

TimeHistograms for Large, Time-Dependent Data

Robert Kosara, Fabian Bendix, Helwig Hauser

VRVis Research Center, Vienna, Austria
<http://www.VRVis.at/vis/>
{Kosara, Bendix, Hauser}@VRVis.at

Abstract

Histograms are a very useful tool for data analysis, because they show the distribution of values over a data dimension. Many data sets in engineering (like computational fluid dynamics, CFD), however, are time-dependent. While standard histograms can certainly show such data sets, they do not account for the special role time plays in physical processes and our perception of the world.

We present TimeHistograms, which are an extension to standard histograms that take time into account. In several 2D and 3D views, the data is presented in different ways that allow the user to understand different aspects of the temporal development of a dimension. A number of interaction techniques are also provided to make best use of the display, and to allow the user to brush in the histograms.

1. Introduction

While time is just another dimension in many data sets in terms of data organization, the way we perceive it and also its influence on physical phenomena is quite unique. Therefore, special methods are needed for time-dependent data.

Getting an overview of data – the major trends and the outliers – is very important in trying to understand the data. This is even more important for data that changes over time. One very popular method for getting such an overview are histograms. Standard histograms, however, provide little interaction and are also of little use for time-varying data.

The context of this work is computational fluid dynamics (CFD). Typical data sets in our experience are usually in the order of magnitude of about 100,000 cells for 100 time steps, or 1,000,000 cells for 10 time steps. Each cell has about 15 to 20 data attributes for each time step.

For dealing with such amounts of data, SimVis [2] was developed (see also Section 5). SimVis provides a number of different views, most of which use methods that are traditionally used in information visualization (InfoVis). Examples for such methods are scatterplots, parallel coordinates, and histograms. These views not only show the data, they also provide the user with the means to interact with the data and specify interesting features by means of brushing (i.e., selecting parts of the data and showing the selection in all views, thus making it possible to see connections).

1.1. Histograms and Time

A histogram is a bar chart, where the number of bars is usually selected by the user. Each bar corresponds to a *bin*, which is a value range on the selected axis. All values in the data set are compared to the bin boundaries, the values that fall into each bin are counted. The height of each bar then represents the number of values in each bin (Figure 1).

We distinguish two types of dimension in the discussion of time-dependent histograms: the data dimension and the display dimension (Figure 1). The traditional histogram is one-dimensional in its data (the bins are defined along one dimension) and drawn in two dimensions. It is of course also possible to draw a standard histogram using 3D graphics. But the more interesting case is that of two data dimensions, one of which is the time. Such a two-dimensional histogram (in terms of data) can be quite easily understood if drawn in 3D, because the user can imagine it being made up of a number of one-data-dimension histograms, one for each time step. Such a two-dimensional histogram can also be drawn in 2D, however, by projecting it into the plane along the count axis (Sections 4.2 and 4.3).

Time is handled in SimVis by means of time steps, which do not have to be equally spaced. Data for each point is only available for these time steps – therefore, they can be considered *event time* (meaning that the time axis is shaped by the occurring data, not in a continuous way). A time-related

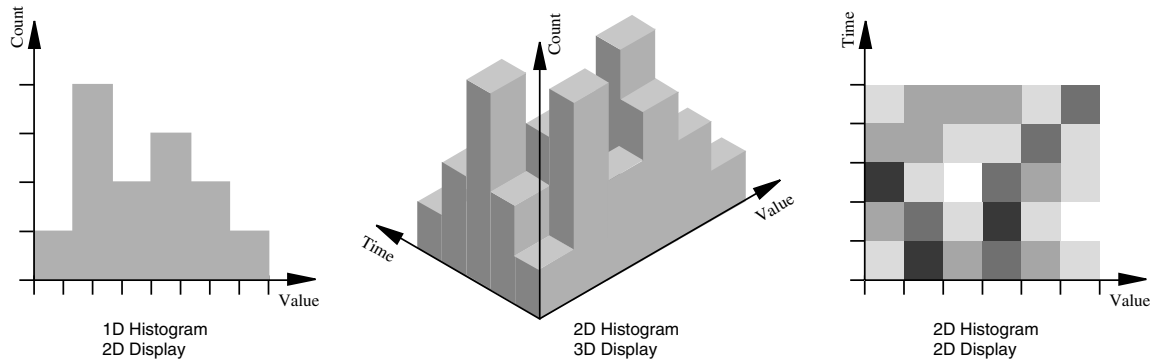


Figure 1: 2D and 3D visualizations of 1D and 2D histograms.

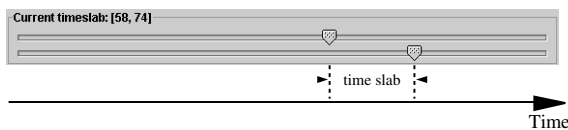


Figure 2: A time slab. The sliders define a time slab from time step 58 to 74 (both are included in the slab).

concept that plays a big role in SimVis is the definition of a time range, called a *time slab* (Figure 2). The current time slab can include one or more time steps, and is important because actions such as brushing apply only to these time steps. Time slabs can be defined as any interval in the total time range of the data set, there is no hierarchical time structure (e.g., combining seconds into minutes). Time slabs are always defined in terms of full time steps in the data set.

1.2. Motivation for Improvements, Goals

For the data described above, standard histograms were of little use. While it was possible to change the time step displayed by the standard histogram, the static image always showed only one time step. This proved to be too little information to be useful in practice.

