Coherent Zooming of Illustrations with 3D-Graphics and Text

Bernhard Preim, Andreas Raab, Thomas Strothotte
Institute for Simulation and Graphics, Otto-von-Guericke-University of Magdeburg
Universitätsplatz 2, D-39106 Magdeburg, Germany
e-mail: {bernhard, raab, tstr}@isg.cs.uni-magdeburg.de

Abstract

In this paper we develop a method for zooming in complex information spaces consisting of geometric models and associated textual explanations. In particular we introduce a technique to extend fisheye views to the exploration of three-dimensional models.

The result of a zooming process is a modified geometric model which is rendered and presented to the user with appropriately selected and well-placed textual labels. The user interacts with such coherent presentations to explore the information space.

Our methods have been implemented in a system called the ZOOM ILLUSTRATOR. The system maintains a close relationship between images and associated text, with interaction on the textual part influencing the appearance of the graphical part and vice versa. We demonstrate our system on medical illustrations, in particular from anatomy.

Keywords: Fisheye Views, 3D-Fisheye, Interactive Illustrations, Image-Text-Relationship

1 Introduction

Interactive 3D-graphics bears a high potential for the explanation of complex spatial phenomena as can be found for example in engineering and anatomy. The interactive exploration of complex 3D-models is crucial for spatial understanding. While this is well-recognized in the literature, not enough effort has been spent to date on flexibly combining interactive 3D-graphics with textual descriptions. This, however, is necessary to exploit the potential of computer graphics for educational purposes.

Borrowing from textbooks gives hints on how to combine images and text. Images often are surrounded by labels referring to their parts via reference lines. Explanations refer to the spatial structure and are enhanced by cross references as to spatial relations. In textbooks, however, explanations are generally not integrated in an illustration but are placed under an image or even on a separate page, which complicates comprehension.

Interactive systems can handle this problem and tailor the presentation to the information requested. In current hypermedia systems, however, this often results in the display of multiple windows, the management of which imposes a high burden to the user. To make matters worse, an obvious coordination of explanations and images is missing.

Based on these observations we developed a system to generate interactive image-text-combinations within *one* window. Users can ask for explanations which are incorporated immediately in the illustration. To avoid overlapping information and the need to rearrange the desktop, fisheye views, as introduced by FURNAS (1986), are used. They simplify navigation because different levels of detail can be presented simultaneously in different parts of an image. The placement and scaling of information is based on user interaction. To emphasize the importance of zoom-techniques for our system, we call it ZOOM ILLUSTRATOR.

To maintain the image-text-relations when interaction occurs, the placement of text is influenced by interaction on the related graphics part and, on the other hand, it influences not only text, but also parts of the 3D-model. This includes changes of material properties to ensure *visibility* and *recognizability* of relevant parts. However, to explain a small detail, it is not enough to emphasize it or to remove objects which occlude it – it is necessary to scale it to make it recognizable. The naive approach to this is to scale the whole model uniformly so that it becomes so large that a significant portion must be clipped. Thus, however, the context is lost. From a cognitive point of view it is more useful to scale up the interesting detail only and automatically scale down others or, in other words to use a distorting fisheye zoom in three dimensions. The ability of a 3D-fisheye view to illustrate local graphical detail in its context allows to apply the ZOOM ILLUSTRATOR to handle not only compact 3D-models, but also models where different parts are distant from each other.

Our paper is organized as follows. Section 2 describes previous work on illustration design and related algorithms for producing fisheye views. Section 3 presents concepts for using zoom techniques in interactive illustrations. Some of the problems involved when applying these concepts to 3D-models are discussed in Section 4. A prototypical system for exploring complex information spaces is presented in Section 5, while concluding remarks are presented in Section 6.

2 Related Work

Our work has been inspired by research on illustration design and on results in the area of fisheye views. Work on illustration design concentrated on techniques for illustrating 3D-models and on combining images and text. As illustrations with complex 3D-models and related text form a large information space, we look for fisheye views to turn principles from static illustrations into an interactive illustrator.

The IBIS-system (*Intent-Based Illustration System*) of SELIGMANN and FEINER (1991) is the pioneering work in the field of 3D illustration design. It is based on an extensive study of what technical illustrators do in order to achieve their intent. Besides techniques to bridge the gap between intent and realization in static illustrations, an interactive component is included. IBIS illustrates complex 3D-models using transparency, cut-aways and insets (small details scaled up in a large image). We learned from their work that *visibility* and *recognizability* of important objects are crucial for illustrating 3D-models. With IBIS, excellent images can be produced; however, textual descriptions are not included.

