Graphical Abstraction and 3D-Hypergraphics: Exploring large geometrical Models

Antonio Krüger [*]   Christoph Stahl [†]

Abstract

The navigation in complex 3D-models is complicated because of two circumstances. First the huge amount of 3D-data slows down the interaction and second the variety of details is likely to overload the cognitive capabilities of the viewer. In such a situation relevant details are often difficult to find. This paper describes an interactive version of the level-of-detail concept (LOD) to speed up the visualization process and to reduce irrelevant details at the same time.

1 Introduction

Supported by the development of low-cost PC 3D-accelerator cards, navigating in complex virtual 3D-Worlds has become more and more possible during the last years. Still two major problems remain that aggravate a user-friendly interaction with huge amounts of graphical data. On the one hand the complexity of attractive 3D-worlds often overcharges the capabilities of the available hardware platform and the interaction becomes unacceptably slow. On the other hand graphics with too many details is likely to overload the cognitive resources of the viewer. As a consequence the user cannot discriminate relevant from irrelevant details and the danger of distraction increases. Lost in such a way the navigation often has to be restarted from well-known or predefined positions.

In the last years several approaches (e.g. [GH95],[CVM+96] and [Hop96]) have been presented to reduce the computational load on the hardware platform but only little steps have been made to reduce the cognitive load of the viewer of 3D-worlds and thus only some attempts (e.g. [Fei85] and [Krü95]) exist that try to solve this problem.

This paper briefly presents an approach addressing both problems. The detailed 3D-world is structured by an object hierarchy where all inner nodes, that usually do not represent geometry explicitly, have an abstracted geometry associated with it. The higher the inner node is located in the hierarchy, the coarser is the graphical representation. A special interface allows the user to navigate through the different available abstractions of the world.

In order to gain more flexibility we propose to connect a graphical abstraction system to the 3D-Browser (e.g. Cosmoplayer). This step allows the user to request simplifications on demand without respect of the 3D-world's object hierarchy.

2 Related Work

The presented work relies both on standard computer graphics LOD techniques and on previous work that has been carried on in the context of intelligent graphics generation.

2.1 Computer Graphics

The automatic generation of these different LODs is not easy and several suggestions have been made. Some of the approaches remove selected vertices or merge adjacent edges in order to reduce the overall complexity. In [Hop96] Hoppe suggests to cast the problem in terms of minimizing an energy function that captures the conflicting goals of simplification and error minimisation. The result are *Progressive Meshes* that have the nice property of

[*] Graduiertenkolleg Kognitionswissenschaft, Universität des Saarlandes

[†] Fachbereich Informatik, Universität des Saarlandes

a very compact representation and thus help to save computer memory and communication bandwidth. Another method uses *Simplification Envelopes* [CVM+96], in a form of an inner hull and an outer hull of the object. A simplified object can be constructed between those two hulls and the distance between both hulls determines the degree of simplification: increasing distances allow for coarser objects.

A signal processing approach that does not preserve topology is described by Rossignac and Borrel [RB93]. All vertices of the model are filtered with the help of a 3D-grid and all the points that are located in one of the grid's cubes are replaced by a single vertex. As with all sampling algorithms aliasing can arise and high-frequency details that might be important are removed.

Modified versions of this algorithm [BK97] allow to costumize the simplification degree and therefore to generate customized simplifications with varying degrees of detail.

Another, more artistic, field of computer graphics is the development of *expressive* renderers that do not aim at the photorealistic reproduction of scenes. Instead, they use nonphotorealistic techniques to obtain images that resemble sketches [SP95], copper plates [Lei94] or even chalk drawings [LS95].

Both approaches to simplification, multiresolution modelling and expressive rendering, concentrate only on one goal at the time. The first on the reduction of computational expenses and the latter on pleasantly looking graphics.

A position that lies somehow inbetween is presented by Horvitz and Lengyel in [HL97]. The primary goal in this work is still to reduce complexity to save computational resources. But this is done under consideration of the perceptual costs of the simplified results. A probability distribution of the attentional focus of the viewer helps to decide which parts of the graphics may be simplified without the viewer even noticing it.

2.2 Intelligent Graphics Generation

Only a few systems have so far addressed the problem of generating abstract graphics automatically which fulfil a certain communicative goal. A first approach was APEX [Fei85] a system using a precomputed approximation hierarchy to make different simplifications available customizing graphics to a presentation goal. The next step was the dynamic generation of abstractions presented by the system *PROXIMA* [Krü95]. Both systems use a merging operator for primitive objects to generate simplified versions of the graphics.

