Explode to Explain – Illustrative Information Visualization

Martin Luboschik and Heidrun Schumann
Institute of Computer Science, University of Rostock, D-18051 Rostock, Germany
{luboschik, schumann}@informatik.uni-rostock.de

Abstract

Due to complexity, modern visualization techniques for large data volumes and complex interrelationships are difficult to understand for non-expert users and even for expert users the visualization result may be difficult to interpret. Often the limited screen space and the risk of occlusion hinders a meaningful explanation of techniques or datasets by additional visual elements. This paper presents a novel way how views from information visualization can be adapted by the use of the well-known illustrative technique "exploded view", to successfully face the problems described above. The application of exploded views gains screen space for an explanation in a smart way and acts explanatory itself. With our approach of illustrating visual representations, the understanding of complex visualization techniques is eased and new comprehensible views on data are given.

1. Introduction

An important aim of information visualization is the visualization of constantly growing large data volumes and the visualization of complex interrelationships. This aim has been successfully achieved by various modern visualization techniques. For example, the use of very small visual primitives arranged in a compact manner is such a successful approach (e.g. pixel-oriented techniques).

However, avoiding visual clutter that is introduced by visualizing large volumes of data and their relationships is an ongoing topic. Furthermore, the growing sophistication and complexity of those visualization techniques leads to images that are difficult to comprehend for non-expert users. Moreover, complex visualization techniques like pixelmaps or treemaps use every pixel of the display, and therefore limit the possibilities for annotations, since additional visual elements may occlude crucial information.

Exploded views are a well-known illustration technique to facilitate a better insight into complex geometries or compact images. They have been successfully applied to different fields but are rarely considered in information visualization. In this paper we propose the application of exploded views for information visualization techniques to face the problems above. We initially describe the idea of exploded views and introduce a new classification scheme (Section 2). Furthermore, we demonstrate how the illustrative technique of exploded views can be used to explain *complex visual arrangements and visualization techniques* to non-expert users (Section 3). In addition, we show how exploded views can also be used to generate new views on data and thus are helpful in the explanation of *data characteristics* (Section 4). The results of our approach are discussed in Section 5.

2. Background

The exploded view is a specific arrangement technique originating from the field of technical drawings. Its principal aim is to show *all* components of complex constructions and their relationships in 3D minimizing occlusions – without changing attributes like opacity. Such views are called meaningful if and only if all details are visible and relationships are depicted unambiguously [14]. To achieve visibility, scaling may vary among visual primitives – restricted by a limited range of scaling factors to avoid identification-problems.

The generation of an exploded view requires the specification of visual primitives to be exploded, an optional grouping of those primitives and the parameterization of the explosion process. The center of explosion, the explosion level and the type have to be specified. The exploded view is based on a visible scaling of distances between geometric primitives and the center of explosion. The explosion level is used for scaling the distances and the type determines whether this explosion is performed in all directions (omnidirectional) or in a single direction (unidirectional).

In [8], the process of volume splitting is distinguished between logical and geometrical splitting. This is a useful categorization, characterizing whether the volume splitting is geometry driven or also controlled by semantic knowledge. Therefore, we introduce a similar classification for exploded views in information visualization:

Geometrical explosion: An explosion is generated and controlled without any knowledge about the underlying data. All visual primitives are treated the same way when increasing their distance to the center of explosion. Optional grouping of visual primitives is only based on their location.

Semantical explosion: Knowledge about the underlying data is necessary to generate and control the exploded view. For example, the explosion process is controlled by logical dependencies of certain components. Optional grouping of visual primitives is based on the given data characteristics, rather than on geometric relationships.

Hybrid explosion: Knowledge about the underlying data is used to define elements to be exploded and for parameterization. The explosion process itself is geometrical and needs no further knowledge. Optional grouping of visual primitives is based on their location, or on data characteristics as well. All visual primitives are treated the same way when increasing their distance to the center of explosion.

