Discovering the Covered: Ghost-Views in Information Visualization

Martin Luboschik, Heidrun Schumann
Institute of Computer Science, University of Rostock, Albert-Einstein-Str. 21, 18051 Rostock, Germany
{luboschik,schumann}@informatik.uni-rostock.de

ABSTRACT

A not negligible number of information visualization techniques uses 3D-geometry to visualize data and structures. Thereby, constantly growing data volumes influence the final visual representation and often result in the occlusion of certain items. Therefore, different approaches have been developed that mainly manipulate item positions to uncover specific items of interest or otherwise use filtering and information hiding to reduce the amount of visible items. This paper presents a novel method to adapt 3D-views from information visualization by the use of the well-known illustrative technique *ghost-view* to successfully address this occlusion problem. Applying ghost-views to 3D information visualization techniques ensures the visibility of selected items by view-dependently manipulating the transparency of unselected data: without any manipulation of positions or continuous context suppression. Our approach is applicable to most 3D visualization techniques. It is interactive and easy to adapt to existing visualization environments.

Keywords: Information visualization, illustrative rendering, ghost-views.

1 INTRODUCTION

In various fields of information visualization, 3D-visualization techniques have been developed to visualize information structures (e.g. in [RG93] [vHvW02] [BCS04]...) and information objects (e.g. [MRC91] [TK98] [TSWS05] ...).

Since the visualization of constantly growing large data volumes is a general aim of information visualization, this becomes a problem for 3D visualization techniques due to the growing probability of occlusions. Card et. al. points out that occlusions are a major problem in 3D information visualization (see [CMS99]).

In the field of information visualization, existing solutions to solve the occlusion problem mainly propagate two strategies: First a view dependent arrangement / distortion of 3D graphic primitives (e.g. in [SCCF96]) is applied that reduces the complexity in presentation space and shows objects that are occluded otherwise. The second approach, called information hiding, reduces complexity in information space resulting in a continuous reduction of visible representatives (see [Fur81]). However, the first approach is not applicable in scenarios where the graphic primitives position is necessary for interpretation of the information. The second approach will fail if the context has to be known to understand the visualized data.

However, in the field of technical and medical illustrations ghost-views are used to facilitate a view on objects of interest that are encapsulated by 3D geometry. These views have been successfully adapted to the field of volume visualization and illustrative rendering of 3D models but are rarely considered in 3D information visualization (Section 2).

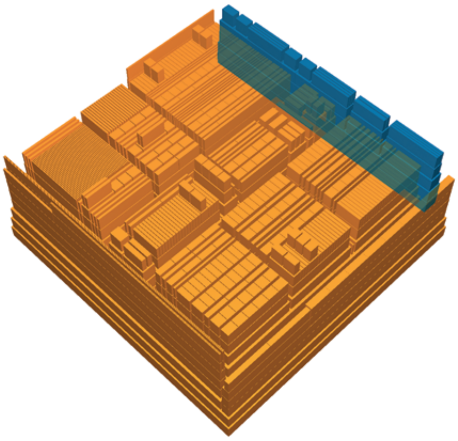

Figure 1: This figure shows the application of the proposed ghost-view technique to a 3D hiearchy visualization called steptree (see [BCS04]).

We propose the application of ghost-views to 3D information visualization techniques to guarantee the visibility of selected items without arrangement manipulation or continuous context suppression (Section 3). Our general approach that takes advance of modern graphics hardware (Section 5), is applicable to several 3D information visualization techniques – demonstrated by two examples (Section 4). The results of our approach are discussed in Section 6.

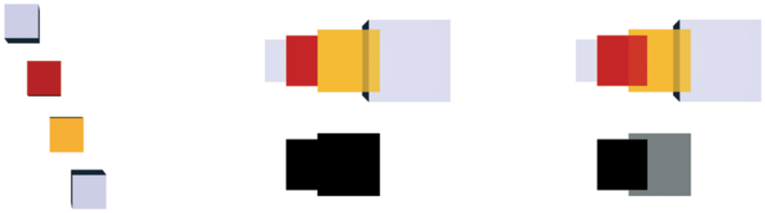

Figure 2: The figure shows a small setup of four boxes, whereby the light blue are unimportant objects and the orange / red boxes are objects of interest (left, shown from top). In the center, the conflict between the objects of interest is solved using standard depth sorting. At the right, gradual differences between the objects of interest ensure the visibility of the red box (importance maps are shown below).