At the same time, the standard histogram provided a good basis to start from because of its visual simplicity, but also because of its familiarity and ubiquity.

TimeHistograms are an extension of standard histograms which take time into account. Our goals in the design of TimeHistograms were the following:

- Give an overview over complex, time-varying data. Histograms are very useful for looking at all the data at once, but doing this in a way that is abstract enough not to overwhelm the user with details.
- Show temporal information in static images. While interaction is a very important part of TimeHistograms, the

user needs to be able to look at static images to get a better impression of what is going on. This is also important for documentation, where static images are needed.

- Retain the easy readability of the standard histogram. Make the additional information as easy to read as possible, and always provide the user with simpler information within the same image to fall back on.
- Support linking and brushing. Besides view-related interaction, the histogram also has to show brushed data, and also allow brushing in both 2D and 3D views.

1.3. Data Set

The data set used in the illustrations in this paper comes from the simulation of gas flow through a T-junction section joining two pipes. There is cold air flowing through the junction from one pipe, when a stream of hot air starts to enter from the other. The two air streams mix until an equilibrium state is reached. The data set consists of about 33,000 cells with 16 dimensions at 100 time steps.

2. Related Work

Histograms are used in both information and scientific visualization applications. This section gives a brief overview over a few uses of histograms that provide some interactivity (other than the selection of the axis). While there are certainly many more examples of histograms implemented in different applications, very little seems to get published about them – at least in the visualization and information visualization literature.

The influence explorer [10] allows the user to explore dependencies in situations where input as well as output values are multi-dimensional. It uses histograms for both the specification of input variables and the display of the results. The user can change the input value ranges on one set of histograms to see the results displayed on the other.

In a similar way, histograms can be brushed to answer

questions in a geographical data application [6], where the histograms act purely as a visualization of input data, which can be brushed to see the output on a map.

Parallel bargrams [11] are histograms with bars rotated by 90 degrees, and put end to end. This way, bargrams show the distribution of values on axes with few different values, and allow the user to select values by simply clicking on them.

A different kind of histogram are adjacency matrices that are used for the display of telecom call information [1]. Each cell in such a matrix is colored by the number of items in it, thus effectively creating a 2D histogram. Similar 2D histograms can be found in fields such as sound processing (time-frequency charts).

Histograms are not restricted to information visualization, however, but are also used for transfer function specification in volume rendering [4]. They show first and second derivatives of the data, and this way allow the user to directly select boundaries between materials – which usually are the most interesting features.

3. Early Approaches to TimeHistograms

Designing good histograms proved to be much more difficult than initially anticipated. This section describes two early attempts that did not lead to useful time-dependent histograms.

One of the developed techniques was to simply exchange the normal bars with History Bars [8], which is shown in Figure 3a. There is an outline of the current histogram and for each bin, multiple lines are drawn at various heights. These lines represent bars at different time steps and they are discriminated by their width, opacity and color.

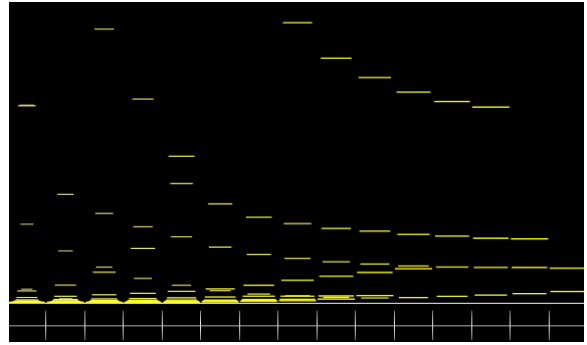
History Bars had good user feedback when used in process visualization, but proved to be problematic in our context. Due to the visual similarity and proximity of the bars, it is very easy to perceive bars from different bins and time steps as one unit (appearing as slightly curved lines in Figure 3a). However, this impression of trends is totally wrong.

Because of that there was the need for a shape that is more self-contained and that forms a better contrast to the rectangular histogram. The resulting mode which is described below in detail is called the Point Mode (Section 4.3.2).

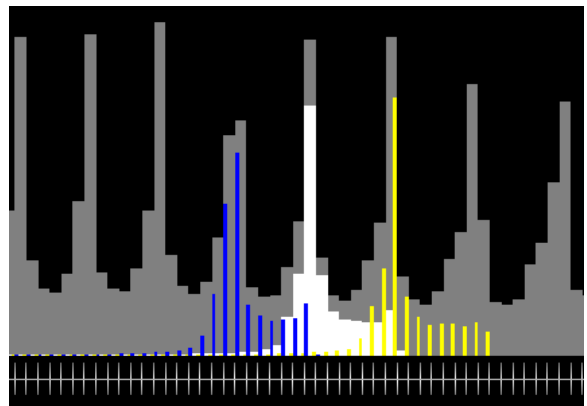
Another unsuccessful idea was to display the next and the previous time step in a normal histogram fashion, but as a smaller “sub-chart” (Figure 3b). Visually separating the two levels proved to be impossible, however, and when moving the time slider, it was impossible to understand the changes. The line mode (Section 4.3.3) is a better solution following a similar idea.