Exploded views have been applied in many fields. In [1] classic step-by-step assembly instructions are drawn by automatically creating exploded views for each assembly step. This was motivated by the fact that a series of optimized instructions is easier to understand than a single complex exploded view diagram. To avoid incomprehensible images known from complex static diagrams, in [10] a semiautomatic approach for interactive exploded views has been developed that is based on 2D-explosion-images. This approach allows to reverse the depicted explosion and thus producing an adjustable dynamic explosion. Both approaches ([1, 10]) use semantical explosions as they need knowledge to integrate or disintegrate complex objects in correct order.

Similar approaches are found in the field of non-photorealistic rendering for the examination of 3D-models (e.g. in [18]). For example, in [13] a hybrid explosion is used to simultaneously explore interior and exterior structures of architectural environments by automatically separating the 3D-model into floors and dispersing them.

In the field of volume visualization exploded views are used to show internal structures by splitting and translating voxels or multiple enclosed iso-surfaces (see [12, 4]).

A first application of exploded views in the field of information visualization is the interaction technique *liquid browsing* [22]. An expansion lens is used to increase the space between the information objects and thus to geometrically explode them. However, up to now exploded views are not sufficiently considered in information visualization, although they are an integral part in other application scenarios. Therefore, in the following sections we demonstrate by the use of examples, that exploded views can also be useful in the context of information visualization to explain techniques and data.

3. Explode to explain visualization techniques

Applying new visualization techniques often requires high effort to understand the new images. Therefore, in many application domains the usage of traditional visualization techniques is often preferred compared to more complex ones.

Traditional techniques (like scatterplots or different diagrams) that can be found in current data processing software are used intuitively and widespread because they are known from the public media, from school education or they are simply self-explanatory. However, often they are not appropriate to visualize complex interrelationships, they generally require much space, and thus are mainly applicable to visualize small volumes of data.

In contrast to these well-known techniques, more complex visualization techniques are found in specialized data processing and visualization software. They are mainly applied by trained users and require intensive familiarization. Complex visualization techniques are mainly designed to explore large data volumes and complex interrelationships to get insight. Compared to traditional techniques, they generally require less screen space to show the same amount of data. However, the principal goal of visualization – to get insight from illegible data by graphically presenting the data by complex visualization techniques – may be hard to achieve for non-expert users, because these techniques are generally not self-explanatory and thus the resulting images are not easy to interpret.

Therefore, we want to support the application of complex techniques by explaining them. For this reason, we concentrate on presentation rather than exploration and suggest a temporal adaptation of visual representations by an explosion and the integration of explanatory elements: we propose the *illustration* of images gained from information visualization. Our general approach can be summarized as follows:

1. Select a region of interest in the visual representation that will be used as a key element to explain the technique.

2. Generate an appropriate explosion scheme based on the underlying technique.

3. Explode in a comprehensible way.

4. Superimpose explanatory elements.

That means, we explain the entire technique by a local extract, that has to be chosen sufficiently (step 1). Thereby the complexity is reduced. In case of explaining techniques, the steps 2 to 4 are user independent and given by the author of the visualization, whereas explaining data (Sect. 4) requires a specification of user interests.

We will demonstrate this approach with the help of two well-known examples from data visualization and structure visualization.

3.1. Pixel-oriented visualization

Pixel-oriented visualization techniques are characterized by mapping each data value onto the color of a single pixel, e.g. each pixel represents a value and thus the value distribution of a very large volume of data can be displayed (see [9]). However, the compact design provides no space for labeling or additional annotations (besides global labeling like headlines or tool-tip labeling). For an adequate understanding of pixel-oriented visualizations the underlying arrangement strategy of pixels has to be known, but this key factor is not comprehensible from the image by itself (see Fig. 1).

Therefore, our approach is to explode images gained from pixel-oriented visualization techniques in such a way that the progress of explosion as well as the additional explanatory elements communicate the arrangement strategy.

