Generalizing Focus+Context Visualization

Helwig Hauser

VRVis Research Center in Vienna, Austria, `http://www.VRVis.at/`,
`Hauser@VRVis.at`

Focus+context visualization is well-known from information visualization: certain data subsets of special interest are shown in more detail (locally enlarged) whereas the rest of the data is provided as context (in reduced space) to support user orientation and navigation.

The key point of this work is a generalized definition of focus+context visualization which extends its applicability also to scientific visualization. We show how different graphics resources such as space, opacity, color, etc., can be used to visually discriminate between data subsets in focus and their respective context. To furthermore demonstrate its general use, we discuss several quite different examples of focus+context visualization with respect to our generalized definition. Finally, we also discuss the very important interaction aspect of focus+context visualization.

1 Introduction

For a long time already, modern society is greatly influenced by computers. Mainly, computers are used to process data of various kind. Additionally, computers are also used to support the acquisition of data, for example, through measurements or computational simulation. Due to a steadily increasing performance of computers (Moore's law), year by year more data is processed. Since users do not extend their capabilities in data-processing at a comparable rate, there is an increasing need for efficient tools to support the processing of large amounts of data.

One very useful opportunity for accessing large amounts of data is visualization. Data is communicated to the user in a visual form to ease processing. Instead of dealing with loads of numbers, the user accesses the data through pictures and a graphical user interface. This approach is especially useful when the data has at least some spatial form inherently associated with it. In many scientific applications, for example, data is tightly related to concrete parts of our real world, e.g., a 3D computer tomography scan of a human body in a medical application or the 3D simulation of air flow around the computer model of a new aircraft.

The main advantage of visualization is that it uses the great bandwidth of the human visual system for visualization. However, also for visualization the amount of data to be shown at once is limited. For very large data sets, details cannot be shown for all of the data at the same time. In this case, the user usually is offered the opportunity to either get an overview of the data (no details), or zoom into specific parts of the data and get all of the details there.

While scientific visualization (SciVis, the visualization of scientific data) has been researched for dozens of years already, more recently also the visualization of non-scientific, abstract data (InfoVis, information visualization) such as bank account data or census data has become popular. In InfoVis, an additional step is required in the visualization process, i.e., the mapping of non-spatial data to a visual form. As the user has to learn this additional mapping to effectively use the visualization (and to successfully build up a mental map of the data–form relation), more care is required to support the user with orientation in the visualization. Careless zooming across multiple levels of details can easily cause an effect like being lost in too many details. Thus, advanced solutions have been developed in this field to supply users with both overview and details of the data at the same time.

2 Focus+Context Visualization

In information visualization, an approach called focus+context visualization (F+C visualization) has been developed which realizes the combination of both a general overview as well as a detailed depiction within one view of the data at the same point in time. Traditionally, focus+context visualization refers to an uneven distortion of visualization space such that relatively more space is provided for a certain subset of the data (data in focus). At the same time the rest of the visualization is compressed to still show the rest of the data as a context for improved user orientation.

The idea of using different magnification factors for different parts of the visualization (in one image) to display information in a F+C style already dates back to the '70s of the 20th century [9, 22]. Furnas' work on the fisheye view [10] in 1981 often is accepted as the historical start into computer-based F+C visualization. In this work, Furnas describes how information is selected for display depending on an a-priori importance and the distance of each data item to the current focus of the visualization. Also in the early 1980s, Spence and Apperley presented the bifocal display as a one-dimensional distortion technique [43] to provide a shrinked context on both the left and the right side of an undistorted focal region in the middle of the visualization.

During the 1980s, both approaches have been generalized and extended [11, 32]. In 1992, Sarkar and Brown presented the graphical fisheye view [41], based on Furnas' work, but more focused on the graphical appearance of the F+C visualization (comparable to a real fisheye lens). One year later, Sarkar et al.

discussed two techniques (orthogonal and polygonal stretching) for F+C visualization based on the concept of a stretchable rubber sheet [42]. In 1994, Leung and Apperley already presented a review of distortion-oriented F+C visualization, including additional approaches such as the perspective wall [36] (see also later), and providing a respective taxonomy [33]. They describe techniques of F+C visualization by the characteristics of the magnification function (being the derivative of the tranformation function from the undistorted view to the F+C view). Doing so, three classes of techniques are differentiated: (1) approaches with a continuous magnification function (such as the graphical fisheye [41]), (2) techniques with piece-wise constant magnification factors (the bifocal display [43], for example), and (3) others (the perspective wall [36], for example).

The perspective wall, presented by Mackinlay et al. in 1991 [36], is based on the concept of "bending backwards" parts of the display on both the left and the right side of the focus region in the center of the screen (similar to the bifocal display [43]). Perspective projection is used to achieve a variation in magnification factors within this kind of F+C visualization. In 1993, this approach was extended to the so-called document lens [40] – also parts above and below the focal region are used for context visualization.

In the domain of distortion techniques with continuous magnification functions, further extensions have been presented after the first half of the 1990s. In 1995, the three-dimensional pliable surface was presented by Carpendale et al. [3], also using perspective projection to achieve different magnification factors in different parts of the display. Gaussian profiles are used to generate magnification (and thus yield to a continuous magnification) and multiple foci are possbile in one view. In 1996, Keahey and Robertson presented non-linear magnification fields as a technique independent from perspective projection and with direct control over the magnification function on every point of a grid over the display [23, 24]. The transformation function is computed in an iterative process, locally optimizing on the difference between the discrete derivative of the transformation field and the input (magnification field).

In 1995, the mapping of hyperbolic space to the plane was used by Lamping et al. [29–31] to achieve F+C visualization, enabling the visualization of infinite space on a limited screen space (at least in principle). In a similar fashion, Kreuseler et al. used the mapping from spherical space to the plane for F+C visualization [28], allowing to move the focal center around the sphere.

The large amount of work on distortion techniques for F+C visualization documents the relevance of this approach, especially in the domain of information visualization. But instead of discussing more details about distortion techniques for F+C visualization or other approaches in this field, we restrict this overview to the above mentioned examples and proceed towards our generalization of F+C visualization. First, however, we briefly discuss how focus is separated from context, an inherently necessary part of every focus+context visualization.

3 Separating Focus from Context

When dealing with focus+context visualization, it is inherently necessary to have a notion of which parts of the data are (at least at one point in time) considered to be "in focus" and what others are not (context part of the data). In the course of this work, we use a so-called *degree-of-interest function* (DOI function), *doi*(), which describes for every item of the data whether (or not) it belongs to the focus (similar to Furnas' definition [10], but normalized to the unit interval $[0, 1]$):

$$doi_{\mathrm{bin}}(\mathtt{data}[i]) = \begin{cases} 1 & \text{if } \mathtt{data}[i] \text{ is part of the focus} \\ 0 & \text{if } \mathtt{data}[i] \text{ is part of the context} \end{cases}$$

In many application scenarios, a binary discrimination between focus and context (as formulated above) is appropriate, i.e., to assume a sharp boundary between the data items in focus and all the others. In many other cases, however, it is more appropriate to allow a smooth change of *doi*()-values between the data items in focus and their context (resulting in a *smooth degree of interest* [6]). In other words, the question of whether a data item belongs to the focus (or not) also can be answered by the use of a fuzzy attribute *doi*():

$$doi(\mathtt{data}[i]) = \begin{cases} 1 & \text{if } \mathtt{data}[i] \text{ is part of the focus} \\ doi \in \,]0,1[& \begin{array}{l}\text{if } \mathtt{data}[i] \text{ is part of the smooth} \\ \text{boundary between focus and context}\end{array} \\ 0 & \text{if } \mathtt{data}[i] \text{ is part of the context} \end{cases} \tag{1}$$

Accordingly, a fractional value of *doi* is interpreted as a percentage of being in focus (or interest). A fractional *doi*()-value can be the result of a multi-valued definition with multiple (still discrete) levels of interest, e.g., $doi \in \{0, 25\%, 50\%, 75\%, 1\}$, a non-sharp definition of what is interesting (e.g., through a definition which is based on continuous spatial distances), or a probabilistic definition (e.g., through a definition incorporating a certain amount of uncertainty).