2 BACKGROUND

A ghost-view is a specific illustration technique that is used to make objects visible that are enclosed by 3D geometry [Die05]. The transparency of the occluding geometry is manipulated locally depending on the current view point to guarantee the visibility of the objects of interest in context of the whole representation. Whereas a cutaway-view hides occluding geometry parts completely, a ghost-view maintains some information about the occluding objects shape (e.g. using semi-transparency, silhouettes or other visual clues).

Both, ghost-views and cutaway-views have been successfully applied in different fields and their automatic generation is an ongoing research topic. In [DWE03] [LRA$^+$07] for example, such illustrative views are generated for complex 3D models and in [HBP$^+$07] [Die05] the output on mobile devices is additionally addressed.

Moreover, ghost-views are also used in the field of volume visualization to make encapsulated volume data visible (e.g. in [BGKG05] [VKG05] [TIP05] [RBG07] [BG07]). In case of such interactive explorative scenarios ghost-views and cutaway-views are mostly calculated view-dependently to guarantee the visibility of objects of interest (e.g. a tumor) for every view point.

The recent publications demonstrate the relevance of those views to illustrative rendering and volume visualization.

In 3D information visualization the same problem exists since objects of interests may not be visible due to occlusion. After complex parametrization steps one can ensure those objects to be visible but even small interactions during the exploration process can make them disappear again. The occlusion will even lead to misinterpretation if the object of interest is the result of a user defined query. A hidden object may be interpreted as an absence of this object. This has to be avoided, since in different application scenarios the existence of specific objects is of high relevance: In a 3D network scenarios for example an important node of interest has to be visible [BTB$^+$99] or a node holding a special kind of data within a hierarchy [TON03]. Moreover, single stocks in a 3D stock market visualization [Wri95] or specific regions in a scenario with spatial data [TSWS05] could by of high interest and should therefore be visible.

In principle, there are two very successful approaches that address this 3D occlusion problem: Information hiding techniques (e.g., in [Fur81] [TSWS05]) and distortion techniques (e.g., in [SCCF96]). Information hiding techniques reduce complexity in the information space to limit the number of opaque visual representatives. The approach of filter fisheye views [Fur81] for example, reduces – dependent of the currently selected object of interest – the number of visible objects by calculating a degree of interest for the surrounding objects. Techniques like this decrease the probability of occlusions for selected objects in a 3D visualization but they are not able to guarantee the visibility. The named approach for example, retains objects with a high degree of interest. These are often visualized close to the selected object and may occlude it (semantic affinity often means spatial affinity). Furthermore, information hiding techniques are hard to apply if the suppressed context is necessary for a meaningful interpretation of the objects of interest (e.g., for comparison).

Solutions that base on view-dependent distortions or arrangements of 3D geometry, operate in presentation space to avoid the occlusion. In [SCCF96] for example, the occluded object of interest and the view point is used to define different scalings and distortions that are applied to the objects of the visualization. The result is a free view at the object of interest. Although this technique shows the whole 3D representation, the visualization may be difficult to interpret if the position of an object of interest itself is an important information in the exploration process.

Besides these principle strategies, there are also technique-specific solutions like the predefined and view-independent use of transparency. In [RG93] [TON03] for example, every enclosing geometry is transparent to enable navigation and interpretation of the visualized hierarchy structure. However, this approach does not guarantee the visibility neither of every visualized nor a single selected hierarchy node.

Although ghost-views are an integral rendering technique in other 3D visualization and rendering scenarios, they are not considered in 3D information visualization yet. Therefore, in the following sections we introduce a simple – visualization independent – approach to generate interactive ghost-views in 3D information visualization and demonstrate this approach with two short examples.

3 A GENERAL APPROACH FOR GHOST-VIEWS IN INFORMATION VISUALIZATION

Inspired by the idea of importance driven rendering presented in [VKG05], which handles volumes of different importance to generate ghost-views and cutaway-views, our approach is based upon a so called *importance map* and a two-pass-rendering. This approach can be summarized as follows:

1. Generate the importance map view-dependently by rendering the objects of interest into a (0/1)-texture.
2. Render all objects using the importance map as a look-up table to determine, whether they are occluding interesting objects or not. In case of occlusion, their transparency is adapted locally.