To generate exploded pixel-oriented visual representations we follow the procedure described in Section 3:

Selecting a region of interest. The selection of a specific region of interest (ROI) is necessary since an explosion of the whole image could be more confusing than explanatory (see [1]). This selection can be specified interactively or automatically. In the case of interactive specification, the user brushes a specific area of interest within the image to be exploded. To communicate the arrangement of pixels, a minimum size of the ROI should be ensured to achieve this aim. Therefore, the ROI is automatically adjusted, replaced and resized to fulfill this requirement, e.g. for the recursive pattern technique the ROI is adapted in such a way that it corresponds to one recursive pattern.

In the case of automatic specification, an appropriate ROI is selected based on computations. Which algorithm has to be carried out for this purpose, is application- and user-dependent. For example, if a user is interested in extreme values, the area that contains these values will be selected automatically.

Generating an explosion. To generate an explosion, first the explosion process has to be specified as geometrical, semantical or hybrid. This decision depends on the underlying visualization technique.

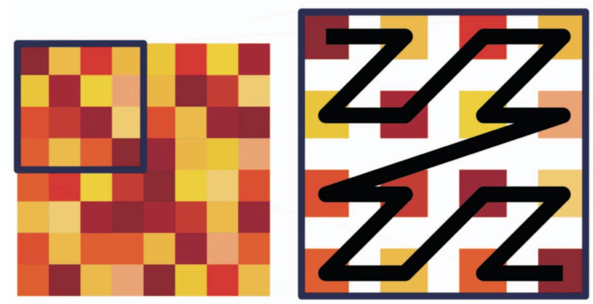


Figure 1. This figure shows the explanation of the Z-curve arrangement strategy for 64 pixels. By exploding the selected region followed by a magnification and superimposing of lines in arrangement order, this strategy becomes comprehensible.

In any case geometrical explosions provide additional space between data values represented by single pixels. This way, data values are separated sufficiently to guarantee an appropriate illustration. Moreover, a scaling is integrated to represent data values by a group of pixels rather than a single pixel (Fig. 1).

Semantical explosions use knowledge about the underlying arrangement strategy to demonstrate the visualization technique. For example, in the recursive pattern technique the recursion is used for explosion generation, since this recursion is the key element to understand the arrangement strategy. In Figure 2 multiple unidirectional explosions are stacked according to the recursion depth, enabling a separation of the ROI recursion-level-by-recursion-level.

Note that the center of explosion should be placed in a way that guarantees the visibility of the whole exploded ROI. Moreover, the exploded view is generated on the selected ROI to explain the applied visualization technique. That means, that the remaining visual representation outside the ROI is slightly suppressed with a semitransparent layer to detach explanation from that representation. Thus the explosion is done above that representation at this so called illustration layer.

Explode in a comprehensible way. Justified by [1] and [10], a controllable dynamic explosion should guarantee comprehensibility. This is done by stepwise incrementing the explosion level(s) up to its final value, shown in an animation. In case of the recursive pattern technique several so called sub-explosions are integrated by an explosion of a higher recursion level. Therefore, first the explosion level of the latter explosion has to be increased to gain space for the following sub-explosions of lower recursion levels.

Figure 2. Explaining the recursive pattern technique: The series shows the stepwise separation of the recursion levels of one pattern from left to right. Additional annotations explain the semantic of the recursion. In this figure one recursive pattern represents the data of one year that is separated into its quarters, months, weeks and days.

Superimposing explanations. The final explanation with visual elements like annotations, lines, arrows etc. depends heavily on the suitability of these elements concerning the explanation of the corresponding technique and the user group.

In Figure 1 we propose superimposed lines, since the space-filling curves used here, in general are illustrated this way. Alternatively, a numbering of the elements of the ROI showing the arrangement order may also be applicable.

Figure 2 shows exploded views for the recursive pattern technique. The stepwise explosion demonstrates the recursion levels, whereas additional annotations explain the semantic of the recursion.