Usually, the specification of the *doi*()-function is tightly coupled with the user interface. Different approaches are used to let the user specify which parts of the data (at one point in time) are of special interest (explicit vs. implicit specification, for example). In section 5 we discuss the interaction aspect of F+C visualization in more detail.

In traditional F+C visualization (space-distortion techniques), the degree of interest $doi(\mathtt{data}[i])$ is directly related to the local magnification used to depict a data item $\mathtt{data}[i]$ (this 1:1–relation only holds to a certain extent of accuracy – in general it is not possible to translate every DOI/magnification function into a corresponding transformation function [23, 33]): the larger the *doi*()-value is, the more screen space is used for visualization, at least locally.

4 Generalized Focus–Context Discrimination

Although the vast majority of research work on F+C visualization has been devoted to space-distortion techniques, the idea of visually discriminating the parts of the data in focus from all the rest, i.e., the context, is more general. In addition to using more space for showing the focus in more detail, other visual dimensions can be used in a similar way. In volume rendering, for example, usually more opacity is used for parts of the data in focus [34], whereas a greater amount of transparency is used for context visualization. Additionally, also color can be effectively used to visually discriminate different parts of the data. In a system called WEAVE [12], for example, those parts of the data which positively respond to a certain user query (i.e., the current focus) are shown in color, whereas the rest of the data (the context) is shown in gray-scale.

Similarly, other visual dimensions, such as image frequencies, rendering style, etc., can be used to achieve focus–context discrimination (see below for examples). We therefore propose to generalized the definition of focus+context visualization in the following way: *focus+context visualization is the uneven use of graphics resources* (space, opacity, color, etc.) *for visualization* with the purpose to visually discriminate data-parts in focus from their context, i.e., the rest of the data. In table 1, we give several examples of F+C visualization which are quite different from each other but which all match the above definition and thereby demonstrate its general character. The examples differ from each other with respect to which graphics resource is (unevenly) used to achieve F+C visualization. Below we discuss some of these examples in more detail (those marked with an asterix in table 1). In table 2 we provide a side-by-side comparison of five sample techniques (one sample image each) with pointers to other parts of this document with more detail as well as also to other pieces of related literature.

4.1 More Opacity for Visualization in Focus

One alternative style of F+C visualization (alternative to space-distortion techniques) is identified in a domain where usually other objectives, slightly different from focus–context discrimination, actually govern the development of new techniques. In volume rendering, all from the beginning on [34], a so-called opacity transfer function (OTF) $\alpha()$ is used to deal with the fact that usually not all of the 3D data can be shown simultaneously at full intensity – OTF $\alpha()$ is used to map the data domain to the unit interval ($1 \leftrightarrow$ opaque, $0 \leftrightarrow$ completely transparent). Using an OTF, different values of opacity/transparency are assigned to different parts of the data. This causes that some parts of the data become more prominently visible in the rendered image while others are not (or only hardly) visible.

Originally, the use of an opacity transfer function was not argued with the need to discriminate parts of the data “in focus” from their “context”.

graphics resource	**approaches**	**sample technique(s)**
space	more space (magnification) for data in focus	graphical fisheye view [41], ...
		F+C process visualization [37]*
opacity	focus rather opaque, context rather transparent	direct volume rendering [34], ...
		RTVR [38]*
color	colored focus in gray context	WEAVE [12], SimVis [5,6,8]*
	focus: saturated/light colors	Geospace [35], RTVR [38]*
frequency	sharp focus, blurred context	semantic depth of field [25,26]*
style	context in reduced style (non-photorealistic rendering)	two-level volume rendering [17,18]*
		NPR-contours [4]*

*... techniques which are described in more detail in section 4

Table 1. Realizing (generalized) F+C visualization by the uneven use of graphics resources (space, opacity, color, etc.) to discriminate parts of the data in focus from the rest (context) – more details in section 4.

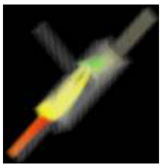
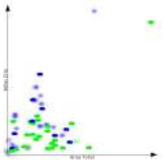
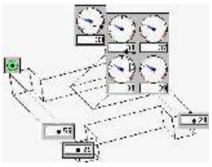
sample image					
graphics resource	opacity	style	color	frequency	space
section	4.1	4.2	4.3	4.4	4.5
related paper(s)	[38]	[4,17,18]	[5,6,8]	[25,26]	[37]

Table 2. Sample images of five different F+C visualization techniques (from left to right): RTVR-based volume rendering, two-level volume rendering, F+C visualization of simulation data, semantic depth of field, F+C process visualization.

However, the goal to visually bring out certain parts of the data in the visualization while reducing the visual appearance of all the rest very well matches the principal idea of F+C visualization. On the basis of a degree-of-interest function, an OTF can be specified by

$$\alpha(\texttt{data}[i]) = a(doi(\texttt{data}[i]))$$

with $a()$ being the identity map ($a(x){=}x$), a simple windowing function (see figure 1), or any other (potentially simple) monotonic map from $[0,1]$ to $[0,1]$. When $doi()$, for example, is defined on the basis of a scaled distance from a pre-defined iso-value – $doi(\texttt{data}[i]) = \max\{1 - s\,|\,\texttt{data}[i]{-}v_{iso}\,|\,,0\}$ –, then one of Levoy's OTF is regenerated with $a()$ being the identity map (or a simple window).

From many years of work on the question of how to specify an optimal opacity transfer function [39] we know that one simple data-dependent

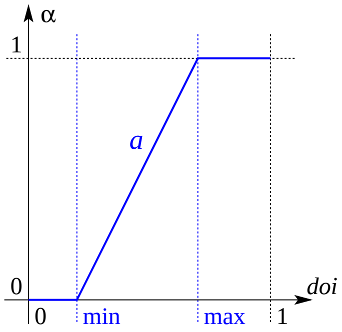


Fig. 1. A simple "window" often is sufficient to map *doi*-values to α-values: *doi*-values up to a certain minimum are mapped to a minimal value of opacity (usually 0), whereas *doi*-values above a certain maximum are mapped to 1 (completely opaque). In between, a linear map from *doi*-values to α-values is used.

function $doi()$ (or $\alpha()$) often is not sufficient to optimally discriminate focus from context in a visualization of 3D data, e.g., 3D medical data or 3D data from computational simulation. Instead, often sophisticated segmentation algorithms are used to do a proper focus–context discrimination before the actual visualization. The result of a segmentation algorithm usually is an n-valued object map $object()$, telling for each and every data item $\texttt{data}[i]$ which object it belongs to.

In two-level volume rendering (2lVR) [13, 17, 18], such an object map is used to improve the F+C visualization of 3D data: instead of directly deriving $doi()$ from the data, the degree of interest is defined on the basis of $object()$, i.e., for all the objects in the data (and not the singular data items) it is determined how interesting they are. This is done, because in many applications the 3D data anyhow is assumed to be composed of objects (in medical applications, for example, a dataset is assumed to be composed of bones, tissue, etc.). Therefore, the user automatically tends to formulate the focus–context discrimination in terms of the data objects (like "I'd like to see the bones and the blood vessels in the context of the skin."). For rendering, two values of opacity are used in two-level volume rendering: in addition to the $object()$-based (global) opacity $\alpha_{global} = a_{global}(doi(object))$, which yields the overall opacity for an object (depending on its degree of interest), a local (object-wise specified) OTF $\alpha_{local}(\texttt{data}[i], object(\texttt{data}[i]))$ is used to individually steer the visual appearance of every object.

For example, assuming $\alpha_{local}(.,1)$ to be a relatively sharp Levoy-OTF (comparably large s) and a_{global} to be the identity map, object 1 would be rendered like an iso-surface with its importance $doi(1)$ directly relating to its opacity. Through this separation of α_{global} and α_{local} the task of emphasizing certain parts of the data (semantical question) is separated from the question of how to render the different parts of the data (syntactical question). Accordingly, the parameterization of two-level volume rendering (adjustment of opacities) is much more intuitive (when compared to the use of a standard OTF only) and thus it is possible to achieve better results in shorter time. See figure 2 for a sample visualization of segmented 3D chest data with the focus on the lung-object.