4. TimeHistograms

One of the goals in the development of TimeHistograms was to retain the simplicity and ease of understanding of the stan-



a) History Bars – different time steps per histogram bar are encoded as the length of the line. Proximity, however, is a stronger cue, so curved lines appear that do not represent real data.



b) Inner History – the sub-charts cannot be separated easily, therefore the bars blend into one confusing plot.

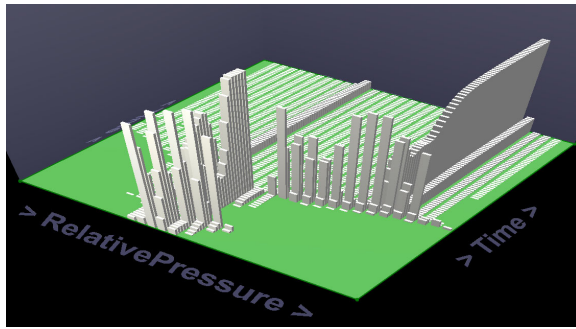
Figure 3: Early unsuccessful, abandoned approaches.

dard histogram as much as possible (Section 1.2). The traditional histogram is still visible in all the views, and is distinguished visually from the additional information. So the user can always look at the simple histogram of the currently selected time step and ignore all other information.

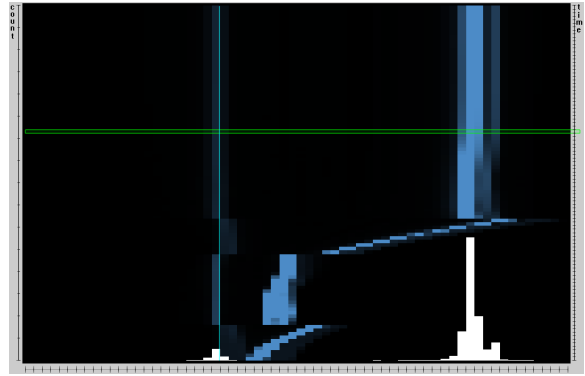
The discussion in this section is structured by the display dimensionality, not the dimensionality of the data (see the discussion in Section 1.1). The 3D histogram is described first, because it provides a good basis for the 2D histograms that are explained later. The 2D time context also is practically a projection of the 3D histogram (Figure 4).

4.1. TimeHistograms in 3D

While a 3D depiction is generally more complicated and requires more interaction, we have found it to be easier to un-



a) TimeHistogram in 3D, showing the development of pressure over time. One can see where the values change a lot and where a stable condition is reached.



b) 2D time context, showing the same data. This is a projection of the 3D histogram onto the background of the 2D histogram. The green frame marks the current time step on the time axis of the time context display (right).

Figure 4: TimeHistograms in 3D and 2D time context.

understand for time-dependent histograms, and also to provide a better overview than the 2D views.

The concept of the 3D TimeHistogram is quite simple: draw the histogram for each time step as a row of cuboids rather than rectangles, and arrange them one behind the other along the time dimension (Figures 1, 4, and 7).

The 3D display is made up of shaded cuboids that give a very good 3D impression. The user can very easily navigate around the histogram. Rotation is done in an object-centered way, and restricted to two axes: a vertical one for rotating the histogram to see it from different sides, and a horizontal one to change the height of the virtual camera over the ground plane. When the height is changed, the axis labels are rotated so that they face the user as much as possible. To see the direction of the axes, a simple indication is added to the labels that shows in which direction the values get larger (they follow the direction indicated by the ‘>’ signs).

It is also possible to zoom in and out (this is done using the mouse wheel), and to pan the display (by grabbing the histogram with the right mouse button and moving it around).

The described interaction is very powerful, yet easy to learn for the user. In addition, this view also implements the common interactions described in Section 4.4.

Another important interaction method is the control of the global height of the histogram bars. This control is simply a magnification of the height dimension, and can be changed in a rather large range. This is important in 2D as well as 3D displays, because there is no way for the program to tell which information the user is interested in. If the user is interested in the large bars, their height might lead to smaller bars not being visible anymore due to the dynamic range of the data.

4.2. Projected 3D Histograms as 2D Context

In addition to the standard histogram, a time context display can be put onto the background of the histogram (Figure 4b). This context overview is conceptually a projection of the 3D histogram along the height dimension: its value axis is parallel to the value axis of the histogram, and the time axis is parallel to the height axis of the 2D histogram.

The height values of the 3D histogram projection in the background are mapped to the opacity of rectangles representing the histogram bins. This mapping is linear and can be influenced by the user with a simple slider, which changes the slope of the mapping. This way, the user can select if he or she wants to see more details in the small or in the large values. Providing more parameters (e.g., which mapping, an offset) was considered but not done in order not to overwhelm the user with too many degrees of freedom.

The goal of the 2D context display is not to show exact values, but trends. It is possible to see if the values increase or decrease over time, or if they stay (approximately) constant. This view shows this information without the need for interaction and without any occlusion.

On the time axis (which is displayed on the right of the display, see Figure 4b), the current time step or time slab is marked with a green rectangle. This connects the time slider with the depiction and also allows the user to compare histogram bar heights with shades of blue, and thus makes it possible to tell which approximate heights bars have that are represented only in the overview (from other time steps). Additionally, the user can select the time step or slab to be displayed directly by clicking on this time axis. This provides a more direct means for navigating to interesting time steps than using the standard time control (Figure 2) would.

The 2D overview is not a mode of the display, but rather

additional information that can be combined with any of the other 2D modes. It is most useful with standard and point mode, however.

