Analyzing Engineering Simulations in a Virtual Environment

Martin Schulz [†‡], Thomas Reuding [‡], Thomas Ertl [†]

[†] University of Erlangen, Computer Graphics Group (IMMD IX)
Am Weichselgarten 9, D-91058 Erlangen, Germany
[‡]BMW AG, Entwicklung Karosserie,
80788 Munich, Germany
Email: schulz@informatik.uni-erlangen.de, thomas.reuding@bmw.de

CR Descriptors: Virtual Environments, Virtual Reality, VRML, Scientific Visualization, Numerical Simulation

Abstract

The use of virtual prototypes generated from engineering simulations can be crucial to the efficient development of innovative products. Performance predictions and functional evaluations of a design are possible long before results of real prototype tests are available. In this article we present new results of on-going research at the University of Erlangen and at *BMW* in developing a virtual environment for carbody engineering applications. They extend to a variety of different fields such as structural mechanics, aerodynamics, acoustics and air conditioning, to name just a few. In each of these fields, the task at hand is twofold: first to predict the physical behavior of a complex technical system; second to use the analysis results to effectively introduce the necessary engineering changes. Creating *VRML* scenes makes it possible to use the Intra-/Internet as a communication platform.

1 Motivation for a Virtual Environment

The concept phase of product development in automotive design is characterized by the need to evaluate complex engineering scenarios under conditions in which the relevant information is either only partially available or the correctness of underlying assumptions regarding the viability of certain technical solutions is not yet proven.

Numerical simulation has always played a key role in this context. Performance predictions and functional evaluations are possible long before results of real prototype tests are available. However, with the rise in model complexity, data quantity, computing performance and accuracy we increasingly find ourselves lacking the tools, methods and metaphors to deal with the information that is being generated. This is supported by the observation at *BMW* that currently about 30 percent of the efforts involved in a typical simulation go into the preprocessing phase and about 10 percent is taken up by the actual computation, while approximately 60 percent goes into the analysis and communication of the results. Clearly there is a strong incentive to reduce the last percentage by means of implementing insightful, intuitive visualization tools which allow effective communication between engineers.

Previous research has shown that a virtual environment can offer a wide variety of analytical postprocessing tools. For example, the *Virtual Windtunnel* project described by Bryson [1, 2] is one of the first applications based on virtual reality techniques that clearly shows the advantages of this approach compared to traditional finite element analysis postprocessing methods. Ye [3] and Yeh [4] also use a virtual environment for the visualization of finite element models which further illustrates the usefulness of this technique.

The study of realistically simulated scenarios in structural and fluid mechanics involves very large, transient data sets. In the past it was far beyond the available hardware capabilities to display those data sets in real-time. However, due to advances in hardware technologies and due to the efforts of a number of researchers in the area of data reduction techniques [5, 6], the visualization of these data sets became possible.

In earlier research by the University of Erlangen and *BMW* we focused on the analysis of results from a finite element solver and performed a time-dependent real-time visualization of the crash performance of a vehicle. This proved the value of providing novel computer-human interface techniques for intuitive and interactive analysis of crash test simulation data [7, 8, 9]. However, several weaknesses were identified: deficiencies in interactive performance, lack of versatility in the configuration of the virtual environment and limited extensibility to new capabilities.

2 System Design Issues

Our system loads finite element models directly which consist of various types of elements (volumes, shells and beams), which are collected into groups. All of the data is time-dependent and comprises, for