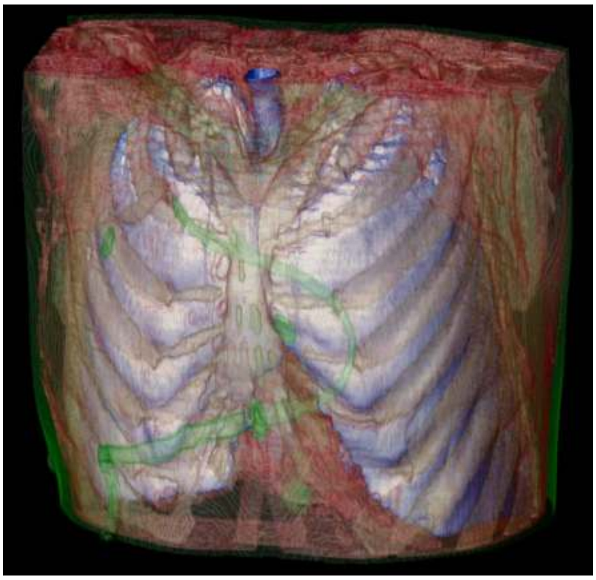


Fig. 2. A segmented CT-dataset of a chest, visualized using two-level volume rendering. Different values of overall opacity have been used for lung (completely opaque), bones (semi-transparent), and skin (very transparent).

In addition to opacity variations, two-level volume rendering also offers alterative ways to achieve a visual focus–context discrimination, for example, by varying the rendering style. But before we futhermore discuss 2lVR, another example for opacity-based F+C visualization is briefly described, which comes from the field of information visualization. Parallel coordinates [19–21] are a well-established technique for the visualization of high-dimensional data. Every n-dimensional data item is plotted as a polyline across n parallel axes in screen space such that a data item's polyline intersects the axes exactly at those points which relate to the data item's n attributes (see figure 3 for a sample image).

When many data items have to be shown simultaneously (tens of thousands or more), problems with overdraw easily occur: many pixels are covered by several (or even many) polylines. The resulting effect is that the visualization looses effectivness due to visual clutter − a classical scenario where F+C visualization can help. Using a DOI-based opacity to draw semi-transparent polylines over each other [15], an improved display is gained which allows for interactive analysis of the n-dimensional data (see figure 4). Note that the ability to interactively focus in such a F+C application is essential here to effectively exploit the visual superiority of this kind of visualization.

4.2 Reduced Style for Context Visualization

Another option of visually distinguishing between objects in focus and their context is to use different rendering styles. In two-level volume rendering, for example, it is possible to use different rendering techniques for differ-

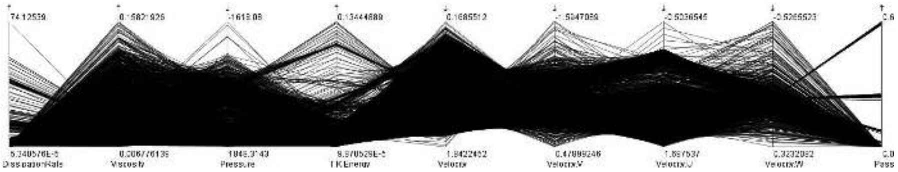


Fig. 3. 9-dimensional data from computational flow simulation (values from 5400 cells of a T-junction grid), visualized with parallel coordinates.

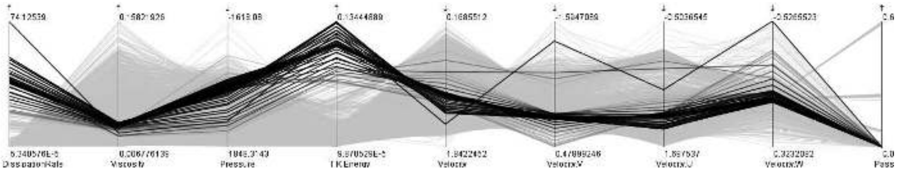


Fig. 4. DOI-based opacity used to visually separate some parts of the data "in focus" (characterized through rather large values of "T.K.Energy") from all the rest (context).

rendering style	**visualization properties**
α-compositing	conveys appearance of semi-transparent 3D medium (F), opacity difficult to control
shaded surface display	well conveys 3D form (F), good transparency control (C)
max. intensity proj.	good for complex forms (F), limited 3D appearance, good transparency control (C)
x-ray rendering	good for overview (C), complex opacity distribution
contour rendering	reduced appearance (C), little problems with occlusion
	F . . . good for focus visualization, C . . . good for context visualization

Table 3. Visualization properties of different rendering styles for 3D visualization together with a rough assessment of how they can be used for F+C visualization. Depending on whether the focus is inside the context (or outside), or if the context is of complex shape (or a rather coherent object), different combinations of rendering styles yield good results for F+C visualization (details in section 4.2).

ent objects in the data. On the global level, the different representations of the data objects are combined using standard compositing (α-blending) to achieve the final image. In addition to standard volume rendering, shaded surface rendering, maximum intensity projection (MIP), x-ray rendering, and non-photorealistic contour rendering can be used to depict an object. In table 3 some visualization properties are listed for different rendering styles in 3D visualization. A good opacity control, for example, favors the visualization as part of the context, because occlusion is easier controlled. The ability to visualize 3D form well, as another example, favors the visualization of data parts in focus. In the following we discuss several useful combinations of different rendering styles for F+C visualization.

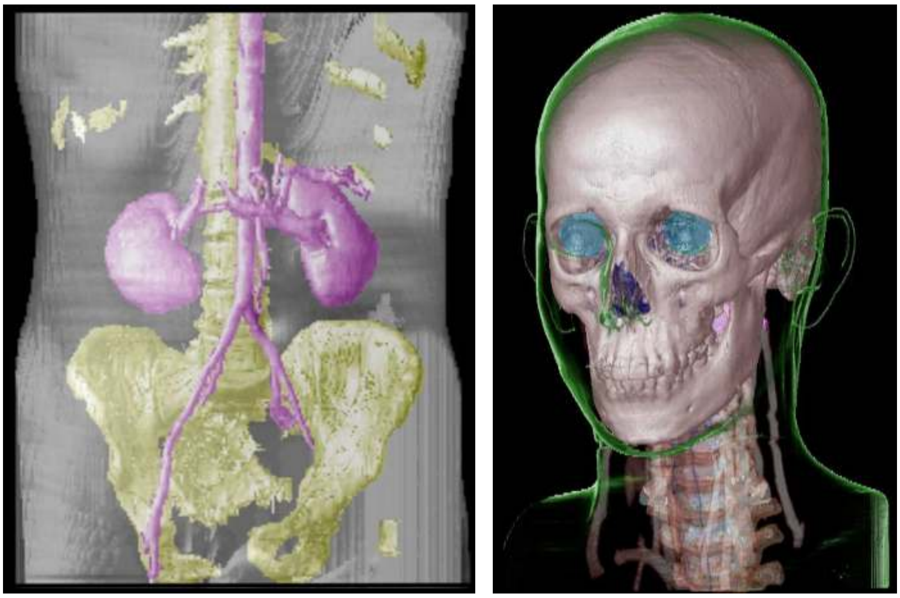


Fig. 5. MIP is useful for context visualization (skin on left side) because of its easy-to-trim opacity. Contour rendering works very well for context visualization (skin on the right) because of its reduced appearance (little problems with occlusion).

Shaded surface display very well acts as visualization of objects in focus, especially if the object(s) in focus are inside the context and, consequently, an opaque surface is used for visualization. This way, usually a strong and sharp appearance of the objects in focus is possible with a good communication of 3D shape. For context visualization, in such a case, the use of contour rendering and/or MIP is very interesting. Contour rendering works fine, because of its reduced appearance (lots of object parts are left away whereas only their contours are shown) and the fact that usually the middle parts of the visualization (where the objects in focus are shown) is rarly occluded (see figure 5, right image). Additionally, also MIP usually is useful for context visualization because of its easy-to-control opacity – only one data value per viewing ray is chosen for display, all object representatives share the same opacity (see figure 5, left image).