Therewith, our ghost-view approach works in information and in presentation space: In information space the objects of interest are defined as key elements in the later ghost-view generation. In the presentation space the visual representatives of these elements are used for transparency adaption of occluding geometry.

During the rendering step also the objects of interest are rendered using the importance map to handle the conflict that will appear if two or more objects of interest share their position in the importance map. For this case we propose two solutions:

- Using automatically applied depth sorting always shows the object of interest in front occluding the others.
- Extending the former (0/1)-importance map to multiple importance levels and allowing gradual differences between those objects of interest. In this way, objects of interest that are not of maximum importance at that position are also adapted in their transparency according to their importance (see Fig. 2).

Since this approach manipulates transparencies of occluding geometry, it can be used for both the generation of ghost-views (semitransparent) and cutaway-views (100%-transparency).

In Section 2 we showed that the position of visual representatives may be crucial for interpretation. Therefore this position should be clear to perceive. Since a cutaway-view may bring the objects of interest visually in front of all other geometry, this perception – especially of depth – may be disordered (see Fig. 3). Therefore, we propagate the use of ghost-views with semitransparent occluding geometries to maintain correct position interpretation.

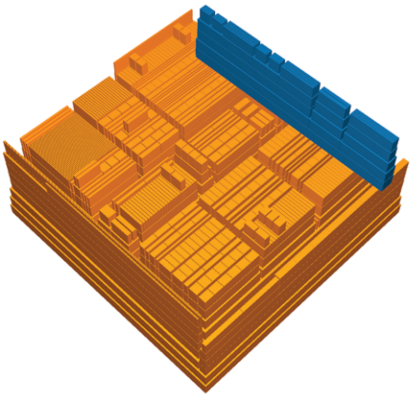

Figure 3: The use of 100% transparency (cutaway-view) without other visual clues disorders depth perception and may lead to misinterpretation (compare to Fig. 1).

The application of ghost-views in different fields created also different approaches for their generation. The commonly used approach in volume visualization is a direct manipulation of voxel parameters (e.g. in [VKG05]). Thereby, the *level of sparseness* is mapped onto the transparency of a voxel for example. The rendering step is generally unmodified. Moreover, there are several approaches for ghost-view creation in technical illustrations. In [Die05] different solutions – reaching from simple CSG operations to complex multipass-rendering steps realizing a depth sorting – are demonstrated. A sorting of geometry is usually necessary to achieve *correct* transparencies that are needed in illustrative rendering.

Although our straight-forward approach does not include a view dependent sorting of geometry, it is sufficient to guarantee the visibility of objects of interest in a 3D information visualization.

4 APPLICATION EXAMPLES

In the following sections we will demonstrate our approach with two different examples from 3D information visualization. The first is a structure visualizing technique called treecube (see [TON03]) that is used for administration of 3D multimedia data. The second example comes from the field of information object visualization and is based upon a spring-model (see [TK98]).

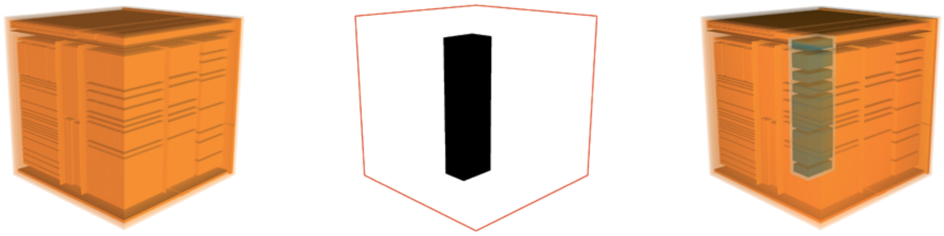

Figure 4: A family of interest (Corellidae) within a phylogenetic dataset is selected and unfortunately hidden in a standard treecube visualization (left). Using an importance map (center) in combination with a GPU-program generates a ghost-view and ensures the objects visibility (right). The red contour in the importance map is only shown for orientation purposes.

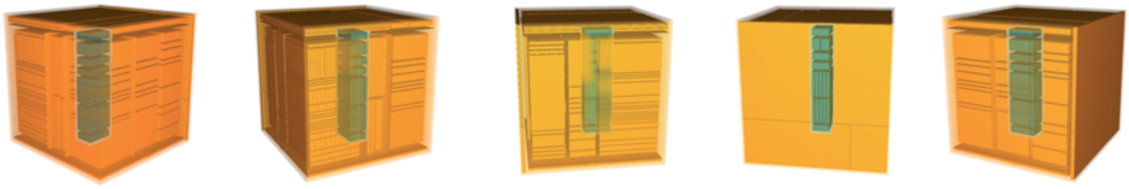

Figure 5: This series shows the view-dependent generation of ghost-views that ensures visibility within this treecube visualization.

