Abstraction in Interactive Computational Visualization[*]

Thomas Strothotte, Bernhard Preim
Department of Simulation and Graphics, Faculty of Computer Science
Otto-von-Guericke University of Magdeburg, Universitätsplatz 2, D-39106 Magdeburg
tstr@isg.cs.uni-magdeburg.de

Abstract

In this paper we introduce the notion that visualizations have a number of points in common with *abstract data types*. In particular, visualizations must be designed such that certain questions which users frequently have or have with a high probability must be answerable quickly and easily. Furthermore, it is recognized that for a given visualization, only certain questions can be answered. This leads to the definition of the process of *abstraction* to render images which cater to the information needs of users. The paper gives an overview of theoretical concepts and practical results pertaining to abstraction in interactive computational visualization.

1 Introduction

In an effort to devise new techniques for rendering 3D images of objects, it is instructive to take a fresh look at areas of application which have thus far alluded the use of rendering techniques. In fact, almost all 3D images which meet our eyes in the media or in books have either been produced by cameras or are hand-drawn i.e. have not been rendered from an underlying model. Among those which are hand-drawn, it can be observed that many of the objects are not drawn entirely to scale, the level of detail varies over the image to a strong extent, and physical models of light, which are at the heart of all computer rendering techniques, are often ignored. In short, such hand-made images represent *abstractions* of physically correct renditions of the objects.

This paper first argues for the need for techniques of abstraction for 3D geometric models. We contend that applications for which only hand-drawn images are widely used would not be amenable to today's computer-based rendering techniques because of the strict adherence to physically-based models of light and the emphasis of rendering techniques on photorealism. We analyze reasons for deviating from such models, survey new techniques for achieving the desired effects, and show examples.

The paper is organized as follows. Chapter 2 presents a theoretical framework for generating images and explains why it is important that rendered images have to cater to the

---

[*] The concepts introduced in this paper are elaborated in the forthcoming book:
Th. Strothotte (ed.) *Abstraction in Interactive Computational Visualization: Exploring Complex Information Spaces,* Springer-Verlag, Berlin-Heidelberg-New York, 1998. Preprints of individual chapters are available from the editor.

information needs of users. In Chapter 3, the process of abstraction is defined and characterized as being able to develop models to take into account a user's dialog context. Chapter 4 shows examples of the process of abstraction and indicates the nature of the underlying algorithms. Concluding remarks are made in Chapter 5.

2 Visualizations Viewed as Abstract Data Types

2.1 Sample Image

First we shall examine an example of a typical hand-drawn image as shown in a medical textbook. Figure 1 shows the image of a part of the human body within a book on surgery. It is used by medical students to learn how to carry out a particular operation, as well as by practicing surgeons for later reference. The image is a case in point for this paper as practically all medical books of this kind show hand-drawn images which scientific illustrators drew by hand, rather than images generated by computer from geometric models. Why?

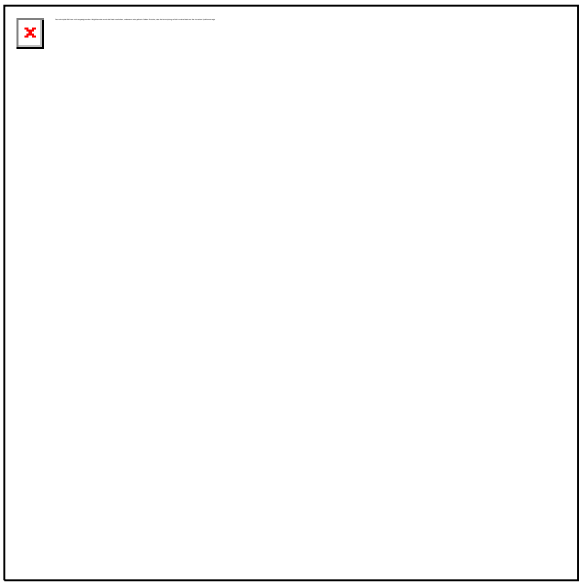

Figure 1: Example of a drawing taken from a textbook on surgery ([Sae87])

First let us analyze the image itself. The following points can be observed: The image

- is hand-drawn with pencil on paper
- presents only some details, while some areas are intentionally left blank
- uses different kinds of cross-hatching and stippling to differentiate between the different kinds of objects,
- emphasizes the curvature of objects

- shows certain parts (like the arteries to the left) simply removed (indeed, they look cut off)
- is labeled in a well-balanced manner.

