and A. Buja. Xgobi: Interactive dynamic data visualization in the X window system. *Journal of Computational and Graphical Statistics*, 7(1):113–130, 1998.
- [25] M. Theus, D. Cook, and H. Hofmann. Scatterplots for massive datasets. In *Proc. of the Interface 2003, 35th Symposium on the Interface*, 2003.
- [26] T. Toutin. Qualitative aspects of chromo-stereoscopy for depth-perception. *Photogrammetric Engineering & Remote Sensing*, 63(2):193–203, February 1997.