In case of context which is inside the objects in focus, like the bones acting as context to blood vessels (as the objects of interest in angiography), for example, shaded surfaces are doing a good job of focus visualization. The surfaces, however, need to be rendered semi-transparent (at least to some extent) to allow the user to peer inside and get visual access to the otherwise occluded context. MIP again is useful for the depiction of the context objects (good transparency control) – see figure 6 (left image) for a sample rendering of such a situation. Similarily, an x-ray simulation sometimes is useful for

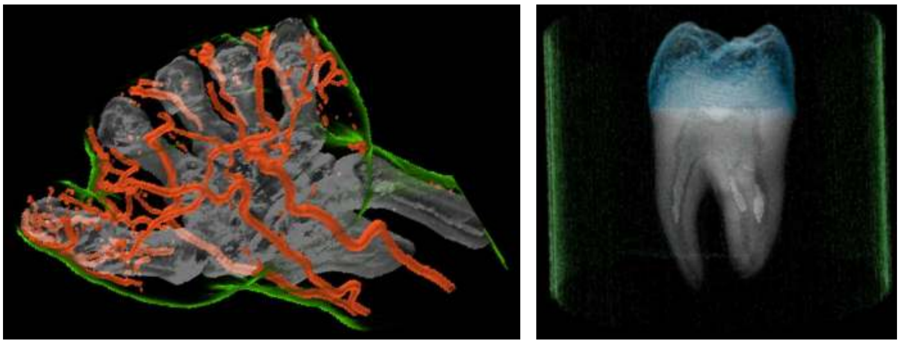


Fig. 6. F+C visualization of CT data of a human hand (left side): the objects of interest (blood vessels) are drawn as semi-transparent surfaces, whereas the bones are rendered using MIP. Contour rendering has been used to depict the skin. An x-ray simulation has been used to depit the dentine of the tooth on the right side (semi-transparent surface rendering of the adamantine and contour rendering of surrounding material).

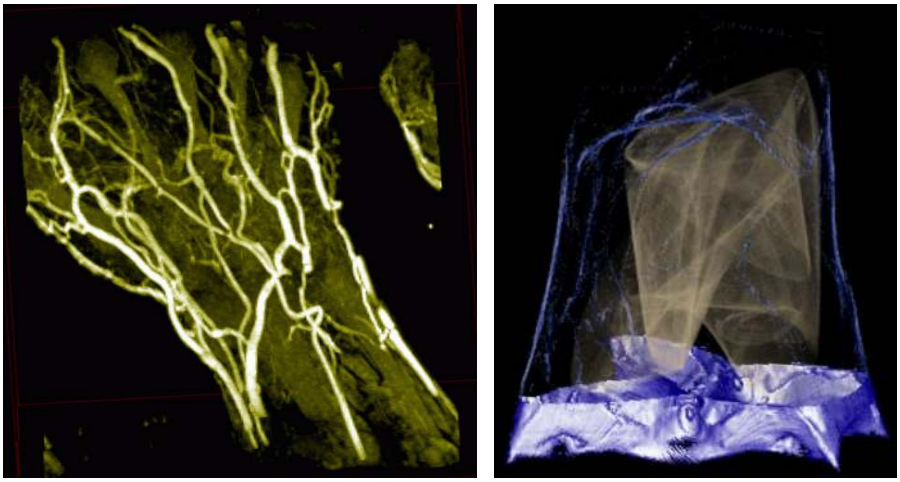


Fig. 7. Two examples of using MIP for complex objects in focus: the system of blood vessels in the CT hand data (left side) and a chaotic attractor within its basin of attraction on the right side (parts of the basin are shown as shaded surface whereas the rest of the basin is shown using contour rendering to minimize occlusion).

context visualization within objects of interest (see figure 6, right image, for an example).

In addition to context rendering, MIP is also useful for depicting objects in focus, especially is they are of complex shape (like an entire system of blood vessels or a chaotic attractor in a dynamical system [1]). In figure 7 two examples of such a visualization are given. On the left side MIP is used to show the blood vessels within the CT hand data. On the right side a complex attractor with fractal shape is visualized using MIP. The context (the basin of attraction, in this case) is shown in two ways: whereas the lower parts are shown as a shaded surface, the upper parts are provided using contour rendering only (to reduce problems with occlusion).

4.3 Eye-catching Colors for Focus Visualization

In addition to opacity and style as discussed in the previous two sections, also color is effectively used to focus within a visualization. From perseptual research on preattentive vision [44, 45] we know, for example, that human observers can very quickly "find" colored parts in a visualization of otherwise gray-scale representations – the "search" succeeds even before the observer actually starts searching in an active manner, i.e., in a time usually shorter than 200 ms from stimulus. Accordingly, coloring some parts of a visualization (which are in focus) and showing all the rest in a gray-scale way, also works fine as a F+C visualization technique.

Gresh et al. presented a system called WEAVE [12] which uses this style of F+C visualization for the display of complex simulation data of a beating human heart. Different views of different types of visualization (a scatterplot, a 3D view, etc.) are used to depict and analyse the multi-dimensional simulation data. To assess the large amount of data, the user is able to select certain data subsets of special interest. These parts of the data are then drawn in color whereas all the rest is displayed in gray-scale style. First of all, the colored parts of the visualization immediately stand out of every view where this kind of focus–context discrimination is used. Secondly, the coloring is done consistently across all the views, so visual linking is established between the views. The same color always indicates the same selection of the data (focus), just visualized differently according to the different views (thereby different characteristics/dimensions of the same data are visualized in the different views). In information visualization this approach is called *linking and brushing* (L&B) – "brushing", because the process of selecting a data subset of interest usually is done directly on one of the linked views, similar as in a drawing program.

In a system called SimVis [5–8], we use this approach to visualize data from computational simulation of processes in the automotive industry. An extended brushing technique called *smooth brushing* [6] allows for a gradual transition of the *doi*-function from the subset of interest (focus, $doi = 1$) to the rest (context, $doi = 0$). For visualization, a gradual reduction of color saturation is used to reflect the continuously diminuishing degree of interest. See figure 8 for a sample result of this kind of visualization, where a data subset of high pressure and high velocity was selected using smooth brushing in the scatterplot on the right. On the left, a visually linked 3D view shows where those areas of high pressure and high velocity lie in the 3D flow domain (a model of a catalytic converter). In addition to the DOI-based variations

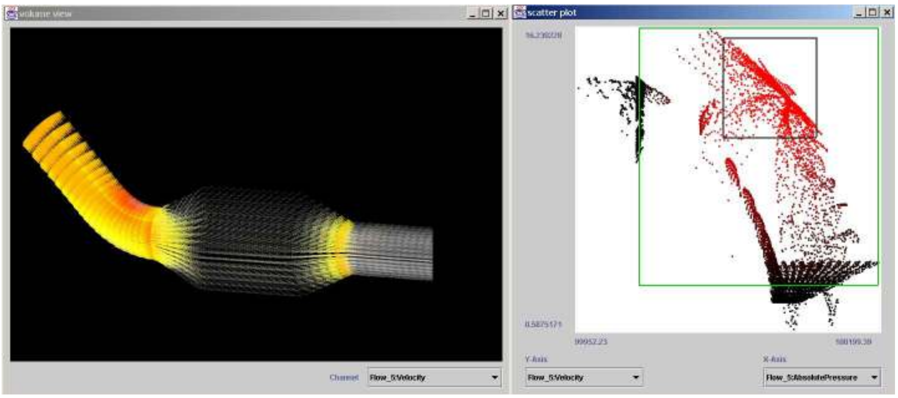


Fig. 8. F+C visualization of CFD data (flow through a catalytic converter). A data subset, represented by values of high pressure and high velocity, has been selected by smooth brushing on the scatterplot on the right. Gradual changes of color saturation on the left (in the 3D view) represent the smooth degree of interest.