The list could be continued on to include many more points. Indeed, graphical features such as the ones mentioned above which enhance the use of images for the purposes of learning have been referred to didactification [Wei89]. One final attribute of such images cannot be seen readily by laypersons: Most are not drawn to scale. This is sometimes noted in the figure caption; alternatively, sometimes medical books have a ”disclaimer” at the beginning of the book explaining that most images are not to scale so as to be able to emphasize and show more detail in those parts which are important for the accompanying text.

2.2 Analysis

To explain why scientific illustrators produce images such as the one shown in Figure 1 which deviate significantly from images as are produced by rendering software, we shall examine abstract data structures. We contend that when an end-user examines an image in order to obtain information, the situation is analogous to program an abstract data type (ADT). Recall that an ADT can be thought of as a mathematical model with a collection of operations defined on that model [AHU83]. For example, sets of integers along with the operations INSERT, FINDmin, and DELETEmin from what we call a priority queue. No other operations than the ones specified in the ADT may be applied to the data. Normally various implementations of a given ADT exist; they vary in the space and time complexity of the operations. For a specific application at hand, the user or programmer chooses the implementation which optimizes the performance in the sense that operations which are performed often must be inexpensive, while those which are performed seldomly can be less efficient.

By analogy, we propose that pictures be thought of as instances of ADTs on which certain operations are defined. We contend that certain questions which an end-user may have can be answered by inspecting a particular rendition of a geometric model, while other questions either cannot be answered quickly, or perhaps even cannot be answered at all. Furthermore, of the many graphical representations imaginable of a geometric model, one which most suits the application at hand is chosen by the user. The questions which an end-user may ask play the role of the operations on an ADT, while the specific rendition chosen plays the role of the implementation of the ADT. Questions which an end-user is most likely to pose or is expected to be posed most often must be answerable by a brief inspection of the image. By contrast, questions which are seldomly posed may require a user to inspect an image for a significantly longer amount of time. Finally, situations may arise in which questions which are not expected cannot be answered using the given rendition.

An example will clarify the analogy. We mentioned earlier that the size relationships of the objects in images such as that shown in Figure 1 are routinely changed in medical books. The reason is that in the context of their use, it is important that certain features be easily recognizable; questions pertaining to these features are expected often, hence the rendition is optimized to enable viewers to answer these questions easily and quickly. By contrast, in the same context the authors of the medical book do not expect end-users to ask questions about the specific sizes of individual objects portrayed in the image. An end-user cannot answer such questions correctly by inspection of the image.

We conjecture that rendering techniques of computer graphics are not yet used in a variety of areas of application because the algorithms for implementing pictures as flexible ADTs have not been devised which can compete with those a human scientific illustrator or graphic artist uses. Hence if computer graphics is to play a decisive role in such areas of application, the process of adapting a rendition to the specific use intended must be developed.

3 Abstraction

We refer to the data from which an image is generated as a *complex information space*. A complex information space is not only large, i.e. consisting of many components, but also heterogeneous meaning that different kinds of components exist. Such an information space typically contains a geometric model, but also information pertaining to that model, such as identifiers for individual parts of the object being modeled, and other information pertaining to their functions. We define *abstraction* as

> *the process by which an extract of a complex information space is refined so as to reflect the importance of the features of the underlying model for the dialog context and visualization goal at hand.*

The process of abstraction is carried out by the computer but is guided by the interaction with the end-user. For effective abstraction, it is decisive to determine what in the complex information space is important for the viewer, and what is not. This may seem like a rather obvious statement, but it is in fact overlooked when an image is generated without taking the dialog history into account.