In contrast to the IBIS-system, the work carried out in the WIP-project (*Knowledge-Based Information Presentation*, see WAHLSTER *et al.* (1993)) targets at the coordination of images and text. This includes sophisticated strategies to annotate an image. They extended the knowedge-based approach to a semi-interactive method to illustration design (see RIST and ANDRÉ (1994)). While the process of generating an illustration has become interactive the goal remains to design a final illustration and not to provide an interactive illustrator.

FURNAS' *Generalized Fisheye Views* (recall FURNAS (1986)) pioneered the idea of fisheye views. The placement of information is guided by a *degree of interest* (DOI) which considers the distance to the user's current *focus point* (FP). The DOI-calculation involves static and dynamic factors.

NOIK (1993) exploits fisheye views for navigation in hypertext systems. We learned from his work that for hypertext systems, the DOI-calculation must consider the *conceptual* distance between different items, rather than the spatial distance. This requires a concept for categorizing nodes, as to which a conceptual distance can be defined. Moreover, such a classification is crucial to generate illustrations focusing on certain aspects.

NOIK (1994) surveyed and categorized Fisheye applications. Existing approaches apply filtering, distorting techniques or both. Filtering Fisheye techniques display or suppress the rendering of nodes according to their DOI. Distorting fisheye views keep more to the photographic nature by applying a non-linear distortion.

From a cognitive point of view, it is desirable that changes between successive views are smoothly animated. This is accomplished with the *Continuously Variable Zoom* introduced by DILL *et al.* (1994). The variable zoom manipulates rectangular areas in which all information to be presented is embedded. If more detail is requested for one piece of information, called a *node*, the corresponding rectangle is enlarged at the expense of others, the size of which are reduced accordingly. As a consequence of the scaling, the representation changes depending on application-specific thresholds. In the terminology of NOIK their algorithm combines filtering and distortion fisheye views.

BARTRAM *et al.* (1994) enhance the variable zoom by providing "contextual assistance", that is by combining the zoom with reasoning techniques to select the most appropriate representation. Recently SCHAFFER *et al.* (1996) summarized the approach.

First approaches to three-dimensional fisheye views were presented by MACKINLAY *et al.* (1991) using a linear transformation of a 1D space onto a *perspective wall* and by ROBERTSON *et al.* (1991) using *cone-trees* to visualize hierarchies in 3D space. NOIK refers to 3D-Fisheye views, which are based on perspective transformations as *Implicit Fisheye Views* (recall NOIK (1994)).

Our system differs from previous illustration systems as we focus on providing flexibility to interactively explore an illustration with complex 3D-models. This poses high demands on the coordination of images and text when working interactively, but reduces the complexity of the generation of an initial layout, as this is only the starting point for interaction. As we apply fisheye views to both the exploration of a 3D-model and to the navigation through text, we offer a uniform interaction and enable users to focus on details in their original context.

3 Zoom Techniques for Illustration Purposes

Integrated image-text illustration systems need to deal with very different information domains. On the one hand we have the graphical domain consisting of a set of scanned or rendered images (the latter being generated from surface or volumetric models). On the other hand there is a lot of textual information to convey, describing function, relations, assembly (see Figure 1, which depicts an information space).

While there are often limited representations available on the graphical side, the textual information representation can be chosen by the several aspects. By contrast, the presentation of textual information is restricted to either displaying or speaking text, whereas the graphical part may be displayed with a variety of attributes (like color or transparency).

Therefore, a coherent illustration system needs to select uniformly the representation (*what* information to convey) as well as the presentation (*how* to convey the information) in the different information domains. Furthermore, the interaction with both parts of the information space should also be uniform to enable users to access the information through a common interface.

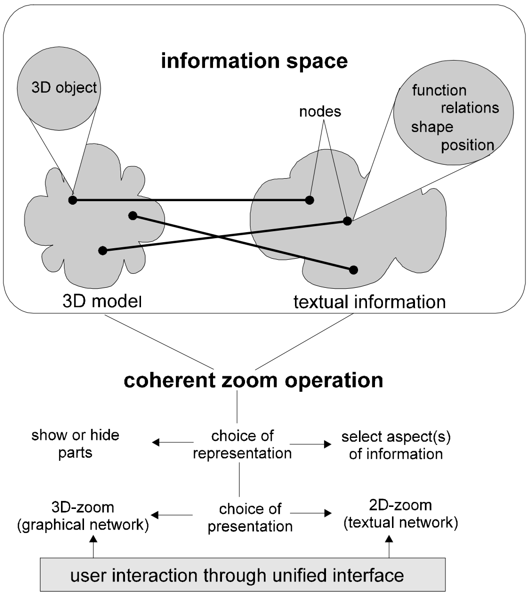

Figure 1: Coherent zoom operations in an integrated image-text illustration system.