4.1 Treecube

A treecube is a 3D extension of the well known hierarchy visualization technique called treemap [Shn92]. A cube is divided along the x,y and z-axis according to the underlying hierarchy (see [TON03] for details). Although the separation planes are semitransparent, it is not guaranteed that every node of the shown hierarchy is visible. Especially nodes that are represented by boxes in the deep interior of the cube are endangered of occlusion.

In [TON03] the treecube holds 3D multimedia data like 3D geometry models. In this case, an object of interest may be a hierarchy node that holds one kind of models (e.g. airplanes) as these models are currently used in a modelling scenario. Moreover, the structural relationship to the surrounding hierarchy is necessary for fast navigation. Therefore, that node and the surrounding structures should be visible for further use or investigation.

We demonstrate the application of ghost-views to the treecube visualization technique by using a small phylogenetic dataset ($\approx$ 580 nodes, part of [FP03]) and one selected subtree of interest (family Corellidae). Even in case of such few visualized hierarchy nodes, the selected subtree is usually not visible (Fig. 4 left).

As described in Section 3 we generate a view-dependent importance map using the selected subtree (Fig. 4, center). Afterwards, this importance map is used as a look-up table in the final rendering step for transparency adaption (Fig. 4, right).

Therewith, the occluded subtree becomes visible for every view point without a continuous context suppression (Fig. 5). This enables an interactive investigation and shows the subtrees context concerning the outer geometry. Nevertheless, other interior structures or the near context of the selected subtree are not visible.

4.2 Spring-model

To demonstrate that ghost-views are also applicable to non-encapsulating 3D visualization techniques we use a simple example, showing numerous information objects as spheres in 3D space. The objects positions are given by their attributes that influence spring-forces towards attracting points distributed on a surrounding sphere (see [TK98]). These techniques are generally used to see the distribution of the whole information space according to its attributes or the development of information objects. Moreover, it enables the comparison of single objects towards the whole volume. Therewith, this technique is generally applied to multivariate data sets (like quality of living parameters of US cities in [TK98]).

In our example we use this technique for the visualization of health data of $\approx$ 230 districts of the federal state Mecklenburg-Vorpommern, Germany in January 2000. Ten different attributes recorded by a health insurance were used to place the visual representatives of the districts. These attributes include the number of influenza cases, the cases of gastro-intestinal diseases, cases of arthropathy and more. Additionally an object holding aggregated data of all districts is visualized to

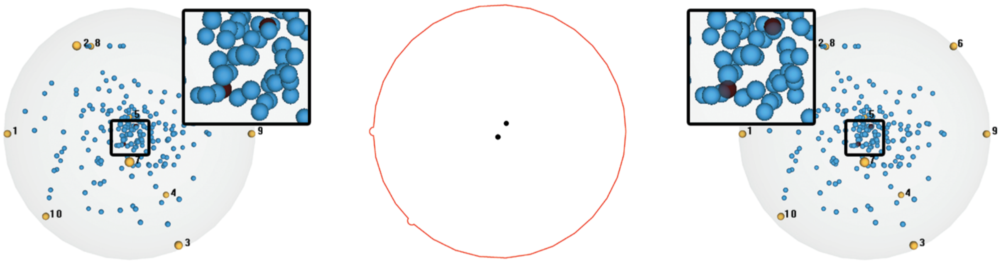

Figure 6: A district of the federal state Mecklenburg-Vorpommern, Germany is compared to an object holding aggregated data of the whole state. Their visual representatives (dark spheres) are difficult to locate in a spring-model visualization (see magnification left). Using the importance map (center) during rendering makes them visible (right). The red contour (center) is only shown for orientation purposes.