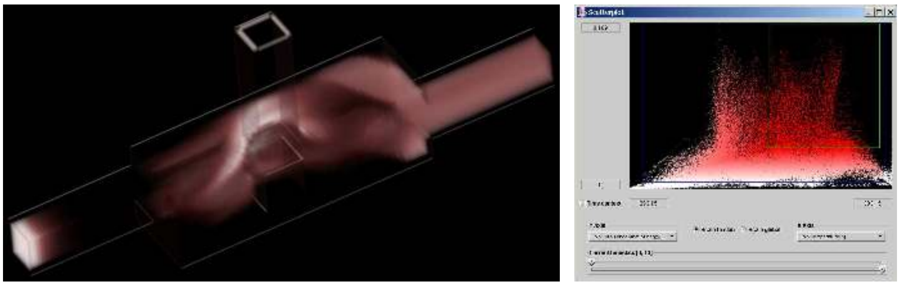


Fig. 9. Visualization of flow through a T-junction. The visualization focuses on a flow subset which is characterized by high temperature and high turbulent kinetic energy. The junction-geometry is added as context (contour rendering).

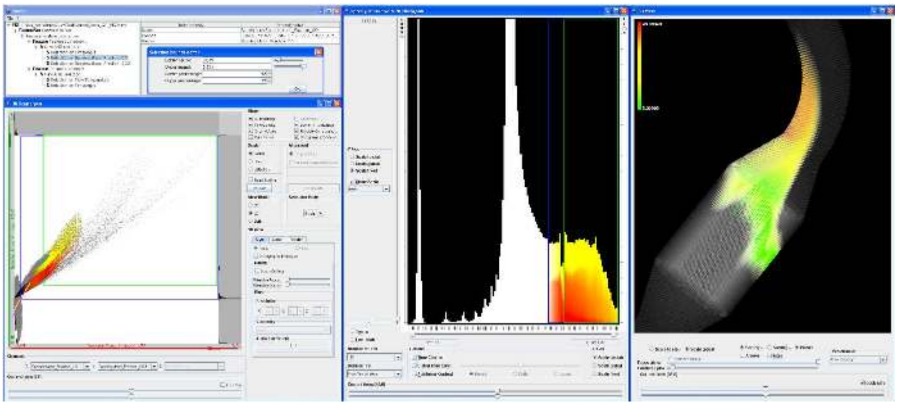


Fig. 10. Flow through a diesel particle filter: a scatterplot and a histogram are used to focus on hot flow which also exhibits large amounts of carbon-oxides (oxidation products CO & CO_2). The 3D view shows the spatial location of the oxidation front at time 35 secs. after the simulation start (color shows velocity magnitudes).

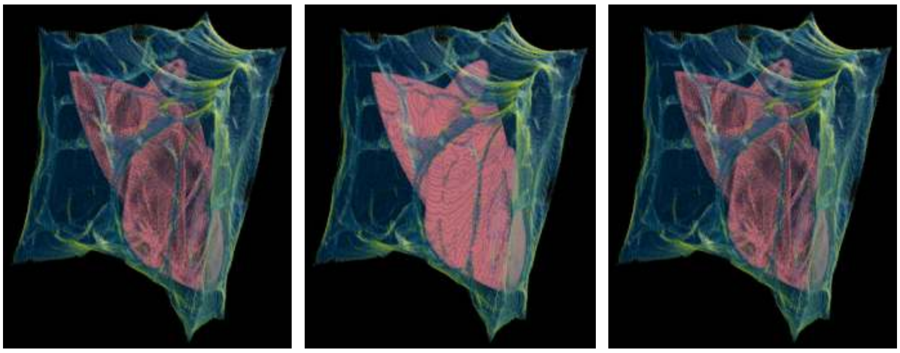
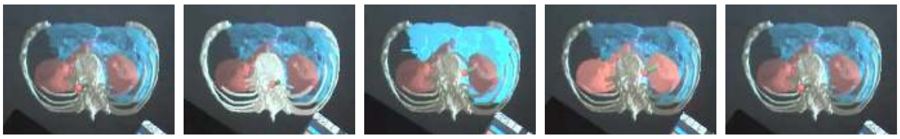


Fig. 11. Object high-lighting in the course of object selection: before the selection ($t = t_0$, left image), right after the selection of the chaotic attractor ($t = t_0 + \approx \frac{1}{2}$sec., middle image), and a little later after high-lighting ($t = t_0 + \approx 1$sec., right image).



— bones liver kidneys —

Fig. 12. Several snapshots from a video which was taken through a session where the user moved a 3D pointing device across a 3D dataset of a human chest in a virtual environment. Visual object high-lighting reflects the current 3D position of the 3D pointing device which is very useful to efficiently position the device in 3D space during object selection.

of color saturation also the glyph size is varied according to the data item's degree of interest (the more interesting the bigger the glyphs used).

In figure 9 volume rendering on the basis of α-compositing [16] was used to depict a subset of a flow through an extended T-junction (characterized through values of high temperature and high turbolent kinetic energy, see scatterplot on the right). In figure 10 another sample snapshot from an interactive visual analysis session (using SimVis) is shown [8]. A scatterplot (on the lower left) and a histogram (in the middle) are used to focus on the oxidation front within a diesel particle filter (characterized by lots of carbon oxides, i.e., oxidation products, and high temperatures). The linked 3D view shows the spatial location of the oxidation front at a certain point in time (35 secs. after the simulation start). In the upper left a tree view is visible which provides direct access to the focussing information, i.e., the *doi* attributions of the data as related to the current analysis step.

In a system called GeoSpace [35], user queries are answered visually through high-lighting the data parts in focus, i.e., those data items which positively respond to the user query. High-lighting is done, for example, by increasing the color lightness. Thereby the selected data subsets visually stand out from the rest of the depiction. In two-level volume rendering, this approach is used to provide feedback to the user during object selection in the 3D domain. For a short time after the selection of an object in the scene, the selected object is shown with a different transfer function (increased color lightness, increased opacity). Thereby a clear visual linking between an object's name or ID and its visual representation as part of the visualization is established (see figure 11).

This is especially useful, when volume visualization is performed in a virtual environment. In this case, especially when 3D objects have to be selected directly through 3D user interaction (for example, by the use of a 3D pointing device), object high-lighting greatly supports the interactive placement of the 3D pointing device. While moving the pointing device, the user immediately gets feedback on which object the pointer currently is pointing towards. Thereby, the user is easily able to efficiently select the one object of special interest without a lot of trial and error (which otherwise is quite normal for 3D direct selection). Figure 12 gives a number of snapshots of a video which was taken during a session where the user moved a 3D pointing device around a 3D dataset of a human chest with different segmented parts of the data. Whenever the 3D pointing device enters another object in the scene, the respective object is rendered in a high-lighted fashion according to the above mentioned transfer function alternation.

4.4 Band-limited Context

Before we come back to the traditional way of F+C visualization (section 4.5), we furthermore describe one additional way of visually discriminate the visualization of data parts in focus from all the rest (context). Again (as compared to the use of eye-catching colors for focus visualization, see section 4.3) it is an argument from perceptual psychology which motivates this alternative approach: the difference between a sharp and blurred object depiction efficiently can be used for visual focus–context discrimination [25, 26], a technique we call *semantic depth of field* (SDOF). In a user study we could prove that the perceptual identification of sharp objects among blurred others indeed is preattentive [27], i.e., is performed by the human perceptual system within a very short time ($< \approx 200\,\mathrm{ms}$).