The application of fisheye zoom techniques to graphical as well as to textual navigation helps to achieve these goals. The choice of a representation can be approached by either explicit user interaction or the computation of what we call the *aspect of interest*, based on the interaction history and on the *level of detail* (LOD). The choice of representation is done by applying fisheye views both in the textual and graphical domain. User-interaction based on zoom operations give comparable feedback if applied to the graphical or textual parts. We will refer to this point later on in the paper.

3.1 Selecting Representations

A major problem in an interactive illustration system is to develop algorithms to compute what should be presented to the user. BARTRAM *et. al.* (1994) used a system of autonomous agents to decide what representations to activate in a hierarchically clustered network. Their *intelligent zoom* defines strategies how to map representations to different DOIs resulting from fisheye zoom techniques.

This technique is sufficient if the selection of an appropriate representation merely adjusts the level of detail (LOD), i.e. if one DOI corresponds exactly to one representation. As was pointed out recently by RÜGER *et al*. (1996), in many applications it is more natural to classify representations not only as to their LOD but also as to their aspect (different representations for the same LOD). By defining a *representation matrix*, the approach is adjustable to different application domains (see Figure 2 for an example of a representation matrix).

The set of representations is different for nodes belonging to different categories. Each category, e.g. an organ-system in anatomy, is characterized by *aspects* under which it can be studied. In this case, the selection of a representation considers not only the DOI, but moreover, the membership of a node to an application-specific category. As a consequence, not only the available level of detail (based on the DOI) has to be taken into account but also the aspect a user is interested in. Therefore, an *aspect of interest* (AOI) based on the interaction history is calculated and used in addition to the DOI to select an appropriate representation.

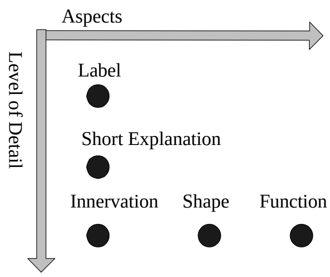

Figure 2: Representation matrix for one category in the ZOOM ILLUSTRATOR. The appropriate representation is chosen among the aspects in available levels of detail.

In the ZOOM ILLUSTRATOR the representations are as follows: all nodes have a *label*-representation. The more extended representations differ depending on the category. So, for instance, muscles have explanations as to their function, the supply with nerves (*innervation*) and their shape, whereas nodes belonging to the category bones simply have an explanation as to their location. Since the muscle-category is an example with different aspects for one LOD, the system must decide whether the user wants to focus on the shape, on the innervation or on the function. Based on the representation matrix, the DOI selects a LOD (a row in the matrix) while the AOI selects the appropriate aspect (a column).

3.2 DOI and AOI Calculation

As mentioned above, the DOI and the AOI form the base for selecting appropriate representations.

Factors influencing the DOI can be classified into static versus dynamic factors. Static factors are summarized in an API (*à priori interest*) function. It considers the nodes' position in the hierarchical structure, the size of the related 3D-object and the amount of textual information available. Nodes with a large amount of text assigned to it are regarded as more important because they offer more links as to cross-references and can serve as starting points for further interaction. The decision which objects to label initially is based on the API. Furthermore the API is evaluated to modify related graphical objects, the transparency value of which is adapted. Note that hierarchical relations between nodes exist, but in fact, they occur scarcely, which makes other factors predominant.

Dynamic factors of the DOI are influenced by zooming operations or by changes of the viewing direction. Zooming on the textual part influences the *conceptual distance* of nodes (recall NOIK (1993)). In our system, the conceptual distance depends on the occurrence of hyper-links and on the layer/sublayer-information. The conceptual distance from node *A* to node *B* is low if they share the same category and if there are hyper-links between them.

Changes of the viewing direction – as they occur when the model is rotated – influence the depth values of graphics objects. This is considered in the DOI-calculation of the related nodes (recall the association between nodes and 3D-objects in Figure 1). This leads to higher DOIs for nodes related to objects nearer to the observer than for nodes related to distant graphical objects. The DOI-calculation is summarized in Equation 1. Note that the terms API and Static DOI are equivalent.

```
(1) Static  DOI=f(size,hierarchy,available text)
(2) Dynamic DOI=f(depth,conceptual distance)
(3) DOI        =Static DOI * Dynamic DOI
```

Equation 1: DOI-calculation

The AOI-calculation is invoked if several aspects with the current DOI exist for a node (recall the matrix in Figure 1). The AOI-calculation is based on a record of

- how often an aspect was presented,
- how long ago it was presented, and
- how long it was visited

These values are combined to produce a numerical value for each aspect in each category. Note that with this AOI-calculation, the selection of a representation is straightforward and does not involve any backtracking.