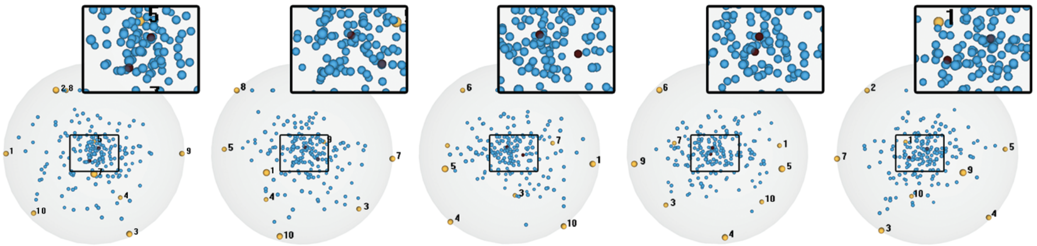

Figure 7: This series shows the view-dependent generation of ghost-views to ensure visibility within a spring model visualization.

show the average. In Figure 6 one of those districts is selected for comparison to this aggregated value (also selected).

Although a relative small data set is visualized, the selected objects are hard to see. To guarantee their visibility, we use our approach of Section 3: Both objects are used to generate the importance map that is afterwards used in the final rendering of the whole data set (see Fig. 6).

Thus, the occluded objects of interest become visible for every view point without unnecessary context suppression or distortions (Fig. 7). Semitransparent occluding objects ensure the depth and shape perception. Moreover, the whole context of the selected objects is visible due to a local transparency adaption and a missing encapsulation within this visualization technique.

5 IMPLEMENTATION

Our approach of importance map based ghost-view generation for 3D information visualization uses GPU programming facilities of recent graphics hardware to keep this approach interactive and applicable to different visualization techniques. The shown examples were implemented prototypal in Java 1.6 using JOGL (JSR-231 1.1) and GLSL (all non-optimized). A fragment shader is used to modify the transparency of the occluding geometry on a per pixel basis. The importance of the currently rendered object is passed to the shader which compares it to the importance map and adapts the pixels transparency if necessary.

The examples run on a Intel Pentium 4, 3.2GHz, 1GB RAM and NVIDIA Quadro NVS 280 PCI-E graphics card at framerates of 11 to 43 fps. This demonstrates that our method works at interactive framerates even on desktop computers that are not optimized for 3D graphics. To additionally speed up the visualization, the importance map can be rendered in lower resolutions resulting in a fringed border between the opaque and transparent parts of an occluding geometry.

The anchors in Figure 6,7 are currently not involved in the ghost-view generation but may at any time. An integration of the simple 2D labeling within our implementation is currently not possible, due to a separate rendering processing. A revised implementation should facilitate this.

6 CONCLUSION

Our introduced approach of applying ghost-views to 3D information visualizations ensures the visibility of important objects in every step of the visual analysis. Hence, it supports the exploration of large data volumes and represents an additional alternative to distortion and information hiding techniques.

The major benefit of our technique is that objects of interest can be perceived at first glance without complex parametrization steps. This visibility is guaranteed during the interactive exploration process.

Furthermore, the use of modern graphics hardware keeps the ghost-view generation interactive and independent of any visualization technique or data structure. Therewith, it is also applicable to other 3D visualization scenarios than the ones described in Section 4. A so called *importance map* is used to determine whether the opacity of an occluding geometry has to be changed. The generation of that importance map is easy to implement and needs only one additional rendering pass. Since generally a lower number of polygons has to be rendered in this generation step, the maximum rendering time is twice of a standard rendering. The map ensures a locally bounded manipulation of transparencies and therewith supports the perception of context information.

Although our approach guarantees visibility without distortions, it reveals only objects of interest. In 3D visualization techniques that produce enclosures, the *near* context of the interesting object is still hidden and may hinder an interpretation. Distortion techniques are usually able to show that context. Another problem arises through the use of semi-transparencies: If attributes of the interesting objects are mapped onto its color, this color changes view-dependently due to different occlusion scenarios (see Fig. 5). This may lead to misinterpretations.

Therefore, further investigations will examine the use of other visual clues than transparency to retain depth and shape perception. Moreover, a combination with information hiding / distortion techniques could solve the problem of hidden near context.

REFERENCES

[BCS04] Thomas Bladh, David A. Carr, and Jeremiah Scholl. Extending tree-maps to three dimensions: A comparative study. In *Proceedings of the Asia-Pacific Conference on Computer-Human Interaction (APCHI'04)*, pages 50–59, 2004.

[BG07] Stefan Bruckner and Eduard Gröller. Style transfer functions for illustrative volume rendering. *Computer Graphics Forum*, 26(3):715–724, 2007.