For our purposes, we view the following features of the process of abstraction which need attention in tools for supporting computer visualization:

- *Geometry*.
  Adjusting the size, shape, color and orientation of parts of a model in combination with their level of detail and their rendering style.
- *Level of detail and rendering style.*
  Gradually removing detail and adjusting the rendering style is necessary. Details can be removed on a gradual scale by any one of or a combination of several techniques.
- *Text-graphics correspondence.*
  Since many images contain at least some text, and often also a figure caption, the linguistic comments must be made to fit of the graphics.
- *Movements*.
  Adjusting the movements of objects in an animation is important to ensure that the abstractions referred to above are in unison with the objects over time.

4 Examples

A variety of techniques have been developed for carrying out the process of abstraction introduced above. We shall briefly describe some of the techniques and illustrate them using specific examples. The interested reader is referred to [STR98] for a complete description of the underlying phenomena.

4.1 3D Zoom

Raab and Rüger [RR98] devise an algorithm for selectively zooming in objects of arbitrary dimensionality and apply it in particular to 3D geometric models. Their method extends the work of Schaffer et al. ([Sch96], see also Furnas [Fur86] and Noik [Noi94] which was restricted to two dimensions. Figure 2 shows examples of renditions of an anatomical model of a foot: Figure 2(a) shows a rendition to scale, Figure 2(b) shows a rendition of the same model, except that one bone has been enlarged, while the algorithm adjusts the sizes of all other objects to make them fit in 3D. Note that if only Figure 2(b) is viewed, it is not *à priori* clear which objects have been enlarged and which ones have been made smaller. Thus in Figure 2(c) the scaling factor of each object is mapped onto its transparency in the visualization, and the enlarged object appears to be in the foreground, thus giving the user a full view of the enlarged object of interest and at the same time indicating how and roughly to what extent the objects have been changed for their visualization.

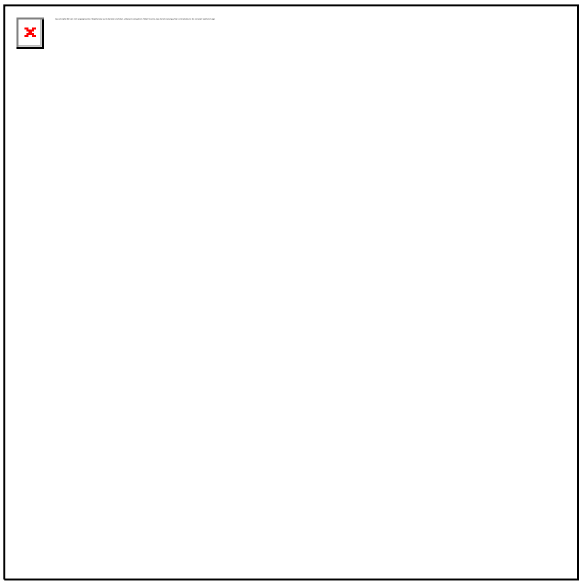

Figure 2: Example of zooming in 3D

Perhaps the most significant aspect of Raab and Rügers method is that they were able to code it as an application independent method called the *pluggable zoom* which can be used on arbitrary data. Indeed, they have developed what they call a *zoom server* which can be accessed via the network to carry out zoom steps (see http://isgwww.cs.uni-magdeburg.de).