One major problem with graphical abstraction techniques is, that they sometimes produce artifacts that may confuse the viewer and unintentionally distract the attention from relevant parts. *PROXIMA's* successor $\mathcal{ARP}$, tries to detect such irritations by a gradual generation of graphical abstractions linked with an explicit evaluation component helping to control the progress of the abstraction process. In addition to the above mentioned systems $\mathcal{ARP}$ uses modified *multiresolution* methods (i.e. [RB93]) to filter details of huge 3D-models.

In order to ensure recognizability of the abstracted objects, the system respects *significant attributes* of the domain. This includes salient subparts of the objects, the object's main axis and relevant shape information (i.e. "cylinder-like object").

Starting from a abstraction goal special *attention rules* determine to which areas of the model the attention of the viewer must be guided to fulfil the requested goal. For example, if the abstraction goal is to"identify switch A" of a device then of course the attention of the viewer must be guided to the region where switch A is located. Furthermore if s/he is not familiar with the device s/he needs visual cues where to find the switch. In this case the attention of the viewer should be guided also to relevant landmarks.

Abstraction rules help to parametrize the abstraction techniques to abstract the areas that are not in the focus of interest. An evaluation component uses simple edge-detection techniques to control this process of graphical abstraction and tries to

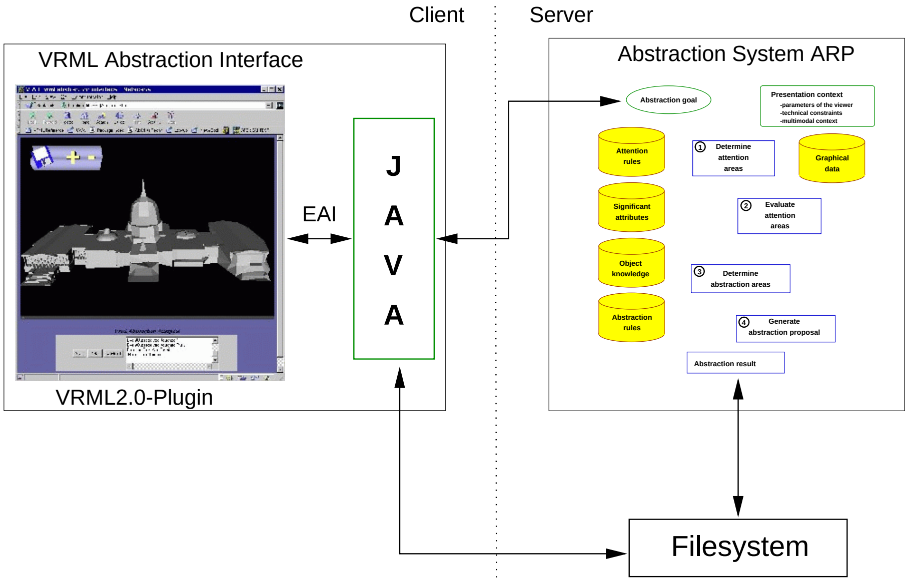

Figure 1: VRML-browser linked to an abstraction system.

detect distracting areas of the image, that are subject for further abstraction. $\mathcal{ARP}$ was originally planned as a subcomponent of DFKI's[1] Intellimedia system *PPP* [WAF+93] where it is used to generate abstracted graphics in the context of interactive explanation of graphical material (e.g. technical documentations and maintenance tasks).

A more elaborated discussion of ARP's functionality is presented in [Krü98].

3 Navigating with the help of graphical abstraction

Our main idea consists in connecting a graphical abstraction system (e.g. $\mathcal{ARP}$) to a 3D-browser (e.g. Cosmoplayer). As figure 1 shows, we achieve this with the help of a JAVA interface and the filesystem. The user can either manipulate directly the model or adjust interface controls to specify the desired degree of graphical abstraction. An abstraction request is passed to the abstraction system that computes the needed simplifications on the fly. The result is saved into a file, read in by the browser and replaces the original geometry. Although a precomputed abstraction hierarchy might be sufficient on a first glance, the dynamic generation is much more flexible and can take into account also the user's navigation goals.

Two different operational modes are actually supported. An automatic abstraction mode and an interactive request for further abstraction.

3.1 Automatic mode

As a default, the most abstracted view is presented at the beginning. At this point the system can - if the goals of the user are known - use these information to customize the simplification degree of the abstraction in an appropriate way. For example, if the user was interested in certain architectural features of a building, i.e. the pillars of the capitol in figure 2A, the abstraction system would try to preserve these details also in the first (highest) presented abstraction. If such information is not available the system aims at providing a good overview over the entire geometric domain with a coarse representation of the model. An automatic update is also imaginable, if the system notices that the user's goals have changed. The most simpliest method of notifying the system of such a new goal is clicking on an object indicating the new interest of the viewer.