The data dimension shown in Figure 4b is the relative pressure, which shows an interesting pattern. After a relatively steady increase, it very quickly settles down into an almost constant pattern after about half of the total time. An interesting “overshooting” can also be seen right before the settling down begins.

4.3. TimeHistograms in 2D

Another way of showing changes over time in a standard histogram is the integration of the time axis into the normal 2D histogram. Here the screen space limits the possibilities of displaying the temporal data distribution.

All extensions presented in this section are based on the standard histogram display. The user can always look at the display and see the original histogram for the current time step. The remaining time steps are then visualized according to the used mode. Each mode augments the visualization in a different way and provides different kinds of information.

4.3.1. Standard Mode

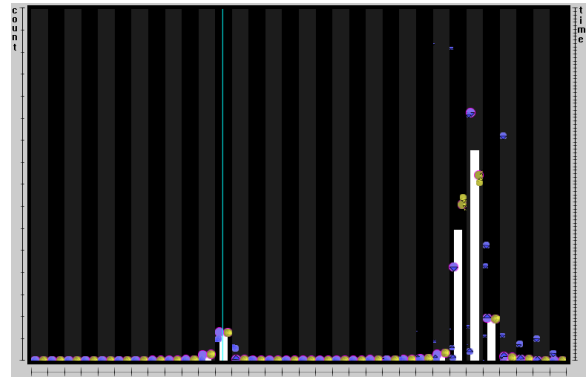
In addition to simply displaying the histogram for an axis, the user can move the currently displayed time step, and this way see the differences. It is also possible to switch on additional context, which is the sum of all histograms in the current time slab. This provides information about a range of time steps and is comparable to the arithmetic mean, because the height of the bars can be scaled arbitrarily and remains proportional (i.e., the display acts as if the sum was then divided by a number close to the number of time steps in the time slab).

Getting information about the temporal aspects of the histogram requires interaction in this mode. With a slider, the user can move the time slab or time step and this way traverse all time steps. If the differences between two successive displays are too big, this traditional histogram will not provide enough temporal context to make it possible for the user to envision the temporal data distribution – which is where the other modes come in.

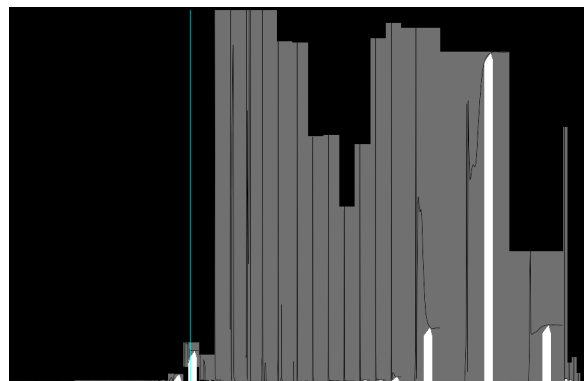
One very simple but quite important feature of the 2D view in TimeHistograms is that the zero on the value axis is marked with a turquoise line (e.g., Figure 4b and Figure 5c, right). This provides very important context to the engineer that is also much easier to read than the numbers giving the limits on the value scale.

4.3.2. Point Mode

In addition to the bars showing the histogram, this mode shows the immediate temporal neighborhood for each bar using small disks or “points” (Figure 5a). The points are



a) Point Mode. The colored disks/points represent the past and future of each bar (see below).



b) Line Mode. Every line chart shows the temporal development of the counts in one bin. The white bars show the current histogram, their tips point to the current values in the graphs.



c) Details of point and line mode. Left: The larger the point, the closer it is to the current time step. Blue points represent the past, yellow points the future. Right: The grey box contains the whole graph, therefore a vertical offset of the box tells the user that the count in this bin is never zero.

Figure 5: The different modes of 2D TimeHistograms.

drawn next to each bar, and represent past and future values for a small number of time steps. Points representing the past are colored blue and appear to the left of the bar, points showing future values are yellow and are drawn to the right of the bar (consistent with the time axis on the time slider). Their size also differs as a function of their distance from the current time step. The closest points are the largest, and they get smaller the farther back or ahead in time they are.

Because it is not easy to compare sizes of points that are far away from each other, small triangles are drawn into the points if they are farther than two diameters away from each other. They point to the next point, telling the user where to look for the next point, while retaining a smoother image when the points are close to each other (the sizes are then easy to compare and the sequence quite obvious). In addition, the points representing the time steps immediately preceding and following the current one, respectively, are also drawn with a purple outline, so that the user can find them quickly. What sounds complicated in the textual description is visually quite simple and relatively easy to follow. Usually, the size change (which is immediately visible) is also sufficient to see the temporal development – the additional information is only there for very complex cases.

The design for this mode built on the lessons learned from the failed attempt (Section 3) to make use of an idea from process visualization of real-time data [8], where values are displayed so that they leave a ‘trail’ that shows their history. The round points form a nice contrast to the rectangular bars, and are therefore easy to distinguish. It is also possible to ‘switch’ between looking at the bars or looking at the points easily.

In order to directly compare histograms from two different time steps, the user can activate a *tooltip* mode (Figure 6d). In this mode, if the user hits a point with the mouse, a polyline is drawn for the histogram of the time step that this point belongs to. This display is more effective to compare entire histograms, while the points are sufficient for telling the direction for a single bar or a small number of bars.