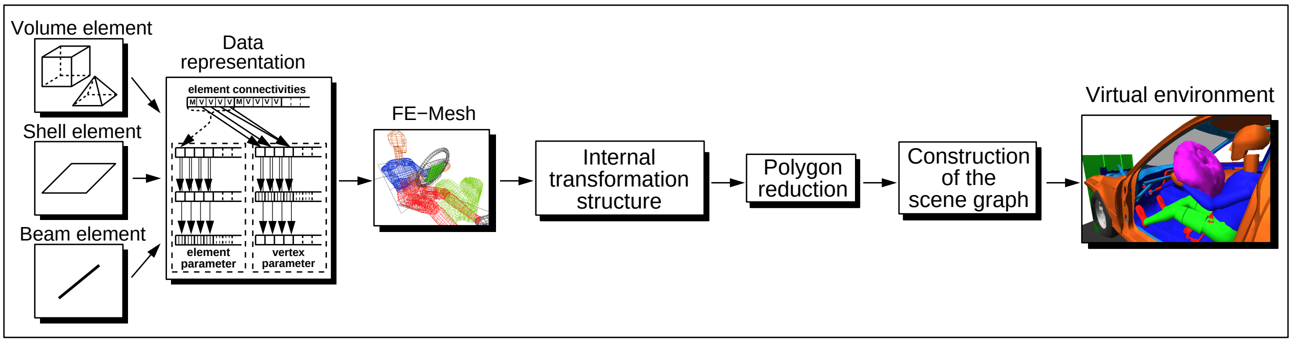


Figure 1: Transformation of the finite element model into the virtual environment.

each time step of the simulation, nodes in global coordinates and elements which reference the groups they belong to as well as their constituent nodes (Figure 1). Our system follows an object-oriented design with the data structured into a class hierarchy which is partly derived from the element structure of the finite element models themselves. Furthermore, it employs data sorting methods to generate new local polygon lists and creates a data structure suitable for the animation of all time steps.

2.1 Polygon Generation and Reduction

Shell and beam elements are converted into polygons in a straightforward manner. In the case of several solids representing a car component, an algorithm was designed that eliminates identical polygons of the adjacent solids that build up the component. For all vehicle components originally modelled as solids, only those polygons which define the outer surface are finally retained.

In order to meet memory requirements and to maintain frame rates of at least ten frames per second, which is necessary to provide a sense of real time interactivity, the polygon mesh of the model needs to be simplified. Since it is important to keep the shape of the model consistent over time, the simplification algorithm is applied to all time steps, identifying and preserving those vertices which carry information which is relevant to the simulation. Figure 2 shows how this technique reduces the mesh of an engine mount in unimportant regions while maintaining a fine mesh in the area of deformation (highlighted). Our method is based on the algorithm of Schroeder [5] and does not create new vertices. This is essential since we need to preserve scalar and vector-valued data computed in the finite element analysis for each vertex of the original set. Typically reductions in the number of polygons on the order of 50 percent are achieved.

2.2 Adapted Virtual Environments

Currently our system supports a variety of interface hardware and interactive functions and we are in the process of addressing new fields of application. We demonstrate the improved capabilities of our virtual environment for interactive “what-if” studies - giving multidisciplinary engineering teams an intuitive sense of the performance of a design as well as facilitating the assessment of alternatives and helping to achieve optimal solutions to conceptual design problems.

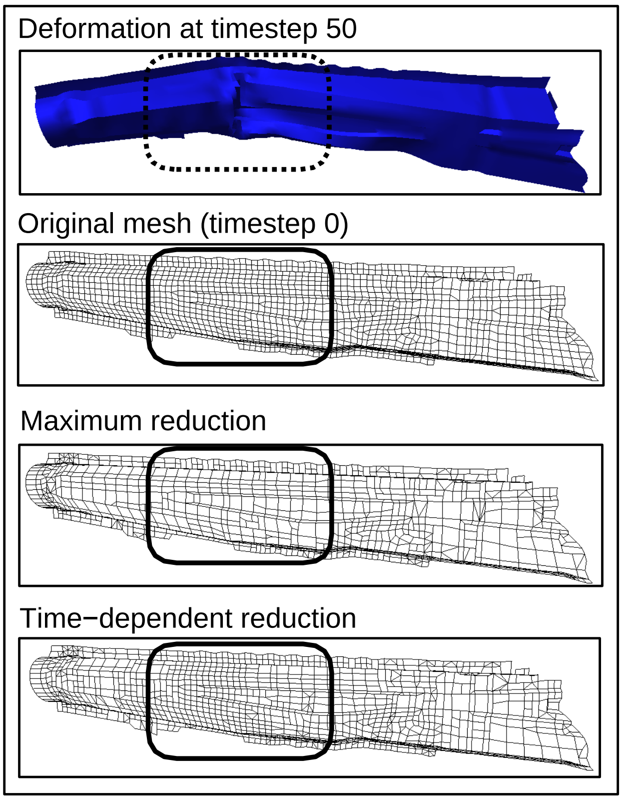


Figure 2: Polygon decimation of the engine mount.

Our system is divided into a high-end VR part and a *VRML* based part for Intra-/Internet access (Figure 3). We think of virtual environments being tailored to a specific application in order to become Adapted Virtual Environments (AVE) [10].