[1]German Research Center for artifical Intelligence

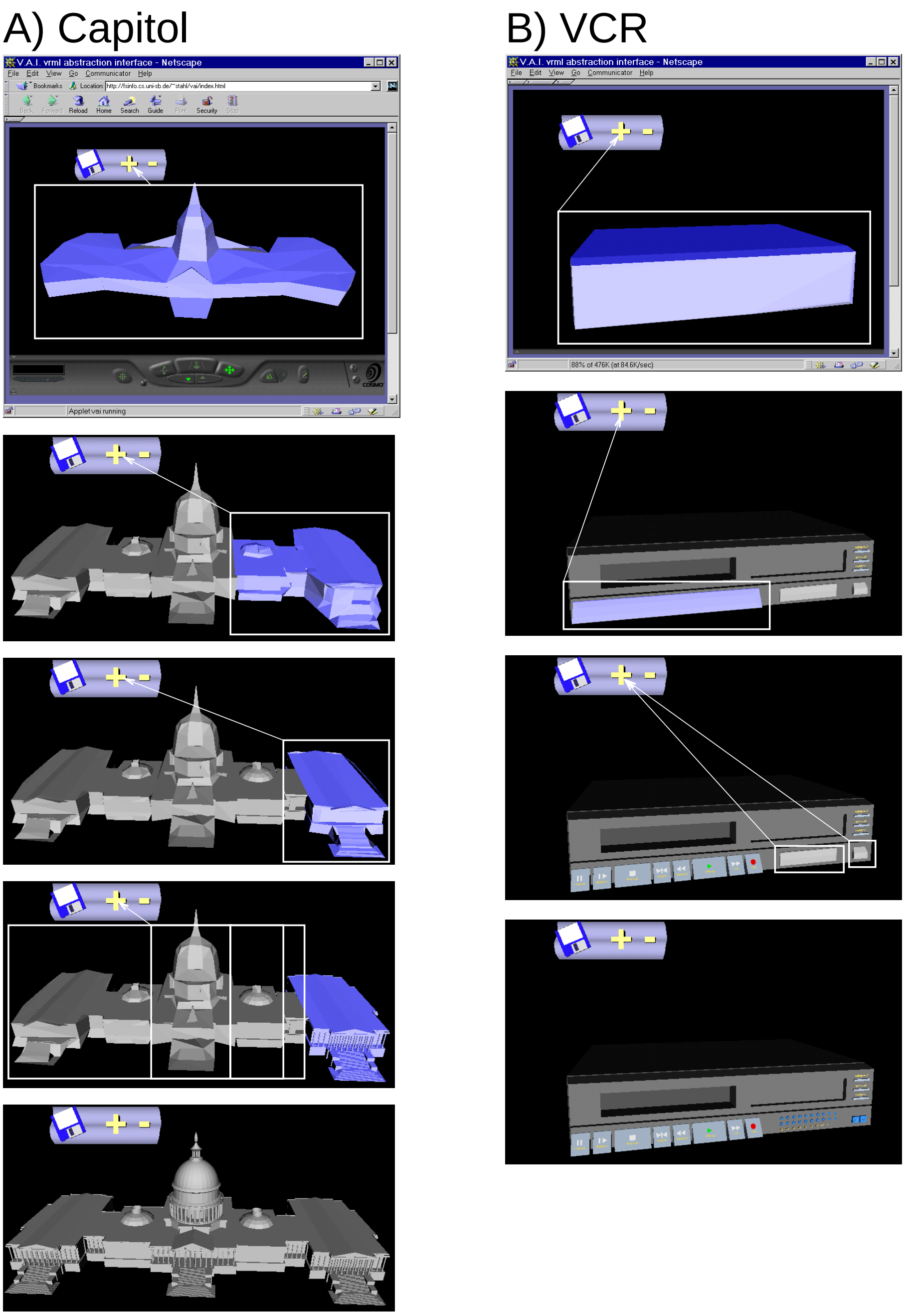

Figure 2: Two Examples: On the left the model of the capitol, on the right the model of a VCR. The white rectangles in the images represent user interaction (clicking with the mouse). The plus and minus on the virtual console in the upper left of the browser window are used to increase/decrease the level of detail.

This leads directly to the second mode of abstraction.

3.2 Interactive mode

The user has always the possibility to express further interest for more details by clicking on the object. According to the object-hierarchy several less abstracted subparts are displayed and recursively the user may reveal more and more details if wanted. In addition if an object is selected it's abstraction degree might be customized.