3.2. Hierarchy visualization

After discussing an exploded view approach for data representation, our second example demonstrates exploded structure visualization. We focus on hierarchy visualization techniques that can be classified into two groups (see [16]): *explicit* and *implicit* techniques. *Explicit* techniques draw *node-and-link* diagrams, whereby the edges of a graph are explicitly represented by lines. *Implicit* techniques communicate the structure implicitly by nesting visual representatives of nodes, or by visualizing neighborhood dependencies of those. Since explicit diagrams require additional screen space to display the edges, the space-filling implicit techniques are generally able to show larger hierarchies within the same screen area. However, the visual output of implicit techniques may be hard to understand especially for non-expert users, because the underlying principle of nesting or subdivision is unknown.

Therefore, we explain the underlying nesting by the use of exploded views. As a popular example of implicit techniques we have focused on treemaps (see [17]). This technique has already been exploded in 2D showing the underlying nesting (see [11]). Since this approach does not provide as much space as necessary for more comprehensive annotations, we suggest a hybrid explosion into the 3^{rd} dimension. This approach is applicable to treemaps as well as to most other implicit 2D-techniques (like [19, 11, 2]) and some 3D-techniques (like [3, 21]).

Selecting a region of interest. As discussed in Section 3.1, a ROI – in this case a subtree of the hierarchy – can be selected interactively or automatically. In the interactive case the user brushes a region in the image. Since a treemap represents nodes of different hierarchy levels at the same screen position, a strategy has to be specified to snap to the correct subtree within this area. A straight forward approach is choosing the node in the uppermost hierarchy level that is completely inside the brushed area to define the ROI. However, it is also possible (depending on the given application scenario) to choose a node at another hierarchy level to define the ROI.

In the case of an automatic selection, an appropriate ROI has to be computed. This computation is application- and user-dependent. For example, if a user is interested in highly structured subtrees, the corresponding nodes will automatically be selected to define the ROI.

Generating an explosion. In our hybrid explosion approach the hierarchy level of the tree nodes is mapped onto a depth value thus enabling the explosion into the 3^{rd} di-

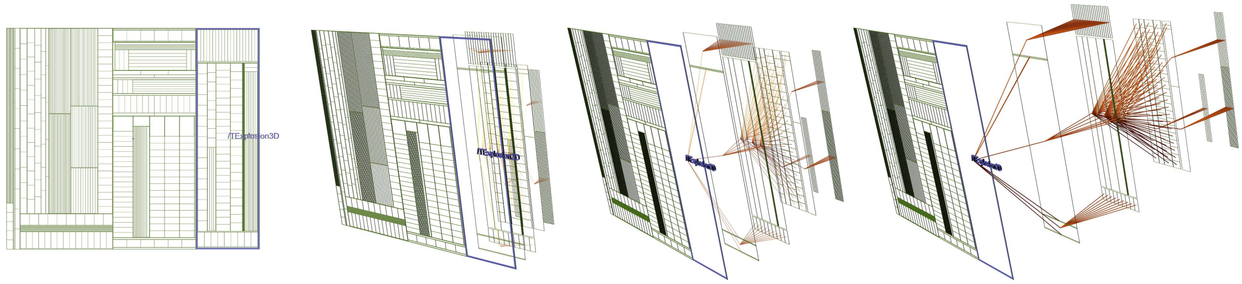


Figure 3. Explaining a treemap: The series shows the view-dependent explosion into depth and the superimposing of the underlying hierarchy. This illustration is applicable to most implicit 2D-techniques.

mension. This depth value is used to calculate a geometrical distance that can be manipulated during the explosion process. The root node of the ROI is the center of explosion thus all visual representatives of lower hierarchy levels are moved with an unidirectional explosion into depth. An omnidirectional explosion may be used to gain more space within each "hierarchy-layer", e.g. for labeling.

Explode in a comprehensible way. Since the explosion is executed into depth, a slight rotation of the view is necessary to reveal the animated explosion (see Fig. 3).