3.3 Selecting Presentations

After an appropriate representation has been chosen, the system needs to present this representation appropriately. In the ZOOM ILLUSTRATOR, this includes the presentation of textual descriptions and the graphical presentation. These problems are discussed in the next sections.

4 Applying Zoom Techniques

To ensure that the image is not occluded by textual information and vice versa, the ZOOM ILLUSTRATOR's window is subdivided into a central part for the image (covering 50% of the screen), as well as a left and a right part for a network with textual descriptions, each of which occupy 25% of the screen (see Figure 3).

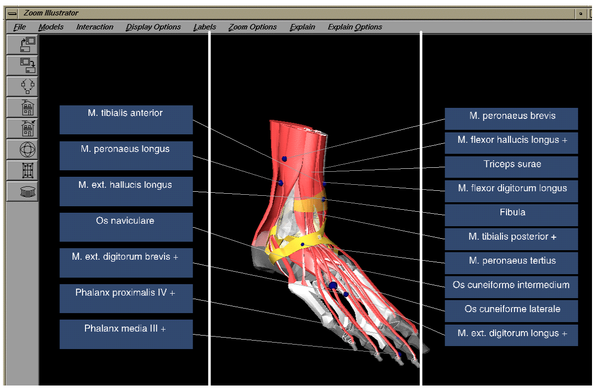

Figure 3: Basic layout with text networks on the left side and right side of the rendered 3D-model with white lines superimposed to show the separate areas

This distribution, however, is only a starting point. Text networks and the 2D-bounding box of the rendered image are parts of a top-level zoom network. When interaction occurs, the DOI of the two networks and the graphics box change, resulting in a varying size (from -10% to 10% compared to the original size). With this top-level zoom, a network can provide more space if several explanations are requested. Moreover, individual nodes are allowed to extend their width by 10% to be able to accommodate more information. The size of the explaining text is limited to some 30 words per explanation. However, in most cases this is enough for describing spatial relations.

Usually, the continuous zoom works independently in the left and right text area to prevent irritating changes in one part due to interaction in the other. However, if one explanation is displayed which consumes all the space available on one side, one or two nodes are moved to the other side. This movement consists of a zoom step to provide the space in the target network and an animated movement.

If users ask for an explanation, the corresponding node is zoomed up to accommodate the required amount of text. Other nodes are scaled down appropriately. If they become too small to display their label, only a rectangle with a reference line is displayed, so that interaction with this node is still possible. If the node becomes even smaller it is completely closed. We describe later how users can prevent nodes from being closed and how to get nodes back which were closed inadvertently.

4.1 Combining Fisheye Zoom on the Textual and on the Graphical Side

In the models seen so far there are typically many objects in very different sizes, leading to the problem of how small objects can be illustrated clearly. In addition, an interactive illustration system should support learning goals directly derived from a graphical model such as:

- recognizing relationships between several objects,
- recognizing the positions of objects in the model,
- inspecting (occluded) objects, and
- inspecting the shapes of objects in the model.

This is often done by simply scaling the overall model for displaying parts of it or adding separate windows containing objects. However, there are several problems with these approaches. Scaling the model is in our understanding not always helpful because the context of the illustration is lost. In the case of the ZOOM ILLUSTRATOR, scaling is even useless, as the textual labels could hide parts of the model or refer to locations outside of the visible area. Separate windows, however, have other problems, as for instance those associated with the user's mental integration of two or more images as well as the navigation in several windows. Furthermore, the windows require space on the screen and have to be placed in a location where they do not hide the current illustration window.

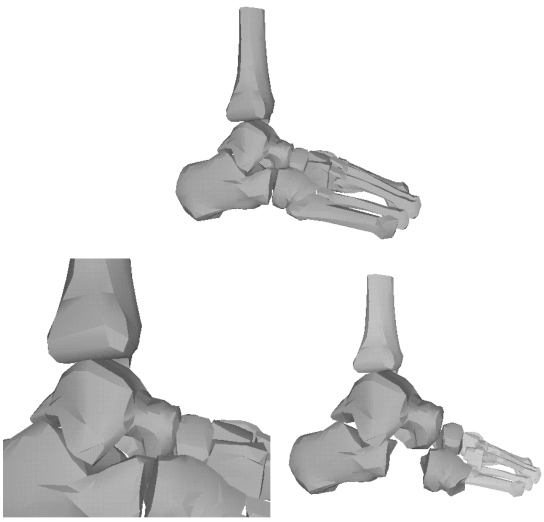

Figure 4: Comparing full-zoom (left) and 3D Fisheye zoom (right) in a 3D model (top)