In order to achieve higher abstraction degrees that do not preserve topology (i.e merging of several buttons to a button bar), object may be grouped and abstracted together. This interaction is done with the help of a virtual console (see figure 2).

4 Examples

Two short examples will illustrate how navigation works with the support of graphical abstraction. Figure 2 shows two models of the capitol and of a VCR.

The details of both models increase from the top to the bottom and the snapshots are taken from a session while a user navigates through the different abstraction degrees.

The white frames indicate which subparts the user has chosen for further investigation. Please notice that at the beginning of both sessions only one subpart is available and that the number is increasing consequently while going down the object's hierarchy. The arrows pointing from the frame to the virtual console reflect the choice of the user to add more details to the model (symbolized by the *plus* on the virtual console). In a similar way backtracking is possible by pressing the *minus* symbol.

Both examples show how a personal abstracted view of the model can be achieved.

5 Further work

The new concept of the navigation in complex 3D-models with the help of graphical abstraction has to be investigated further more. Beside technical problems that still must be solved, it is important to understand which different types of interaction are useful. Therefore several different 3D-domains (e.g. technical devices or buildings) must be tested.

We are preparing also a more general user study, to determine how far graphical abstraction can be used to focus on certain details of a 3D-model. With the help of gaze-tracking equipment we hope to get more insights where graphical abstraction is really useful.

The actual implementation allows for navigation through precomputed abstractions along the object's hierarchy. The next step is to enlarge the functionality of the system enabling the user to ask for dynamic abstractions of selected objects.

A the moment only 2D-input devices are supported. In addition 3D-devices could be used to define viewports that distinguish highly abstracted areas from more detailed regions of the 3D-model.

References

[BK97] BUTZ, ANDREAS und ANTONIO KRÜGER: *Zur Auswahl von Abstraktionsgraden*. In: DEUSSEN, OLIVER und PETER LORENZ (Herausgeber): *Simulation und Animation*, Seiten 147–158, Budapest, 1997. SCS Society for Computer Simulation Europe.

[CVM+96] COHEN, JONATHAN, AMITABH VARSHNEY, DINESH MANOCHA, GREG TURK, HANS WEBER, PAKAJ AGARWAL, FREDERICK BROOKS und WILLIAM WRIGHT: *Simplification Envelopes*. In: *Proceedings of Siggraph'96*, New Orleans, 1996.

[Fei85] FEINER, STEVEN: *Apex: An experiment in the automated creation of pictorial explanations*. IEEE Computer Graphics and Applications, 5(11):117–123, 1985.

[GH95] GARLAND, MICHAEL and PAUL S. HECKBERT: *fast* *polygonal approximation of terrains and height fileds*. Technical Report CMU-CS-95-181, Carnegie Mellon University, Pittsburgh, PA 15213, 1995.

[HL97] HORVITZ, ERIC and JED LENGYEL: *Perception, attention, and resources: A decision-theoretic approach to graphics rendering*. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI'97)*, pages 238 – 249, Providence, RI, USA, 1997.

[Hop96] HOPPE, HUGUES: *Progressive meshes*. In *Proceedings of Siggraph'96*, New Orleans, 1996.

[Krü95] KRÜGER, ANTONIO: *PROXIMA: Ein System zur Generierung graphischer Abstraktionen*. Diplomarbeit, Universität des Saarlandes, Fachbereich für Informatik, Prof. Dr. W. Wahlster, 1995.

[Krü98] KRÜGER, ANTONIO: *Automatic graphical abstraction in intent-based 3d illustrations*. In *Advanced Visual Interfaces '98*, pages 47–55. ACM Press, 1998.

[Lei94] LEISTER, W.: *Computer generated copper-plates*. Computer Graphics Forum, 13(1):69–77, 1994.

[LS95] LANSDOWN, JOHN and SIMON SCHOFIELD: *Expressive rendering: A review of nonphotorealistic techniques*. IEEE Computer Graphics and Applications, pages 29–37, May 1995.

[RB93] ROSSIGNAC, JAREK and PAUL BORREL: *Multi-resolution 3D approximations for rendering complex scenes*. Technical Report RC 17697, IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY 10598, 1993.

[SP95] SCHLECHTWEG, STEFAN and BERNHARD PREIM: *Emphasising in line-drawings*. Norsid-Info, 1/95:9–10, 1995.

[WAF+93] WAHLSTER, W., E. ANDRÉ, W. FINKLER, H.-J. PROFITLICH, and T. RIST: *Plan-based integration of natural language and graphics generation*. In *Artificial Intelligence*, number 63, pages 387–427. Elsvier Science Publisher, 1993.