4.2 Zoom Navigation

A complex information space may also contain significant amounts of textual information which is normally placed within separate windows. Rüger et al. [RUP98] show how to use the concept of abstraction for navigation in graphical user interfaces. For each object, Rüger et. al. propose the use of what they call an *aspect of interest*, which specifies which information pertaining to the object is of current interest. Resizing the visualization of the object leads to different information which can be displayed, depending on the current aspect. For example, in their user interface to the simulation system Create! [RB95], an enlargement of an icon for a buffer can either lead to the display of a parameter dialog or a list of constant definitions (Figure 3). The resizing of objects is illustrated in Figure 4 and shows how objects are moved about depending on the space available, while the visualization is adjusted to fit the dialog context at hand.

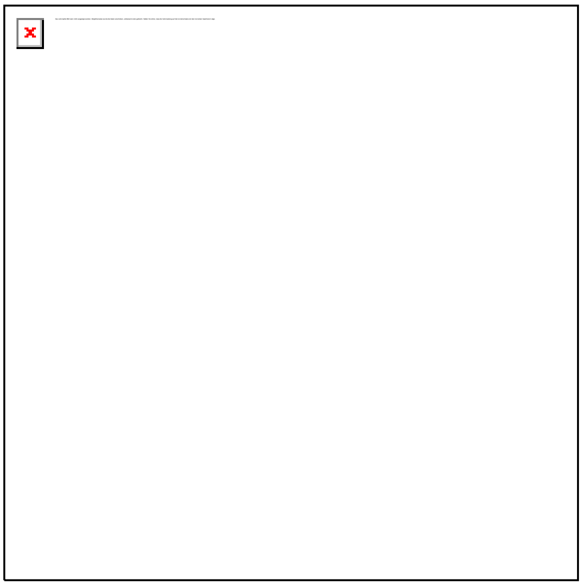

Figure 3: Nodes with activated presentations. Left: a parameter dialog, right: a list of constant definitions is chosen as the most appropriate representation

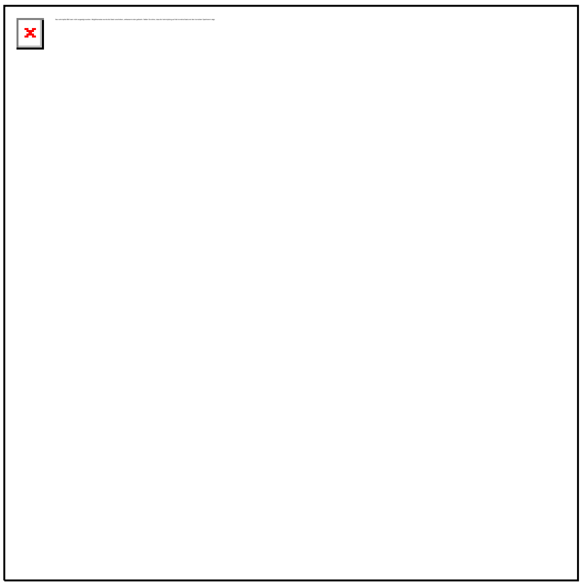

Figure 4: Left: Zoom Navigator with Interval Structure overlayed, Center: One element zoomed to open a dialogue, Right: Different dialogues activated and zoomed

Zoom navigation offers the possibility for the system to gather useful information about the context of the users input. This context information pertains to what information was available to the user at any given point in the interaction. This in turn enables a richer interaction history which can be exploited for example for debugging purposes.

4.3 Animation for Explanations

One of the main purposes of abstraction is to guide the user's attention to important features of an image. An alternative to changing the object's geometry is to use an appropriate camera path in an animation, and to highlight certain aspects of the object by using a pointing device. Experience has shown that great care must be taken with the movements of pointing devices; in particular, when showing the extent of an object, it is often appropriate to approximate the contour; particularly when the details of the contour are not of essence, uneven movements of the pointing device distract from the overall message of the presentation.

Helbing and Preim [HP98] develop an animation system for explaining parts of objects to viewers. A sample of five key frames is shown in Figure 5. Figure 5(a) shows a normal view of an engine; for an explanation of the exhaust pipes, the engine must be rotated into a different position, shown in Figure 5(b) and 5(c). Arrows are used in Figure 5(d) to clarify the shape of the exhaust pipes, and in Figure 5(e) to illustrate their structure.