Therefore, a different approach has been used, integrating the fisheye zoom techniques into the graphical part (see Figure 4). Based on two-dimensional techniques, the fisheye zoom has been extended to allow navigation in different dimensions (see RAAB, RÜGER (1996)). In the 3D case, the regions covered by objects in the model are expressed as 3D bounding boxes allowing the application of the interval structure independently in each direction (i.e. along the *x*-, *y*- and *z*-axis) and afterwards the re-construction of new bounding boxes. The extension includes the handling of overlapping boxes, which is crucial for illustrating 3D-models. In contrast to implicit Fisheye Views, as for example in Cone Trees and the Perspective Wall (recall ROBERTSON *et al.* (1991) and MACKINLAY *et al.* (1991)), the 3D-Zoom by RAAB explicitely distorts a 3D-model. RAAB, and RÜGER (1996) present a detailed description of the algorithms behind the 3D-Zoom. The advantages of this approach for the ZOOM ILLUSTRATOR are numerous.

- *Detail and context*: Objects can be zoomed inside the model's space, enabling the enlarged illustration of small objects in the context of the overall model (see Figure 4).
- *Constant space*: The space occupied by the model is held constant. As in the 2D case (where the screen determines the area to display information), the 3D space available for displaying graphics can be defined to avoid overlapping graphics and text.
- *Uniform interaction:* Both display techniques are based on the same techniques and do therefore behave uniformly during interaction.

The last point is in particular of interest when using the ZOOM ILLUSTRATOR for educational purposes. The overhead to understand how navigation and interaction work decreases – once a user has understood textual navigation, he or she will probably manage to cope with the graphical interaction.

Although primarily used to display small objects in detail, the 3D-Zoom can be exploited in different ways. For instance, imagine a user wants to know about a particular part in the geometric model, which is not (or not yet) part of the textual network. By interacting with the graphics, the user expresses interest in a particular part. This, in turn, can be used to show up textual explanations of the graphical part. Even more interesting, the amount of textual information displayed can be coupled with degree of interest expressed implicitly during the interaction with the graphics.

In this sense, the 3D-Zoom allows for dual interaction with the textual as well as with the graphical part of the ZOOM ILLUSTRATOR. In addition, as the display and interaction techniques are similar, a uniform behavior is obtained when interacting with the textual and graphical parts.

4.2 Adaptive Graphical Zoom

We still have to describe how we incorporate the graphical zoom in the user interface. At first glance it might be not convincing to zoom within the 3D-model at all, because the study of topological relations is an important issue and these are distorted to a certain extent by the graphical zoom. However, adaptive scalings are a common praxis in traditional teaching materials (see WEIDENMANN (1993) for a discussion of the didactic effects of such modifications). If adaptive zooming is an appropriate means of focusing in static illustrations, the same ought be true for interactive illustrations as the user can see the change in an animated movement instead of being confronted with a final image. When the system carries out a distortion it does so by presenting an animation, i.e. it shows the actual movement from the undistorted to the distorted view. This lets the user know the extent of the distortion. In the final image a user can see the items of interest better than before the distortion.

The graphical zoom is dedicated to the emphasis of 3D-objects which are small in relation to the overall model size. Even these objects are zoomed carefully (scale factors vary between 1 and 2), to ensure that the resulting image is not distorted heavily. Figure 5 gives an example of an illustration with a muscle enlarged to support its verbal explanation (left side).

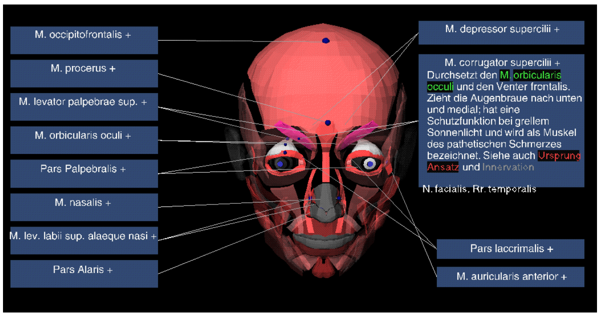

Figure 5: One muscle (above the eyes) has been enlarged to be explained while others have been scaled down and moved away

To prevent heavily distorted views, we do not provide full access to the graphical zoom but invoke it only in a restricted way initiated from the system. It is important that the user can reset the zoom so that all changes on the relative sizes are undone.

4.3 Enhancing Navigation in Textual Information

One important question when applying zoom techniques concerns hiding nodes automatically. While the zoom algorithm generally produces comprehensible layouts, it might be irritating, especially for beginners, when a node disappears due to the size request of another node. Even when users understand what has happened they might not know what they are supposed to do to get a node back which has disappeared inadvertently. The method to get the hidden node back by scaling down another node is not very intuitive and, moreover, a trial-and-error process. This raises two questions:

- How to prevent nodes from being closed (preventive action)?
- How to get nodes back which have been closed (curative action)?