When using the F+C distortion (Section 4.4.1) to zoom into parts of the histogram, bars are drawn for the immediate temporal neighborhood, and points are then drawn for some more time steps ahead and back (Figure 6c). This adds to the information present in the display, without using much more space. At the same time, the points and bars’ heights are visually so similar, that there is almost no discontinuity between them, and it is very easy to tell the shape of the bars that are scrolled in when moving the time slider from the points.

The same dimension (relative pressure) is used in Figure 5a as in the description of the 2D overview (Section 4.2). The little overshooting can also be told in this mode, even though it is not that obvious. The histogram for the current time step can be read much more precisely, though.

4.3.3. Line Mode

A different 2D representation of the time-dependent histogram uses a more familiar metaphor. A simple line graph is drawn inside each box, showing the temporal development for that bin (Figure 5b). Inside the box, the time axis is parallel to the value axis of the histogram (this is similar to the idea of dimensional stacking [5]). A small ‘pointer’ is drawn on top of each bar of the histogram which points to the current position on the graph.

The whole line graph for each bar is contained in a gray box. This box shows the user the minimum and maximum values for each of the bars. If the value count in the bin is never zero, the user will notice this because the box will not touch the base line in this case.

Since there is usually not enough screen space to properly see the time line, Focus+Context distortion (Section 4.4) makes it possible to see a small set of bars in detail.

But even without interaction, this mode provides two levels of information. When looking at the global structure (white on dark), one sees the histogram for the current time step. When looking at the details (black on grey), one can read the development of each bar over time.

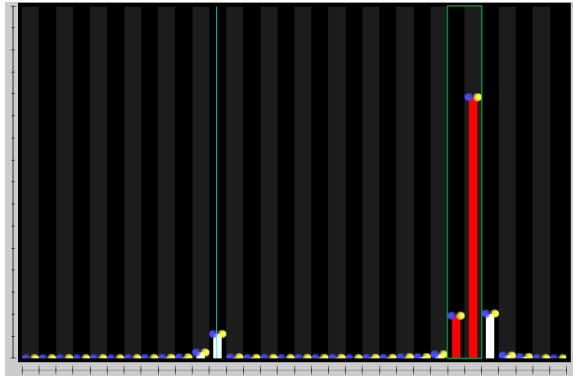
4.4. Common Interactions

Independently of the view, there are a few interactions that are always possible. They are implemented in slightly different ways, but are conceptually the same.

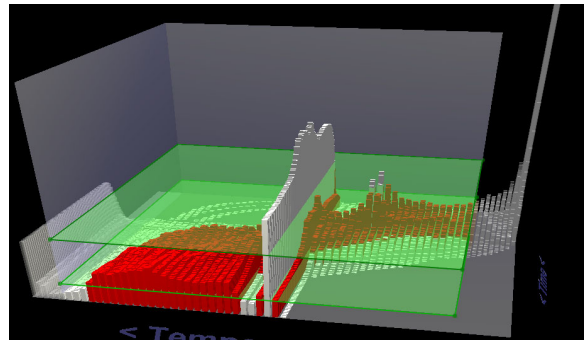
Perhaps the most important one is the zooming into the data. The user can specify ranges, in both the value and the time dimensions (the time slab). In the 2D views, only the data within these regions are displayed. Time is treated a bit differently here, because the user can not only change the upper and lower limit of the time slab to be displayed, but can also move the complete slab. This has proved to be useful in the other SimVis views, and seems to be a rather natural way of working with time for complex data.

4.4.1. Focus+Context (F+C)

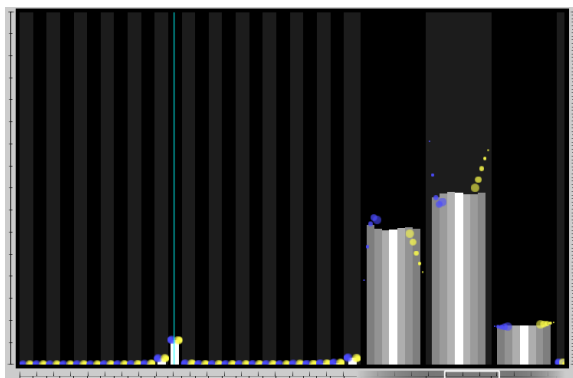
In the 2D views, a classical focus+context method is implemented that is similar to the perspective wall [7]. The user selects a region that can then be enlarged, while the rest of the display is compressed (Figure 6c). This distortion is not binary like the TableLens [9], but changes from the center of interest to the outer parts. The same distortion is applied to a whole bin, however. The distortion is not only visible in the bars and the background, but there are also two scales at the bottom of the display: one shows the undistorted scale, and one the distorted one. This provides the user not only with the possibility to get more space for the display, but also different information can be displayed in the additional space (e.g., in point mode, bars are shown for the immediate temporal neighborhood). The enlarged region can also be moved, to explore the data with the current settings.