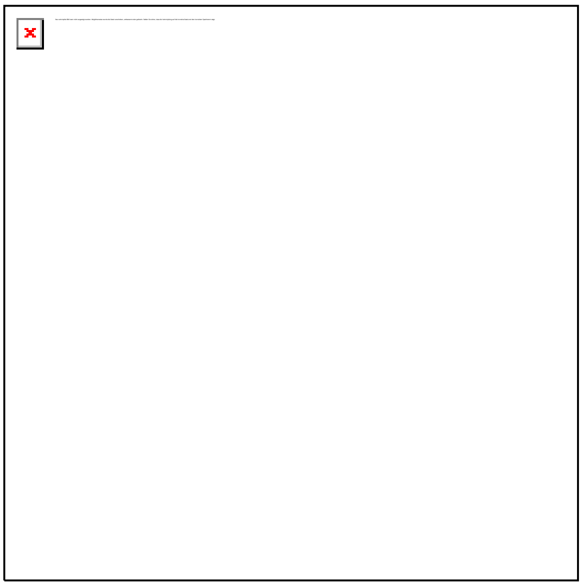

Figure 5: Explanation of an exhaust pipe. In order: original situation, after an automatic rotation to recognize the exhaust pipe, continued rotation, arrows clarify the shape, and structure of the exhaust pipe. Note that only in a color reproduction of the image can the arrows be recognized easily, and that the objects' movements are also decisive to the viewers' interpretation of the images.

5 Concluding Remarks

This paper has introduced some of the fundamental notions introduced and elaborated in [STR98]. The fundamental thesis is that rendering must develop beyond producing images which correspond in all facets to an underlying generic geometric model of an object. A wide variety of applications require more effective illustrations which guide the users' attention, provide only such detail as is necessary for the dialog context at hand. The rendering style is also linked to both the interaction context in interactive systems, and the surrounding text.

5.1 Acknowledgements

The authors wish to thank their colleagues who contributed to the volume on abstraction [STR98] for their collaboration. Thanks also to Petra Janka, Petra Specht and Sylvia Zabel for all the administrative arrangements around the book and this paper.

5.2 References

STR98: Th. Strothotte (ed.) *Abstraction in Interactive Computational Visualization - Exploring Complex Information Spaces,* Springer-Verlag Berlin-Heidelberg-New York, in print

RR98: A. Raab, M. Rüger *Zooming in 1, 2, and 3 Dimensions* in STR98

RPR98: M. Rüger, B. Preim *Zoom Navigation - Exploring Complex Information and Application Spaces,* in STR98

RUP98: M. Rüger, K. Ullrich, I. Pitt *Zoom Navigation in User Interfaces,* in STR98

HP98: R. Helbing, B. Preim *Interaction Facilities and High-Level Support for Animation Design,* in STR98

Fur86: G. W. Furnas *Generalized fisheye views,* in Proceedings of CHI'96, Boston, MA, April 1986, Human Factors in Computing Systems, pages 16-23,1986

Noi94: E. Noik *A space of presentation emphasis techniques for visualizing graphs,* in Proceedings of Computer Graphics '94, Banff, Alberta, 1994, pages 225-233

Sch96 D. Schaffer et. al. *Navigating hierarchically clustered networks through fisheye and full-zoom methods,* ACM Transactions on Computer-Human Interaction, 3(2):162, June 1996

RB95: M. Rüger, Th. Behlau *Create!: An Object-Oriented IDE for Discrete Event Simulation,* in Proceedings of the Winter Simulation Conference '95, Arlington, Washington D.C., December 1995, pages 775-780

Wei89 B. Weidenmann *Informative Bilder (Was sie können, wie man sie didaktisch nutzt und wie man sie nicht verwenden sollte),* Pädagogik, pages 30-34, September 1989