Superimposing explanatory elements. In our example, we focus on the explanation of the used nesting. Therefore, we show the underlying structure *explicitly* by superimposing the connecting lines among the visual representatives of nodes.

4. Explode to explain data

In Section 3 we have discussed the general procedure of generating an exploded view to explain techniques like pixelmaps or treemaps. In this section we examine, how exploded views can further support the understanding of underlying data, by generating enhanced views on that data.

Due to the high density of represented information by using complex visualization techniques, it is often difficult to locate specific data items of interest. However, the localization of these specific data items can be a key factor to interpreting the image and to understanding the underlying data. Exploded views can also be applied to support the understanding of data. In this case the data is explained rather than the visualization technique.

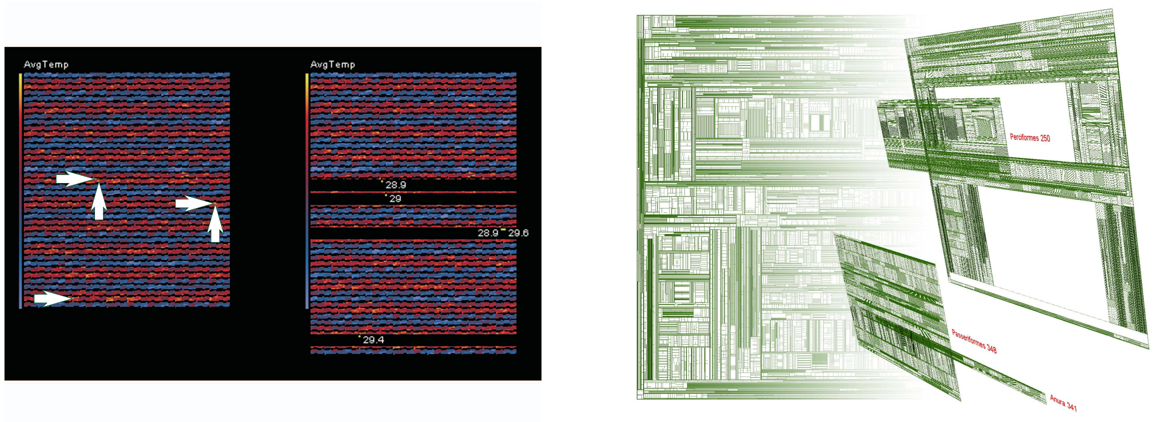
For this purpose the explosion process has to be enhanced by the following aspects:

- Choosing ROIs based on data values of interest.
- Accentuating the data of interest to support localization.
- Annotation to support the understanding of depicted data.

To explain data, the *ROI-selection* is based on an automatic analysis of data characteristics, rather than on interactive input. A given user profile or an interactive specification of user interests, e.g. detecting extreme values, determines the ROI to be exploded. This automatic ROI-selection may base on any computation that provides a subset of the originally shown data (e.g. filtering, clustering...). Nevertheless, an interactive ROI selection should still be possible to brush additional regions to be explained in more detail by exploding them.

Since the explosion process described in Section 3 guarantees a comprehensible procedure and supports a better recognition of the ROIs, step 2 and 3 of our general approach are not affected. However, for an appropriate labeling of specific data items of interest, the explosion generation has to provide enough screen space in these regions to be annotated.

The last step, *superimposing explanatory elements*, is adapted in two ways. First, those visual primitives representing the most important values are accentuated, e.g. by a magnification or annotating arrows. Secondly, further annotations can be added to explain the data, e.g. by labeling the associated graphical primitives. This annotation may also include a replacement of visual primitives by more explanatory representatives to show the data in more detail, e.g. a single dot is replaced by a small icon representing additional information. In the following, we will discuss this approach, using the same examples as in Section 3.



(a)

(b)

Figure 4. Accentuation and labeling of interesting data items by applying exploded views: a) Within a recursive pattern technique, the five highest values (arrows, left) are accentuated and labeled (right). b) The three most complex biological orders within the phylum of vertebrates – represented by a treemap – are accentuated. Furthermore they are labeled with their Latin name and their complexity value (Horton-Strahler number).