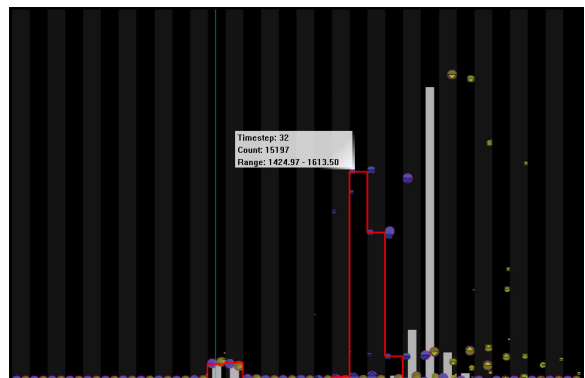
a) Brushing in 2D – the user has selected the values between the two vertical green lines. Brushed bins are now drawn in red. The turquoise line shows where zero is on the value axis.



b) Brushing in 3D – all bins whose counts lie between the two horizontal planes are brushed (displayed in red).



c) Table lens-like Focus+Context (F+C) – parts of the display can be zoomed in to show more information, like more time-steps as a bar chart.



d) Tooltip mode – a tooltip is shown that gives the user information about the bin (current time step, data point count, value range) as well as a second histogram (in red) for the point the mouse is currently pointing at. Differences (right) as well as similarities (left) are visible.

Figure 6: Interaction in TimeHistograms.

In 3D, the F+C display is different. The bars representing the values are not only pushed out of the cube representing the current value range, but their footprint is also made smaller, and they are drawn translucently. This provides a very clear metaphor for the F+C display (the user can see what happens while moving the sliders), and also makes it possible to see through large occluding structures (Figure 7). Such large structures are often found at the border of the value range, because they represent values from start or boundary conditions.

4.4.2. Brushing

Brushing in the histogram is done in terms of bins, both in the 2D and 3D views. When the user makes a selection in 2D (Figure 6a), all bins that are touched by the selection are completely selected (and the selection is expanded). Because the user does not know the distribution of values inside the

bin, partially brushing bins does not appear to be sensible. In the temporal dimension, only the points in the current time slab are brushed.

In 3D, brushing works in a different dimension than in 2D, namely in terms of the value count in the bins. The user can drag two horizontal planes inside the box of the 3D histogram in the height direction, so that the planes intersect with some or all of the cuboids. Each of the planes represents a certain count value. All data values are selected that belong to bins which have a height in the range defined by the two selection planes (Figure 6b). This selection only applies to the bins in the focus, i.e., those that have not been pushed out into the context region.

Arguably, brushing by bin count rather than by bin could work in a very similar way in 2D, but has not yet been implemented. This is planned for the future, though.

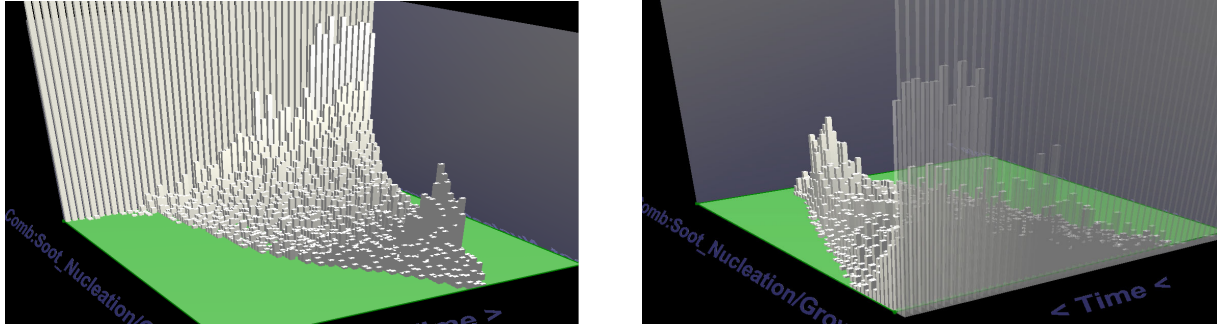


Figure 7: Getting rid of large occluders near the borders. The occluder at the far end of the left image would block the view from the other side. Pushing a few rows of the histogram into the context allows the user to see the data from another perspective, and even to get an idea of the context data.

4.4.3. Scaling

The program also provides the possibility to automatically scale the value and the height axis. The user can request each of the axes separately to be either scaled to the current time slab or the complete data set. Scaling the value range such that empty bins at the lower and upper end of the scale are left out can cause problems with understanding change. When the time slab is moved and the data mostly moves along the value axis, the user must keep track of the numeric values that give the locations of the boundaries. But for quickly zooming into the currently relevant values, this mode is very useful.

Scaling in the height dimension, as mundane as it may sound, is a very important interaction in TimeHistograms. Automatic scaling of this axis is very difficult, because of the extreme dynamic range of the values (one bar might represent 200,000 data items, while another might only have a count of 50). Therefore, the user must be able to change the scaling to change the focus to large or small bars. In the 2D view, if a bar gets taller than can be displayed in the window, a small arrow is drawn onto it, showing the user that the bar actually extends beyond that border. In 3D, this is not necessary, because the user better understands that he/she is only looking through a “window” at the data, and it is also easier to see the full extent of the bar (by changing the viewpoint).

4.4.4. Number of Bins, Tooltip

Another very simple interaction is the definition of the number of bins. At the moment, the user can select between 16, 32, 64, 128, 256, 512, and 1024 bins. The reason for using powers of two is an implementation detail (calculations for display distortions are easier this way), but it would be very easy to change this to allow arbitrary numbers. From our experience, the numbers 32 and 64 are the most common choices, and there seems to be no need for an intermediate number of bins. Choosing the right bin size is usually considered a very important question for histograms, but for this

application domain, we have found it to not be very critical. The data is relatively smoothly distributed, and the values are never precisely the same, so they tend to “leak” to neighbouring bins. The histograms at different resolutions therefore look very similar and hold few surprises for the user.