We present one possible solution for each question.

4.3.1 Prevent Nodes from Being Closed

To prevent nodes from being closed (preventive strategy), we introduce an additional network, a *pinwall*, as a container for some privileged nodes (up to 4). These nodes are not exposed to the zoom and can therefore not be closed, however they cannot be scaled up to show an explanation as long as they belong to the pinwall. This strategy is more natural than modifying the zoom algorithm itself.

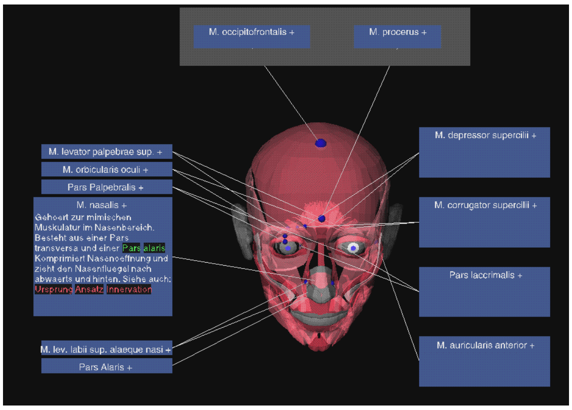

Figure 6: An illustration with two nodes residing at fixed positions at the pinwall (upper part)

The user can initiate an animated movement of a node to the pinwall (above the rendered image) where it remains at a fixed position (see Figure 6). The node is scaled so that just its label can be accommodated. The movement to the pinwall is followed by a zoom step to consume the space in the source network no longer needed. Nodes residing at the pinwall are still connected to the image via reference lines, which are updated as usual if the 3D-model is rotated.

4.3.2 Selection of Hidden Nodes

To get a node back once it has disappeared (curative strategy), it should be possible to select it via its label. The straightforward idea to do this is to construct a hierarchical menu with layers, sublayers and individual nodes to select a node, which will be subsequently zoomed up. However, there is a clear cognitive gap between a menu – either a pop-up-menu or a pull-down menu – and the illustration. Therefore we designed a 3D-widget we call a *rondell* which contains all labels grouped according to their category. The design of this 3D-widget is inspired by the work carried out at XEROX Parc on 3D-interfaces (recall MACKINLAY *et al.* (1991)).

The *rondell* can be rotated by clicking at the disks at the lower and the upper part one for a rotation to the left and one for the rotation to the right. The color of the nodes which are closed is a saturated blue (instead of a weak gray for the nodes already presented), to encourage the user to invoke the node (see Figure 7).

In the terminology of NOIK, the Rondell presents an implicit 3D-Fisheye View based on perspective transformations. It is capable of displaying non-hierarchical data. Informal tests with medical students revealed that the ability of the rondell to browse to all textual information available is regarded as useful and therefore justifies the screen space occupied.

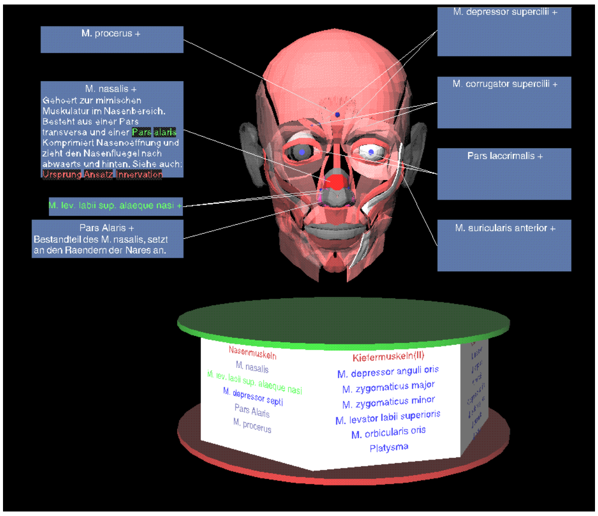

Figure 7: A label from the rondell was selected at which is displayed in the illustration (left side)

The user can select a label on the rondell, which results in the display of the corresponding node with an additional highlighting to emphasize what has happened. The rondell can be rotated by clicking on the top or bottom part, the top part being used for rotations to the left, the right part for rotating to the right. Furthermore, the rondell is used for exploration of the 3D-model. If a certain option is set, a simplified material editor is presented. This allows to show/hide the related graphics part and to change its color. Informal studies with users indicate that they like this freedom, especially to be able to hide objects occluding something essential. This kind of interaction is recorded and influences the DOI-calculation for nodes of the text-networks.