[BGKG05] Stefan Bruckner, Sören Grimm, Armin Kanitsar, and Eduard Gröller. Illustrative context-preserving volume rendering. In *Proceedings of EUROGRAPHICS - IEEE VGTC Symposium on Visualization (EuroVis'05)*, pages 69–76, 2005.

[BTB$^+$99] Steve Benford, Ian Taylor, David Brailsford, Boriana Koleva, Mike Craven, Mike Fraser, Gail Reynard, and Chris Greenhalgh. Three dimensional visualization of the world wide web. *ACM Computing Surveys*, 31(4):25, 1999.

[CMS99] Stuard Card, Jock Mackinlay, and Ben Shneiderman. *Readings in Information Visualization – Using Vision To Think*, chapter 2.2, pages 57–61. Morgan Kaufmann, 1999.

[Die05] Joachim Diepstraten. *Interactive Visualization Methods for Mobile Device Applications*. PhD thesis, University of Stuttgart, Germany, 2005.

[DWE03] Joachim Diepstraten, Daniel Weiskopf, and Thomas Ertl. Interactive cutaway illustrations. *Computer Graphics Forum*, 22(3):523–532, 2003.

[FP03] Jean-Daniel Fekete and Catherine Plaisant. Visualization and Pair Wise Comparison of Trees, IEEE InfoVis 2003 Contest, 2003.

[Fur81] George W. Furnas. The FISHEYE view: A new look at structured files. Technical Report #81-11221-9, Bell Laboratories, 1981.

[HBP$^+$07] Jingshu Huang, Brian Bue, Avin Pattath, David S. Ebert, and Krystal M. Thomas. Interactive illustrative rendering on mobile devices. *IEEE Computer Graphics and Applications*, 27(3):48–56, 2007.

[LRA$^+$07] Wilmot Li, Lincoln Ritter, Maneesh Agrawala, Brian Curless, and David Salesin. Interactive cutaway illustrations of complex 3d models. *ACM Transactions on Graphics*, 26(3):31, 2007.

[MRC91] Jock D. Mackinlay, George G. Robertson, and Stuart K. Card. The perspective wall: Detail and context smoothly integrated. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'91)*, pages 173–179, 1991.

[RBG07] Peter Rautek, Stefan Bruckner, and Eduard Gröller. Semantics driven illustrative rendering. Technical report, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2007.

[RG93] Jun Rekimoto and Mark Green. The information cube: Using transparency in 3d information visualization. In *Proceedings of the Third Annual Workshop on Information Technologies & Systems (WITS'93)*, pages 125–132, 1993.

[SCCF96] M. Sheelagh, T. Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Distortion viewing techniques for 3-dimensional data. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'96)*, pages 46–53, 1996.

[Shn92] Ben Shneiderman. Tree visualization with tree-maps: A 2-d space-filling approach. *ACM Transactions on Graphics*, 11(1):92–99, 1992.

[TIP05] Christian Tietjen, Tobias Isenberg, and Bernhard Preim. Combining silhouettes, shading, and volume rendering for surgery education and planning. In *Proceedings of the Eurographics / IEEE VGTC Symposium on Visualization (EuroVis'05)*, pages 303–310, 2005.

[TK98] Holger Theisel and Matthias Kreuseler. An enhanced spring model for information visualization. *Computer Graphics Forum*, 17(3):335–344, 1998.

[TON03] Yoichi Tanaka, Yoshihiro Okada, and Koichi Niijima. Treecube: Visualization tool for browsing 3d multimedia data. In *Proceedings of the 7th International Conference on Information Visualization (IV'03)*, pages 427–432, 2003.

[TSWS05] Christian Tominski, Petra Schulze-Wollgast, and Heidrun Schumann. 3d information visualization for time dependent data on maps. In *Proceedings of the 9th International Conference on Information Visualisation (IV'05)*, pages 175–181, 2005.

[vHvW02] Frank van Ham and Jarke J. van Wijk. Beamtrees: Compact visualization of large hierarchies. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, pages 31–39, 2002.

[VKG05] Ivan Viola, Armin Kanitsar, and Eduard Gröller. Importance-driven feature enhancement in volume visualization. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):408–418, 2005.

[Wri95] William Wright. Research report: Information animation applications in the capital markets. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'95)*, pages 19–25, 1995.