4.1. Pixel-oriented visualization

To support the recognition of interesting data, first the user has to specify his interests. Basing on this specification, appropriate automatic computations are carried out (see [20]). The result are *automatically derived ROIs* that are separated by an explosion. That means the visual representation is appropriately split either horizontally or vertically at each element of the ROIs. The image parts defined that way are exploded and thus gaps between these parts are generated (see Fig. 4a).

For an *accentuation*, only the elements of the ROI are magnified and placed inside these attention attracting gaps. Furthermore, this approach is used to gain space for the *labeling* of those specific elements of interest. For example, in Figure 4a extreme values of a climate dataset are labeled (daily data of 100 years $\approx$ 36500 data items). Note that labels generally have a larger width than height and a vertical explosion should therefore be preferred.

4.2. Hierarchy visualization

First a *ROI* has to be *automatically determined* that includes the subtree of interest. To accentuate or label this ROI, current approaches can be used in principle (see [15, 5, 19]). However, in general these approaches manipulate the surrounding context. Therefore, our approach is to *accentuate* the ROI by exploding it to the 3^{rd} dimension. Instead of the hierarchy-level (see Sect. 3.2), a specific depth value for each ROI is used to separate the ROI from the context. This way, an orthogonal view shows the traditional treemap representation, whereas a slight rotation accentuates the separated ROIs without manipulating the context. Figure 4b demonstrates this approach applied to a biological classification dataset ($\approx$ 36000 nodes - part of [6]). In our example the user is interested in nodes associated with a high complexity value (in our case approximated by Horton-Strahler numbers, see [7]). Subtrees which are defined by such complex nodes, are marked as ROIs and explode in depth, whereas the depth values are based on the complexity values. Inspired by [5], we recommend a *labeling* at the different depth layers outside the visual representatives, to guarantee an adequate annotation for different ROI shapes. Which nodes within a ROI have to be labeled to explain data is application- and user dependent. For example, in Figure 4b only the root nodes within the ROI – representing the complex biological orders – are labeled.

5. Discussion

Explode to explain is *not* designed to replace any complex visualization technique, but to enhance them. The proposed exploded views for information visualization are a very first attempt to *explain visualization techniques*. This is done by demonstrating the generation of visual representations through a dynamic explosion procedure. This explanation is based on a region of interest of the image of sufficient size to guarantee comprehensibility.

Furthermore, we use exploded views to improve the communication of visually represented information and thus, to support the *explanation of data*. Thereby, the original view of the visualization technique is still given. Thus, the application of exploded views is an *additional* feature to call attention to specific areas of the representation and to supply additional information by annotations. For example, Figure 4a communicates the concrete values of extreme temperatures at a first glance.

However, our approach requires sufficient screen space to explain the mentioned subareas of the image by an explosion. Therefore, the huge volume of data that is shown is reduced to the data items of interest drawn within a ROI. To explain visualization techniques, this limitation on a specific ROI is sufficient, and to focus on data of interest, it is also sufficient to explode specific subareas.

This work was developed within the graduate school 466 at the institute of computer science at the University of Rostock – integrating researchers from different disciplines like electrical engineering, law and economics as non-expert users. First tests with participants of this graduate school have shown that our approach of exploded views significantly supports understanding of complex visualization techniques, increasing the willingness of using them. Their positive feedback encourages our intent in further investigation of exploded views in information visualization.

This will include the extension of automatic computations and an enhanced specification of user interests. Furthermore a user study is under development to quantify the qualitative feedback from the PhD students of our graduate school.

References

[1] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, and B. Tversky. Designing Effective Step-By-Step Assembly Instructions. *SIGGRAPH2003*, 22(3):828–837, 2003.