While users tend to recognize the rondell as an appropriate 3D-widget and indicate they like it, it also has disadvantages:

- *Familiarity*: A rondell is less familiar than conventional menus and its usage must be learned.
- *Performance*: A 3D-widget with 3D-text requires considerable computing resources to render, which either slows down the system or requires to reduce the quality (resolution) of the text-presentation.

4.4 Post-Processing to Enhance the Layout

On the textual part, the result of the zoom algorithm is not directly mapped onto the presentation. Instead, the result is evaluated as to whether annoying effects would arise. If this is the case, a post-process is carried out, to improve this situation. These effects are due to the fact that the zoom itself is continuous while the selected representations (and the space they require) are discrete.

The goal of the post-process is to use the space available appropriately and to avoid that too many representations change at once. For a single node, the number of changes occurring can be reduced in introducing thresholds (the size at which a more detailed representation is displayed is larger than the size at which it is hidden). Thresholding provides a tolerance and temporal coherence. However, it has only a local effect and is limited to small tolerances to avoid overlapping of information.

The post-process recalculates the space available based on the current DOI. This recalculation takes place after the zoom step but before the selected representations actually change.

In the following it is described which problematic effects occur often when applying the zoom-algorithm in a straight-forward manner and how to avoid them.

- *Disappearance of many labels at once*

If a space request for one node reduces the sizes of others so that several of them (simultaneously) become just a little bit too small to be labeled.

Postprocess: The recalculation cause that nodes with the lower DOIs get even smaller while others get larger to accommodate their labels (see Figure 8).

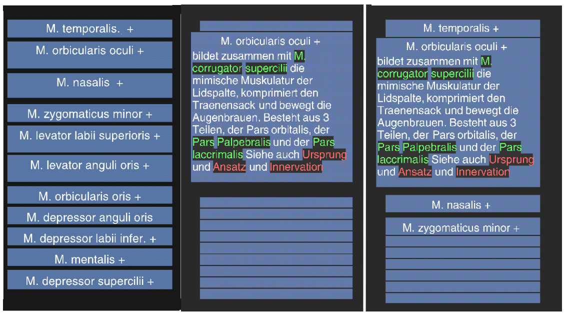

Figure 8: Parts of text-networks with empty rectangles representing nodes too small to display their label.
Several labels disappear at the same time (middle image). Reduction is achieved with a post-process based on the current DOI (right image)

- *Disappearance of a long explanation*

When a long explanation must disappear because the node's size is just too small, but still quite large, the space is also not used appropriately.

Postprocess: The space which is not needed for one node is redistributed to others.

- *Disappearance of a recently chosen representation*

It is likely to happen, that the zoom causes representations to disappear which have recently been requested. This might be the most annoying effect. Consider the following case: The user requests an explanation e_1 for node n_1, the system performs a zoom-step to provide just the necessary space and the following request to display an explanation e_2 would reduce the size of n_1 and remove e_1 immediately (see Figure 9).

The post-process ensures, that the selected representation of a node is *conservative*, so that small changes of a node's size do not influence its representation.

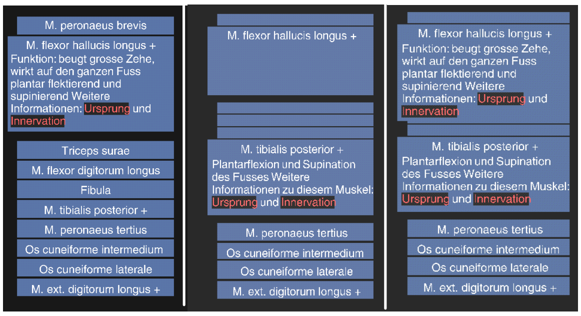

Figure 9: Request of a second explanation.
Left: before zoom step, middle: zoomstep without a post-process, right after post-process.
Without the post-process the previously displayed explanation is hidden (upper part in the middle image), while both can be displayed with small changes (right side)

5 Implementation

The ZOOM ILLUSTRATOR is implemented with Open Inventor™ on a Silicon Graphics Indigo² Workstation. This platform allows transforming 3D-models of medium range complexity (between 5K Polygons to 15K Polygons) in near real-time (between 6 and 9 frames per second). Open Inventor™ provides powerful interaction facilities and an elegant way to manipulate its internal scene-description. However, these interaction facilities are not for free, but need considerable computing resources. Because real time response is crucial for the system, careful quality-speed trade-offs must be chosen.

Customization of Zoom Techniques

Our zoom implementation is influenced by several parameters, the most important of which are adaptable via the interface. The speed of the zoom is an important parameter for customization. For a beginner it is helpful to see how zoom works and continuously changes all nodes of a network. For an experienced user, however, it is annoying, if it takes too long to animate a zoom step. Therefore the user can choose the speed and thereby the level of continuity in the zooming process.