The tooltip, while available in all modes, is especially useful in point context mode. For each bar, the tooltip shows the time step, the number of values in the bin, and the precise bin boundaries. The time step is interesting because in point context mode, the tooltip can also display an alternative histogram in addition to the current one (Section 4.3.2).

4.4.5. Using A Different Second Dimension

Instead of time, a different dimension can be selected as the second dimension (Figure 8). The resulting histogram shows the distribution of values into bins defined by the two dimensions. There is a very simple duality between this display and a scatterplot: Seen from above, the histogram looks like a scatterplot (if a sufficient number of bins is selected). This is similar to the 2D overview (Section 4.2). But the scatterplot does not show how many values were plotted onto the same pixel – this information can only be seen in the histogram.

5. Integration in SimVis

SimVis (described briefly in the introduction) is a framework for interactive specification of features in simulation data. The definition of such features is modeled as a tree of brushes, which are selections of data. These selections can be combined using logical operations to define more complex features (e.g., regions where the flow is slow, the temperature is high, and where the rate of change was small in a certain time period).

The framework makes use of multiple, linked views from information as well as scientific visualization. The result of the feature specification is linked to all views and can be seen in various views. The TimeHistogram is one of these views.

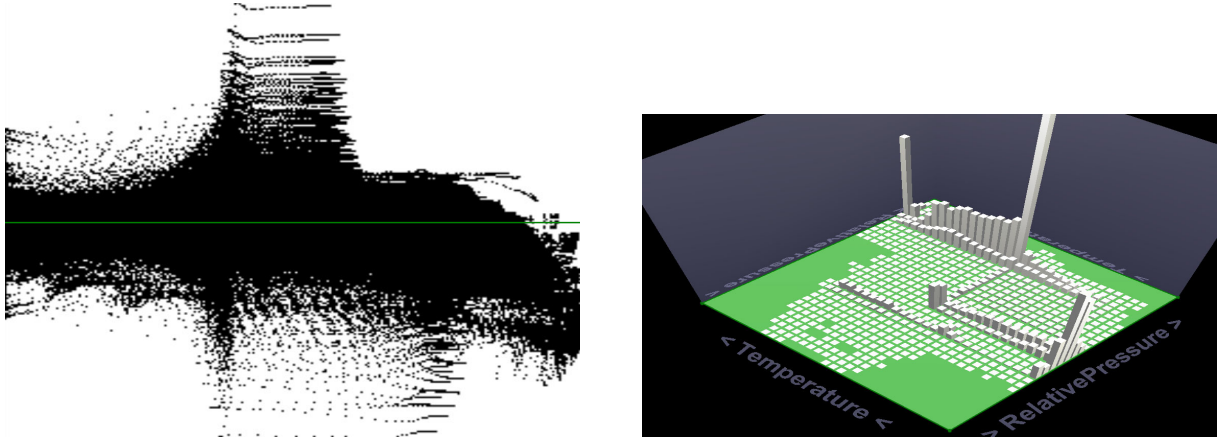


Figure 8: Using the TimeHistogram to plot two dimensions against each other – both plots show the same dimensions. In contrast to the scatterplot, the histogram shows the extreme clustering of a lot of values around very distinct structures.

TimeHistograms can both show brushed data and allow the user to brush directly in the histogram. In SimVis, two different levels of brushing are distinguished: brushes that were done in the current view, and the globally defined brush (which is the combination of all the brushes, so the points in this brush might not contain any of the points brushed in a particular view). The color of the bars changes gradually from white to red (or yellow) the more brushed data points they contain.

All modes of TimeHistograms run at interactive frame rates for the data sets described in the introduction.

6. Discussion and Experiences

TimeHistograms have not been the subject of a formal user study yet, but they have been presented to CFD experts who are working with the SimVis application. The results of this rather informal evaluation are described here, along with a few examples of how interaction works in practice with TimeHistograms.

We found to our surprise that the 3D display is much more intuitive and accessible than the 2D modes. Users could relate to it better and also understand the meaning of the graphical display more easily. The occlusion problems that we normally associate (occlusion, difficult navigation, etc.) were no real issue.

At this point, we also found a problem in the initial point mode design, which did not have the marks for the first time step and the small arrows pointing to the next or previous time step. We found that we needed this additional information so that users would not overlook points or not be able to tell which came first (the size difference can be hard to see when the points are far apart).

TimeHistograms were also used in a recent case study on

a Diesel exhaust system [3], where they mainly served to tell when the chemical processes (burning of the soot) had settled down. To do this, one needs to look at the histogram and to tell when the changes between time steps become so small that they can be ignored. The TimeHistogram proved to be very valuable for this purpose.

When implementing a 2D and a 3D view of essentially the same data, the question of which performs better comes to mind. While we found that 3D provided a much better introduction to time-dependent histograms for the novice users, we also found some problems with it. One is certainly occlusion, which is a problem inherent in 3D. It is much easier to miss a feature in the 3D histogram than when using the 2D overview, for example (Figure 9). But the 2D modes also bear the possibility of "temporal occlusion" t , due to the limited amount of time steps they can display at once.

Concluding, the combination of both 2D and 3D views on the same data offers a lot of ways to gain insight into the data by depicting its different aspects.