The high-end part includes a VR-lab, the workplace environment and the projection environment. This part of the system also contains methods for converting finite elements into geometry information and for creating *VRML* files of selected objects. The devices that are used in a particular user session are specified at start-up. Several interface hardware devices are

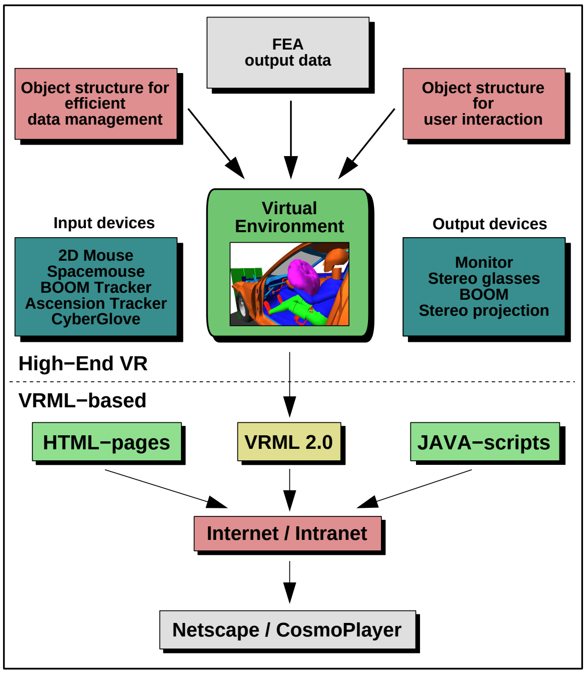


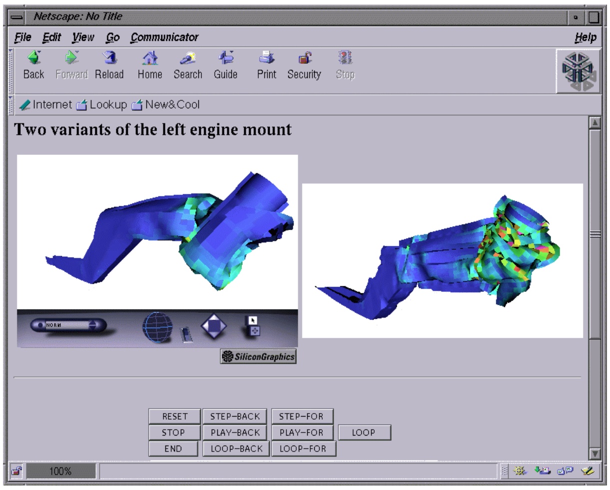
Figure 3: Structure of the visualization system

currently supported, including a *FakeSpace BOOM*, stereoscopic large-screen projection as well as the conventional workstation display. In addition, the user can choose between a *CyberGlove*, a 3D mouse or a conventional 2D mouse and a keyboard for input.

This flexibility supports different usage scenarios - from two or three engineers discussing details up to design-build teams of ten people engaged in a project meeting. A simple 3D menu is used to control the visualization. The animation may be stopped and run forward or backward at a user-defined rate in incremental steps.

From the comments of users we learned that stereoscopic viewing is a valuable feature in all environments. The best results were achieved with a two-channel, large screen projection, where the viewers wore simple polarized glasses that do not require a transmitter and which allow the individuals to move freely around in front of the screen. However, we found test users somewhat apprehensive of the 3D mouse as a navigational tool and work is continuing in this area to provide the end user with an intuitive, comfortable means of traversing the virtual space.

Since not every application needs the full bandwidth of immersion, *VRML* files can be used for documentation and communication. The benefits in the second part lies in the link between the expensive high-end environment necessary for the scientific engineering computations and the widespread available low-cost workstations and PC's. Embedding the created *VRML* scenes in *HTML* pages and *JAVA* scripts, which control the visualization in the web browser, propagates the analyzed results to each involved engineer. Two synchronized *CosmoPlayers* within a *Netscape Navigator* allow the direct comparison of two different design variants (Figure 4).



Two variants of the left engine mount

Figure 4: Visualization of two different design variants in two synchronized *CosmoPlayers* within *Netscape*

3 Dynamic Analysis of Numerical Simulations

Our virtual environment for car-body engineering applications supports visualization methods for the exploration of different time-dependent phenomena. Crash test, sheet metal forming and vibration analysis are based on deforming geometric elements (solids, shells, beams), acoustic radiation is based on unstructured scalar volume elements.