In 2D, the basic idea of SDOF (semantic depth of field) is (a) to assume a camera model with a depth-of-field effect in rendering and (b) to virtually displace parts of the visualization along the viewing axis to achieve a blurred or sharp depiction of irrelevant and relevant parts of the data, respectively (see figure 13). With a lens-based camera, objects are only displayed sharply if they are located at the focal distance from the lens. Object which are displaced along the viewing axis are blurred according to their distance from the lens. Therefore, the displacement in the depth direction is done according to the degree-of-interest values which are associated to all the data elements (and

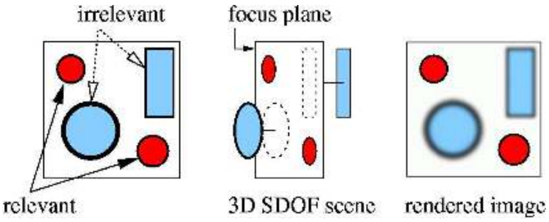


Fig. 13. The basic idea of SDOF for 2D visualization: assuming a lens-based camera model for rendering, the visualization objects are virtually moved back or forth along the viewing direction to achieve a blurred and sharp depiction for irrelevant and relevant data items, respectively.

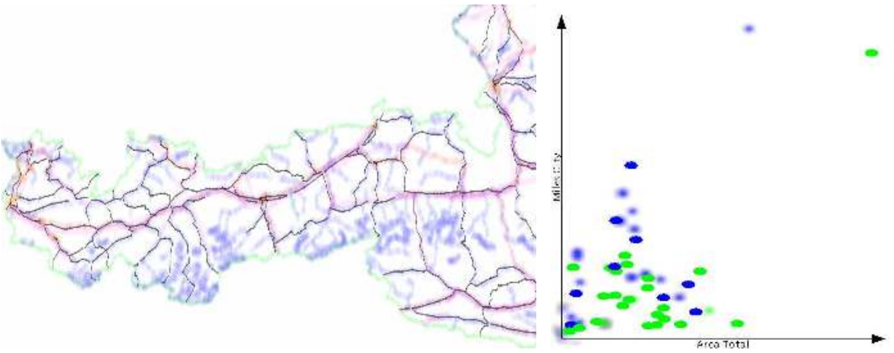


Fig. 14. Two examples of an SDOF visualization: streets standing out of an SDOF map visualization on the left (other parts of the map blurred) and a scatterplot with SDOF effect on the right.

not as a spatial function of the data as it is in real-world photography). For 3D visualization, a similar SDOF model exists [25, 26].

Confronted with the result of such a SDOF visualization (see figure 14), the user can immediately identify the data subsets in focus (similar to photography where sharpness also directly correlates to the fact of being in focus). Therefore, this kind of F+C visualization becomes especially useful when the DOI assignement is done implicitly, e.g., through brushing of invisible dimensions (with a range slider, for example) or through defining the DOI value by how well a data item matches a certain user query [2]. In all these cases the first task a user usually performs is to identify which data items actually have been assigned a high DOI value (and which not). With SDOF this is easily possible as the sharp parts of the visualization, representing the relevant data items, stand out of the depiction automatically.

4.5 More Space for Details

After discussing four alternative ways of realizing focus–context discrimination in visualization (based on the variation of opacities, styles, colors, and

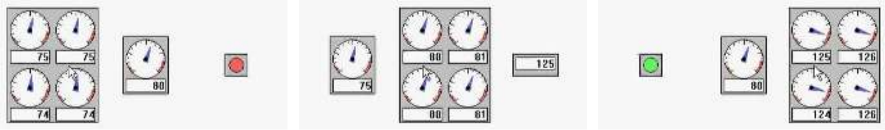


Fig. 15. F+C process visualization: depending on where the user points, the virtual instruments are drawn at a smaller or larger level of detail (from left to right: the pointer is moved from left to right).

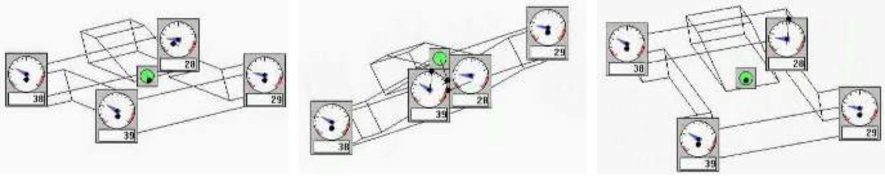


Fig. 16. 3D anchoring and collision avoidance in F+C process visualization: virtual instruments are placed at the screen-projection of that 3D point which is related to the data origin, for example, a sensor (3D anchoring); to avoid cluttering due to overlapping dials a physically-based spring model is used to relocate instruments such that they do not overlap (collision avoidance).

frequencies), we come back to the traditional way of F+C visualization, i.e., to the variation of magnification factors within a single image. This kind of completes the picture of our generalization. In section 2 we already discussed the extensive block of literature on this kind of F+C visualization. In our case, we have applied this classic principle to process visualization [37] where this has not been done before.

In process visualization, data which is streaming in from a larger number of processes has to be presented to a user such that process surveillance as well as interactive analysis is possible. In analogy to traditional process visualization, where processes are visualized with analog instruments like gauges or other display devices, programs for process visualization (at least partially) mimic this kind of visualization with virtual instruments. One disadvantage of virtual instruments is that they take up quite a lot of screen space. When multiple streams of process data have to be shown simultaneously, not enough screen space is available to show all the data with regularily sized instruments. In such a situation, distortion-based F+C visualization becomes useful.

To achieve F+C visualization of process data, several levels of detail have been designed for the different virtual instruments in use. The different levels of detail use different amounts of screen space, ranging from a small lamp, color-coding the process data, up to a fully fletched virtual instrument, using a hundred times the amount of screen space as compared to the lamp. If not all of the data can be shown at the highest level of detail simultaneously (due to lack of screen space), different levels of detail can be combined according to DOI values of the different data items. See figure 15 for an example, where three streams of process data are visualized. DOI values inversely correlate to the distance between the respective virtual instrument and the pointer which is interactively moved by the user (from left to right). Thereby, those virtual instruments which are nearest to the pointer are displayed at the highest level of detail whereas with increasing distance from the pointer lower levels of detail are used.

In process visualization, data usually originates at concrete 3D locations like a sensor at a certain place or a simulation output with a specific 3D position. Accordingly, the visualization of process data can be organized on the screen such that this relation between the virtual instruments and the related 3D model becomes obvious. In a prototype implementation of F+C process visualization, we first draw the underlying 3D model as a wire-frame rendering. Then, the virtual instruments are shown on top of the wire-frame model at those screen coordinates which correlate to the screen-projection of the corresponding 3D locations of the data sources (see figure 16).

With such a layout strategy (called *3D anchoring* – the virtual instruments are "anchored" at their respective 3D source locations), it can easily happen that screen projections of sensor locations lie near each other such that a naïve implementation of 3D anchoring would cause overlapping virtual instruments. In our prototype implementation we therefore use a physically-based spring model to resolve for non-overlapping instruments (collision avoidance). See figure 16 for three snapshots of this prototype which were taken while the user rotated the 3D model (the black dots, which are connected to the centers of the instruments with black lines, mark the screen-projections of the 3D anchors, i.e., the 2D locations where in the optimal case the virtual instrument should be displayed).

5 Interaction

Focus+context visualization requires interaction. Most important, the user needs to have interactive means to focus in a F+C visualization, i.e., he or she needs to steer which parts of the data have to be shown in focus. Accordingly, focussing also includes interactive means to navigate in the visualization, i.e., to change from the visualization of one part of the data (in focus) to another. For applications of F+C visualization, different approaches to focussing are available (see table 4 for an overview of some of them), which can be classified with respect to several different aspects. One question is of whether focussing is done directly on the visualization (or not). Another quesion is of whether focussing is done explicitly, i.e., by either directly brushing the data items of interest or naming them explicitly. Thirdly, the question of whether the user actively performs the focussing (or the system does it for the user) also classifies the different approaches to focussing.

Most intuitive, *explicit selection* of especially interesting data subsets *directly on the view* results in a (new) specification of the current focus. Prominent examples of this kind of focussing are *brushing* on the one side (as used,

focussing	**action**	**selection**	**user**	**sample applications**
brushing	on the view	explicit	active	SimVis [5–8], parallel coordinates [15]
pointing				RTVR [38], process visualization [37]
selection	off-view			SDOF [25,26], RTVR [38]
range slider		implicit		SimVis [5–8], SDOF [25,26]
querying				SimVis [5–8]
plot-based		both	passive	SDOF [25,26]
alerting				process visualization [37]

Table 4. Different approaches to focussing – techniques can be classified according to whether they act directly on the view (or not), their definition is explicit (or implicit), or whether they are triggered by the user (or not). This differentiation is discussed in more detail in section 5.

for example, in the previously described SimVis system) and *pointing* on the other side (used in F+C process visualization as well as in 3D visualization using RTVR). Similarily, the user can *explicitly focus* by selecting objects through an *off-view list of objects* (as used in volume visualization using RTVR, for example, and the SDOF-visualized map viewer where layers can be selected off-view).