[2] M. Balzer, O. Deussen, and C. Lewerentz. Voronoi treemaps for the visualization of software metrics. In *Proceedings of the 2005 ACM symposium on Software visualization (SoftVis'05)*, pages 165–172, 2005.

[3] T. Bladh, D. A. Carr, and J. Scholl. Extending tree-maps to three dimensions: a comparative study. In *Proceedings of the 6th Asia-Pacific Conference on Computer-Human Interaction (APCHI'04)*, pages 50–59, 2004.

[4] S. Bruckner and M. E. Gröller. Exploded views for volume data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1077–1084, 2006.

[5] J.-D. Fekete and C. Plaisant. Interactive information visualization of a million items. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 117–124, 2002.

[6] J.-D. Fekete and C. Plaisant. Visualization and pair wise comparison of trees. IEEE InfoVis 2003 Contest, www.cs.umd.edu/hcil/iv03contest, 2003.

[7] I. Herman, M. Delest, and G. Melancon. Tree visualisation and navigation clues for information visualisation. *Computer Graphics Forum*, 17(2):153–165, 1998.

[8] S. Islam, S. Dipankar, D. Silver, and M. Chen. Spatial and temporal splitting of scalar fields in volume graphics. In *IEEE Symposium on Volume Visualization and Graphics (VV'04)*, pages 87–94, 2004.

[9] D. A. Keim. Pixel-oriented visualization techniques for exploring very large databases. *Journal of Computational and Graphical Statistics*, 5(1):58–77, 1996.

[10] W. Li, M. Agrawala, and D. Salesin. Interactive image-based exploded view diagrams. In *GI '04: Proceedings of the 2004 conference on Graphics interface*, pages 203–212, 2004.

[11] S. Liu, N. Cao, H. Lv, and H. Su. The visual funding navigator: analysis of the nsf funding information. In *Proceedings of the 15th ACM international conference on Information and knowledge management (CIKM '06)*, pages 882–883, 2006.

[12] M. J. McGuffin, L. Tancau, and R. Balakrishnan. Using deformations for browsing volumetric data. In *Proceedings of IEEE Visualization (VIS) 2003*, pages 401–408, 2003.

[13] C. Niederauer, M. Houston, M. Agrawala, and G. Humphreys. Non-invasive interactive visualization of dynamic architectural environments. In *Proceedings of the 2003 Symposium on Interactive 3D Graphics (SI3D '03)*, pages 55–58, 2003.

[14] G. Poetter. *Die Anleitung zur Anleitung : Leitfaden zur Erstellung technischer Dokumentationen*. Vogel Buchverlag, 1994.

[15] S. Schlechtweg, P. Schulze-Wollgast, and H. Schumann. Interactive treemaps with detail on demand to support information search in documents. In *Data Visualization 2004. Eurographics/IEEE TCVG Visualization Symposium Proceedings*, pages 121–128, 2004.

[16] H.-J. Schulz and H. Schumann. Visualizing graphs - a generalized view. In *Proceedings of the conference on Information Visualization (IV'06)*, pages 166–173, 2006.

[17] B. Shneiderman. Tree visualization with tree-maps: A 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[18] H. Sonnet, S. Carpendale, and T. Strothotte. Integrating expanding annotations with a 3d explosion probe. In *Proceedings of the working conference on Advanced Visual Interfaces (AVI '04)*, pages 63–70, 2004.

[19] J. Stasko and E. Zhang. Focus+context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Proceedings of the IEEE Symposium on Information Vizualization (InfoVis'00)*, pages 57–65, 2000.

[20] C. Tominski. *Event-Based Visualization for User-Centered Visual Analysis*. PhD thesis, University of Rostock, Germany, 2006.

[21] F. van Ham and J. J. van Wijk. Beamtrees: Compact visualization of large hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 31–39, 2002.

[22] C. Waldeck and D. Balfanz. Mobile liquid 2d scatter space (ml2dss). In *Proceedings of the eighth international conference on Information Visualisation (IV'04)*, pages 494–498, 2004.