7. Conclusions

TimeHistograms are an extension of the well-known histograms for time-dependent data. They are a useful addition that provides additional insight, particularly for large and complex data sets. The presented techniques provide different views on the data in 2D and 3D, which require interaction, but also provide a lot of additional information.

While the 3D view was originally anticipated to be problematic due to the usual perception problems, we found it to be very useful and easy to understand for our audience, and even to serve as an introduction to the more demanding 2D views.

TimeHistograms were demonstrated with scientific data

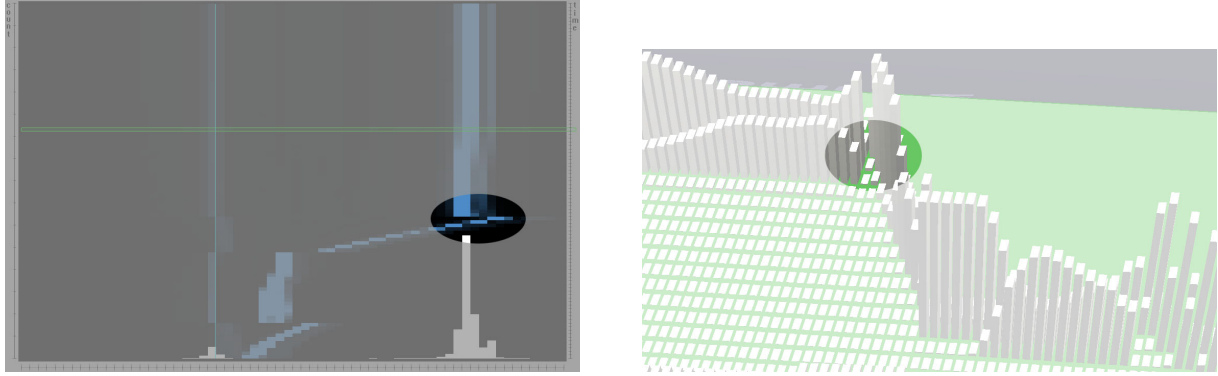


Figure 9: Working around occlusion using interaction: Seeing the small interesting detail marked in the left image requires quite some interaction to be clearly visible in the 3D view on the right. In such a case, the 2D view reveals details at one glance that could have easily been missed in 3D.

in this paper, but they are certainly also useful for time-varying data that is not bound to a physical object.

Future work includes more in-depth evaluation of the technique with users, as well as more work on the 3D view. Brushing in that view should be made more flexible, and also the separation of different time steps and labeling of histograms in 3D requires some more work. More flexible brushing in both 2D and 3D modes will also be another interesting topic to work on.

Acknowledgements

This work has been done in the scope of the basic research on visualization at the VRVis Research Center in Vienna, Austria (<http://www.vrvis.at/>), which is funded by the Austrian research program Kplus. The data set used in the images is courtesy of AVL List GmbH, Graz, Austria. We would like to thank the reviewers for their very insightful and detailed remarks.

References

- [1] James Abello and Jeffrey Korn. MGV: A system for visualizing massive multidigraphs. *Transactions on Visualization and Computer Graphics*, 8(1):21–38, 2002.
- [2] Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of the 5th Joint IEEE TCVG - EUROGRAPHICS Symposium on Visualization (VisSym 2003)*. ACM Press, 2003.
- [3] Helmut Doleisch, Michael Mayer, Martin Gasser, Roland Wanker, and Helwig Hauser. Case study: Visual analysis of complex, time-dependent simulation results of a diesel exhaust system. In *Proceedings of the Joint Eurographics – IEEE TCVG Symposium on Visualization (VisSym 2004)*, 2004.
- [4] Joe Kniss, Gordon Kindlmann, and Charles Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings IEEE Visualization 2001 (Vis'01)*, pages 255–262. IEEE Computer Society Press, 2001.
- [5] J. LeBlanc, M. O. Ward, and N. Wittels. Exploring N-dimensional databases. In *Proceedings of IEEE Visualization 1990 (Vis'90)*, pages 230–239. IEEE Computer Society, 1990.
- [6] Qing Li and Chris North. Empirical comparison of dynamic query sliders and brushing histograms. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'03)*, pages 147–154. IEEE Computer Society Press, 2003.
- [7] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of ACM CHI '91 Conference on Human Factors in Computing Systems*, pages 173–179, 1991.
- [8] Krešimir Matković, Helwig Hauser, Reinhard Sainitzer, and Eduard Gröller. Process visualization with levels of detail. In *IEEE Symposium on Information Visualization 2002 (InfoVis 2002)*, pages 67–70. IEEE Computer Society Press, 2002.
- [9] Ramana Rao and Stuart K. Card. The table lens: Merging graphical and symbolic representations in an interactive focus+context visualization for tabular information. In *Proceedings of ACM CHI '94 Conference on Human Factors in Computing Systems*, pages 318–322, 1994. Color plates on pages 481–482.
- [10] Lisa Tweedie, Robert Spence, Huw Dawkes, and Hua Su. Externalising abstract mathematical models. In *Proceedings on Human Factors in Computing Systems*, page 406ff. ACM Press, 1996.
- [11] Kent Wittenburg, Tom Lanning, Michael Heinrichs, and Michael Stanton. Parallel bargrams for consumer-based information exploration and choice. In *Proceedings of the 14th annual ACM symposium on User interface software and technology*, pages 51–60. ACM Press, 2001.