More complex, and a little less intuitive, *implicit selection* also serves for focussing. In the simpler case, selections on invisible axes can be used to describe what currently is most interesting (as also used in SimVis, for example). Alternatively, also complex queries can be used to achieve implicit focussing. Again SimVis is an example: a so-called feature definition language has been developed for the purpose of formally describing what actually is of greatest interest to the user [5].

In addition to methods where the user actively steers which parts of the data are to be visualized in focus, there are other cases, where the system has this role. In a tutoring system, for example, a predefined plot describes which parts of the visualization are in focus at which point in time. This kind of focussing was used in a chess tutoring system with the purpose of showing historic competitions to moderately experienced users (see figure 17). In F+C process visualization it is possible to let the system assign DOI values according to whether (or not) the values of a certain sensor lie inside (or outside) a certain safety interval. In case of an alert (value out of range) the user immediately is confronted with a F+C display where most visual emphasis is put on the values in question.

6 Summary and Conclusions

Taking a step back, we can try to round up the matters discussed up to now and to summarize the most imporant points addressed. In the beginning we started out with a discussion of the well-established approach of *focus+context*

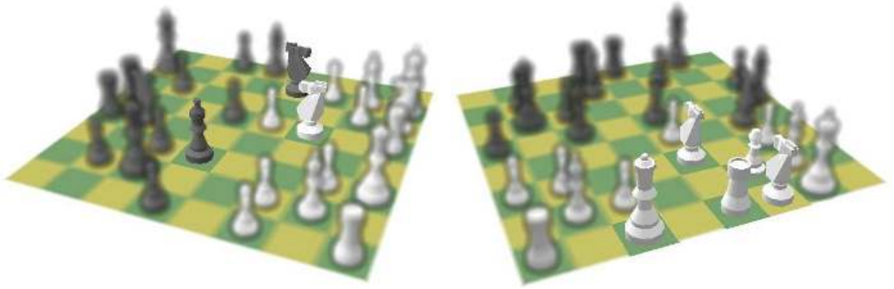


Fig. 17. SDOF-visualized chess tutoring system: through selective sharpness the system shows which pieces threaten (left image) or cover (right image) the white horse on E3.

visualization (F+C visualization) as known from information visualization. It is usually associated with the process of providing more space in a visualization for the detailed depiction of some selected parts of the data (those in focus) while still showing the rest of the data in reduced size to provide context information for better orientation and navigation.

This idea of integrating data subsets in focus with their respective context within one visualization also can be found in other fields, especially in scientific visualization. There, however, usually other means than space distortion are used to achieve F+C visualization. In scientific visualization the spatial arrangement of a visualization is tightly coupled with the spatial arrangement of the data origin, e.g., the 3D layout of patients in medical applications or the 3D setup of a flow simulation, and therefore usually resists uneven distortions. In volume visualization, for example, the use of opacity is varied to achieve F+C visualization of 3D data. In the 3D visualization of segmented data (two-level volume rendering, 2lVR), different styles are used to graphically distinguish between objects in focus and their context. Non-photorealistic contour rendering, for example, is very useful for context visualization. In the visualization of data from computational simulation (WEAVE, SimVis), the use of eye-catching colors (within a gray-scale context) also very well serves for F+C visualization. Similarily, the differentiation between a sharp and blurred depiction can yield to F+C visualization (SDOF). All this variety of possible realizations of focus+context visualization yields to a more general definition of F+C visualization: *focus+context visualization is the uneven use of graphics resources*, such as space, opacity, color, frequencies, and style, *for visualization* with the purpose to visually discriminate those parts of the data in focus from all the rest.

A discussion of several concrete examples of different types of F+C visualization shows that often several graphics resources are used to do the focus–context discrimination. In F+C volume visualization by the use of RTVR and 2lVR, for example, in some cases all three of opacity, rendering styles, and coloring are varied to achieve F+C visualization (see figure 5, left side, for a sample image). In F+C visualization of 3D data from computational flow visualization (SimVis), coloring, opacity, and glyph size are adjusted according to the DOI values of the data to achieve the desired visual discrimination (see figure 8, left side, for a sample image). Looking through the glasses of our generalized definition of focus+context visualization at the very broad field of applications shows how useful this approach of graphically integrating data subsets in focus and their respective context within a visualization actually is and how general its applicability is.

In addition to the discussion about different ways to graphically discriminate focus from context, also the interactive aspect of F+C visualization is discussed. Once, focus+context visualization is established, it immediately becomes essential to provide sufficient interactive means for focussing, i.e., to select which parts of the data actually are to be drawn in focus or to navigate through a F+C display. Different options of how to categorize focussing with respect to how it is done (on the view vs. off-view focussing; explicit vs. implicit selection; active/passive user) help to give an overview about available strategies. Another way of looking at focussing, however, is to differentiate user goals: whereas in one case the user wants to see more (details) of certain data subsets (→ space distortion, style variations), in other cases the user just wants to visually emphasize the graphical depiction of certain parts (→ opacity, color variations). In again other cases, the visualization goal is to visually attract the user towards a certain subset of the visualization (→ SDOF, coloring, space distortion). Sometimes, these goals do overlap in an application or are followed upon each other during analysis (first the user needs to be attracted, for example, to a sensor out of range, then the user wants to investigate this sensor data in more detail).

Despite the main result that focus+context visualization indeed is in general applicable and useful (almost regardless of the application field), another conclusion of this work is that scientific visualization and information visualization do not lie far apart from each other, but can mutually support each other. There are very good ideas on both sides and visualization systems which integrate approaches from both fields can gain superb advantages over pure SciVis- or InfoVis-solutions [14].

7 Acknowledgments

This work is based on a lot of related work which would have been impossible without the great contributions of many. To name just a few of them, grateful thanks go to Lukas Mroz, Csébfalvi Balázs, Gian-Italo Bischi, Berk Özer, Anton Fuhrmann, Helmut Doleisch, Martin Gasser, Matej Mlejnek, Markus Hadwiger, Florian Ledermann, Robert Kosara, Silvia Miksch, Krešimir Matković, Wolfgang Rieger, Wolfgang Meyer, and especially to M. Eduard Gröller and Werner Purgathofer for their patient advice throughout all the years which have been related to this work. For funding, thanks go to K plus, an Austrian

governmental funding program, which is supporting VRVis since year 2000 and thus also is responsible that most of the work discussed here actually could be done. For more information, see related papers [4–6, 8, 15–18, 25–27, 37, 38] and `http://www.VRVis.at/vis/`.