Furthermore, the user can decide whether nodes which are mentioned in an explanation as hyper-links are automatically scaled up to accommodate at least their label. To turn this feature on, is useful if only one or two nodes are mentioned, but in the case of more nodes it may be irritating if the request for one explanation results in such a radical change of the presentation.

Besides this, the user can decide whether transparency is exploited to de-emphasize objects. Because transparency is expensive users can switch the use of transparency on and off. When transparency is switched off, wire-frame rendering is used to deaccentuate objects which hide relevant objects.

6 Concluding Remarks

Successful interactive illustration systems require facilities to explore text and graphics while maintaining a close relation between these basic media. In this paper we have presented a methodology to bring this principle into being. While it may seem natural at first glance to use fisheye views for navigating in textual information our key contribution is that we make it work uniformly on 3D-geometric models and in hierarchically organized textual information spaces.

Room for improvement in our system exists in the area of flexible rendering. It would be desirable if parts of the 3D-model which have been zoomed up or zoomed down would be rendered differently so as to further accentuate or deaccentuate them. Also, the degrees of freedom inherent in the 3D fisheye zoom have not yet been fully exploited.

It is desirable for an interactive system to produce figure captions to further assist a user in orienting himself in the information space he or she is exploring. In textbooks, complex images are described carefully by captions. They explain how the "real" situation differs from the depicted image and why changes have been performed. Captions often include comments about what has been scaled, reinforced or simplified. We conjecture that helpful captions can be generated using the internal representation of the modifications which have been made on the 3D-model.

The development of our system has been accompanied by informal tests with medical students and colleagues. Especially the customization of zoom techniques is inspired by these tests. However, a more systematic usability study is necessary to evaluate the effectiveness of the interaction facilities offered.

References

André, E. and T. Rist (1994)
"Multimedia Presentations: The Support of Passive and Active Viewing", *AAAI Spring Symposium on Intelligent Multimedia Multimodal Systems*, Stanford, Mars, pp. 22-29

Bartram, L., R. Ovans, J. Dill, M. Dyck, A. Ho, and W.S. Havens (1994)
"Contextual Assistance in User Interfaces to Complex, Time-Critical Systems: The Intelligent Zoom", *Proc. Graphics Interface,* Banff, May, pp. 216-224

Dill, J., L. Bartram, A. Ho, and F. Henigmann (1994)
"A Continuously Variable Zoom for Navigating Large Hierarchical Networks", *Proc. of IEEE Conference on Systems, Man and Cybernetics*, pp. 386-390

Furnas, G.W. (1986)
"Generalized Fisheye Views", *Proc. ACM SIGCHI'86 Conference on Human Factors in Computing Systems*, Boston, April, pp. 16-23

Mackinlay, J.D., G.G. Robertson, and S.K. Card (1991)
"The Perspective Wall: Detail and context smoothly integrated", *Proc. of CHI '91*, New Orleans, April, pp. 173-179

Noik, E.G. (1993)
"Exploring Large Hyperdocuments: Fisheye Views of Nested Networks", *Proc. of ACM Hypertext and Hypermedia*, Seattle, pp. 192-205

Noik, E.G. (1994)
"A Space of Presentation Emphasis Techniques for Visualizing Graphs", *Proc. of Graphics Interface*, Banff, May, pp. 225-233

Raab, A. and M. Rüger (1996)
"3D-Zoom: Interactive Visualization of Structures and Relations in Complex Graphics", *Proc. of Bildanalyse und -synthese*, Erlangen, November, pp. 123-132

Robertson, G.G., J.D Mackinlay, and S.K. Card (1991)
"Cone Trees: Animated 3D Visualization of Hierarchical Information", *Proc. of CHI*, New Orleans, April, pp. 189- 202

Rüger, M., B. Preim, and A. Ritter (1996)
"Zoom Navigation: Exploring Large Information and Application Spaces", *Proc. of Advanced Visual Interfaces*, Gubbio, May, pp. 40-48

Schaffer, D., Z. Zuo, S. Greenberg, L. Bartram, J. Dill, S. Dubs, and M. Roseman (1996)
"Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods", *ACM Transactions on Human-Computer-Interaction*, Vol. 3, No 2, pp. 162-188

Seligmann, D.D. and S.K. Feiner (1991)
"Automated generation of intent-based 3D-illustrations", *Computer Graphics 25(4),* Chicago, July, pp. 123-132

Wahlster, W., E. André, W. Finkler, H.-J Profitlich, and T. Rist (1993)
"Plan-Based Integration of Natural Language and Graphics Generation", in *AI-Journal* 63, pp. 387-427

Weidenmann, B. (1993)
"Informierende Bilder", *Wissenserwerb mit Bildern*, Huber-Verlag, Bern, pp. 29-57