References

1. Hamdy Agiza, Gian-Italo Bischi, and Michael Kopel. Multistability in a dynamic Cournot game with three oligopolists. *Mathematics and Computers in Simulation*, 51:63–90, 1999.
2. Christopher Ahlberg and Ben Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with Starfield displays. In *Proc. of ACM CHI'94 Conf. on Human Factors in Computing Systems*, pages 313–317, 1994.
3. M. Sheelagh Carpendale, David Cowperthwaite, and David Fracchia. 3-dimensional pliable surfaces: For the effective presentation of visual information. In *Proc. of the ACM Symp. on User Interface Software and Technology*, Information Navigation, pages 217–226, 1995.
4. Balázs Csébfalvi, Lukas Mroz, Helwig Hauser, Andreas König, and Eduard Gröller. Fast visualization of object contours by non-photorealistic volume rendering. *Computer Graphics Forum*, 20(3):C 452–C 460, 2001.
5. Helmut Doleisch, Martin Gasser, and Helwig Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proc. of the Joint* IEEE TCVG – EG *Symp. on Visualization*, pages 239–248, 2003.
6. Helmut Doleisch and Helwig Hauser. Smooth brushing for focus+context visualization of simulation data in 3D. *Journal of WSCG*, 10(1):147–154, 2002.
7. Helmut Doleisch, Michael Mayer, Martin Gasser, Peter Priesching, and Helwig Hauser. Interactive feature specification for simulation data on time-varying grids. In *Proc. of Conf. Simulation and Visualization*, pages 291–304, 2005.
8. 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 *Proc. of the Joint* IEEE TCVG – EG *Symp. on Visualization*, pages 91–96, Konstanz, Germany, May 2004.
9. William Augustus Farrand. *Information Display in Interactive Design*. PhD thesis, University of California, Los Angeles, CA, 1973.
10. George Furnas. The Fisheye view: A new look at structured files. Technical Memorandum #81-11221-9, Bell Labs, 1981. Reprinted in Card et al., Readings in Information Visualization: Using Vision to Think.
11. George Furnas. Generalized Fisheye views. In Marilyn M. Mantei and Peter Orbeton, editors, *Proc. of the ACM Conf. on Human Factors in Computer Systems*, SIGCHI Bulletin, pages 16–23, 1986.
12. Donna Gresh, Bernice Rogowitz, Raimond Winslow, David Scollan, and Christina Yung. WEAVE: A system for visually linking 3-D and statistical visualizations, applied to cardiac simulation and measurement data. In *IEEE Visualization 2000*, pages 489–492, 2000.
13. Markus Hadwiger, Christoph Berger, and Helwig Hauser. High-quality two-level volume rendering of segmented data sets on consumer graphics hardware. In *Proc. of IEEE Visualization 2003*, pages 301–308, 2003.

14. Helwig Hauser. Towards new grounds in visualization. *ACM SIGGRAPH Computer Graphics*, 39(2), 2005.
15. Helwig Hauser, Florian Ledermann, and Helmut Doleisch. Angular brushing for extended parallel coordinates. In *2002 IEEE Symp. on Information Visualization (InfoVis '02)*, pages 127–130. IEEE, October 2002.
16. Helwig Hauser and Matej Mlejnek. Interactive volume visualization of complex flow semantics. In *Proc. of the 8th Fall Workshop on Vision, Modeling, and Visualization*, pages 191–198, München, Germany, November 2003.
17. Helwig Hauser, Lukas Mroz, Gian-Italo Bischi, and Eduard Gröller. Two-level volume rendering - fusing MIP and DVR. In *Proc. of IEEE Visualization 2000*, pages 211–218, 2000.
18. Helwig Hauser, Lukas Mroz, Gian-Italo Bischi, and Eduard Gröller. Two-level volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 7(3):242–252, 2001.
19. Alfred Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–92, 1985.
20. Alfred Inselberg. A survey of parallel coordinates. In Hans-Christian Hege and Konrad Polthier, editors, *Mathematical Visualization*, pages 167–179. Springer Verlag, Heidelberg, 1998.
21. Alfred Inselberg and Bernard Dimsdale. Parallel coordinates: a tool for visualizing multidimensional geometry. In *Proc. of IEEE Visualization '90*, pages 361–378, 1990.
22. Naftali Kadmon and Eli Shlomi. A polyfocal projection for statistical surfaces. *The Cartography Journal*, 15(1):36–41, 1978.
23. T. Alan Keahey and Edward Robertson. Techniques for non-linear magnification transformations. In *1996 IEEE Symp. on Information Visualization (InfoVis '96)*, pages 38–45. IEEE, 1996.
24. T. Alan Keahey and Edward Robertson. Nonlinear magnification fields. In *IEEE Symp. on Information Visualization (InfoVis '97)*, pages 51–58. IEEE, October 1997.
25. Robert Kosara, Silvia Miksch, and Helwig Hauser. Semantic depth of field. In *Proc. of the 2001 IEEE Symp. on Information Visualization (InfoVis 2001)*, pages 97–104. IEEE Computer Society Press, 2001.
26. Robert Kosara, Silvia Miksch, and Helwig Hauser. Focus + context taken literally. *IEEE Computer Graphics and Applications*, 22(1):22–29, 2002.
27. Robert Kosara, Silvia Miksch, Helwig Hauser, Johann Schrammel, Verena Giller, and Manfred Tscheligi. Useful properties of semantic depth of field for better F+C visualization. In *Proc. of the Joint* IEEE TCVG – EG *Symp. on Visualization*, pages 205–210, 2003.
28. Matthias Kreuseler, Norma López, and Heidrun Schumann. A scalable framework for information visualization. In *Proc. Information Vizualization*, pages 27–36, Salt Lake City, USA, October 2000. IEEE.
29. John Lamping and Ramana Rao. The hyperbolic browser: A focus + context technique for visualizing large hierarchies. *Journal of Visual Languages and Computing*, 7(1):33–35, 1996.
30. John Lamping and Ramana Rao. Visualizing large trees using the hyperbolic browser. In Michael J. Tauber, editor, *Proc. of the 1996 Conf. on Human Factors in Computing Systems, CHI 96: April 13–18, 1996, Vancouver, BC, Canada*, pages 388–389, New York, NY 10036, USA, April 1996. ACM Press.

31. John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Proc. CHI'95*. ACM, 1995.
32. Ying Leung. Human-computer interface techniques for map based diagrams. In *Proc. of the Third International Conf. on Human-Computer Interaction*, volume 2 of *Designing and Using Human-Computer Interfaces and Knowledge Based Systems; Graphics*, pages 361–368, 1989.
33. Ying Leung and Mark Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interaction*, 1(2):126–160, June 1994.
34. Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics & Applications*, 8(5):29–37, 1988.
35. Ishantha Lokuge and Suguru Ishizaki. Geospace: An interactive visualization system for exploring complex information spaces. In *Proc. of the ACM CHI '95 Conf. on Human Factors in Computing Systems*, 1995.
36. Jock Mackinlay, George Robertson, and Stuart Card. The perspective wall: Detail and context smoothly integrated. In *Proc. of ACM CHI Conf. on Human Factors in Computing Systems*, Information Visualization, pages 173–179, 1991.
37. Krešimir Matković, Helwig Hauser, Reinhard Sainitzer, and Eduard Gröller. Process visualization with levels of detail. In Pak Chung Wong and Keith Andrews, editors, *Proc. IEEE Symp. Information Visualization, InfoVis*, pages 67–70. IEEE Computer Society, 28–29 October 2002.
38. Lukas Mroz and Helwig Hauser. RTVR - a flexible java library for interactive volume rendering. In *IEEE Visualization 2001*, pages 279–286, October 2001.
39. Hans-Peter Pfister, Bill Lorensen, Chandrajit Bajaj, Gordon Kindlmann, William Schroeder, Lisa Sobierajski-Avila, Ken Martin, Raghu Machiraju, and Jinho Lee. Visualization viewpoints: The transfer function bake-off. *IEEE Computer Graphics and Applications*, 21(3):16–23, 2001.
40. George Robertson and Jock Mackinlay. The document lens. In *Proc. of the ACM Symp. on User Interface Software and Technology*, Visualizing Information, pages 101–108, 1993.
41. Manojit Sarkar and Marc Brown. Graphical fisheye views of graphs. In *Proc. of ACM CHI'92 Conf. on Human Factors in Computing Systems*, Visualizing Objects, Graphs, and Video, pages 83–91, 1992.
42. Manojit Sarkar, Scott Snibbe, Oren Tversky, and Steven Reiss. Stretching the rubber sheet: A metaphor for visualizing large layouts on small screens. In *Proc. of the ACM Symp. on User Interface Software and Technology*, Visualizing Information, pages 81–91, 1993.
43. Robert Spence and Mark Apperley. Data base navigation: An office environment for the professional. *Behaviour and Information Technology*, 1(1):43–54, 1982.
44. Anne Treisman. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31:156–177, 1985.
45. Colin Ware. *Information Visualization: Perception for Design.* Morgan Kaufmann Publishers, 2000.