3.1 Impact Dynamics

The crash test simulation results are calculated by the non-linear, transient, dynamic finite element solver *PAM-CRASH*. A single crash test data set consists of about sixty discrete time steps each of which contains about 400,000 finite elements representing the geometry and a set of scalar values for each element. Geometry data and physical properties data like stress, strain, acceleration or velocity are visualized to illustrate the performance of the whole vehicle or any subset of components (Figure 5).

3.2 Sheet Metal Forming

The simulation of sheet metal forming is used to predict the distribution of residual stress and the distribution of thickness after the stamping. Furthermore the propagation of ripples during stamping as well as their final position is obtained. As in the case of crash simulation, geometry together with physical properties is displayed (Figure 6). The main difference lies in the complexity of the grid. The tailored blank is made up of a time-dependent hierarchically refined grid with new polygons generated in every time step. The geometry of the die and the blank-holder surfaces can be shared through all time steps by applying a suitable translation matrix to reduce the memory requirements. In the virtual environment we can create

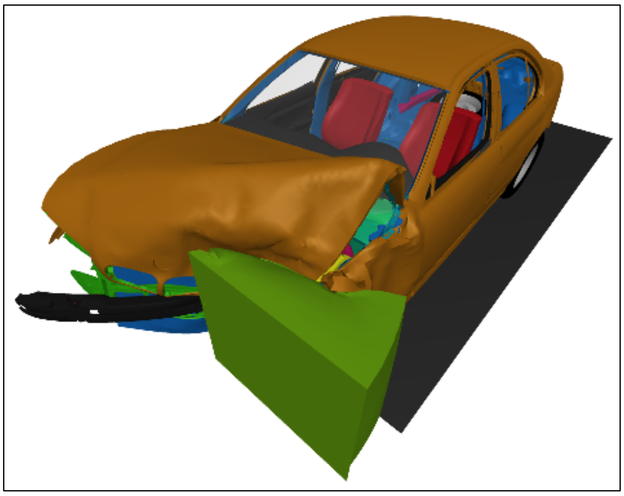


Figure 5: Full car crash in the virtual environment.

a transparent press tool and offer dynamic insight into the forming of complex geometries.

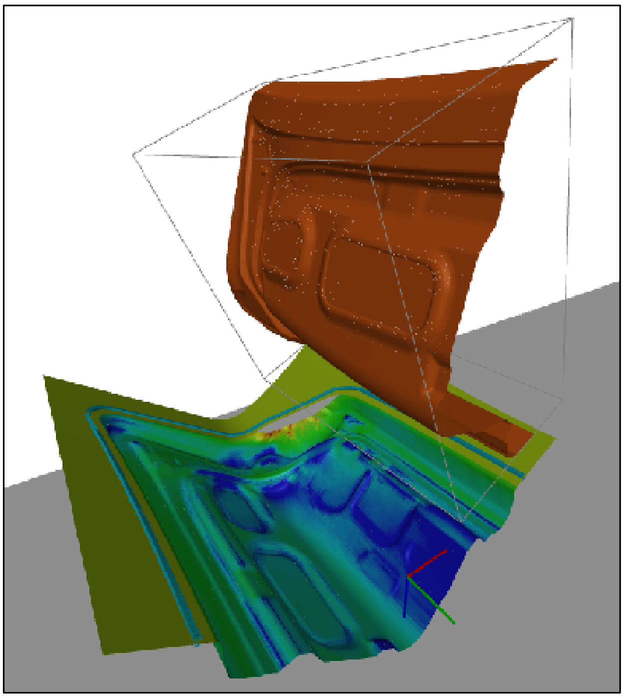


Figure 6: Sheet metal forming: the upper forming tool has been moved

3.3 Vibration

Vibrational analysis is computed as a linear modal eigenvalue simulation of the car-body structure. The car structure consists of about 200,000 finite elements. As opposed to the crash test simulation there is only one complete geometry description which contains the nodes and the element connectivities. Over the course of the animation the vehicle vibrates with a user-defined amplification factor and the engineer can analyze the simulation more clearly than by interpreting complex displacement vectors (Figure 7).

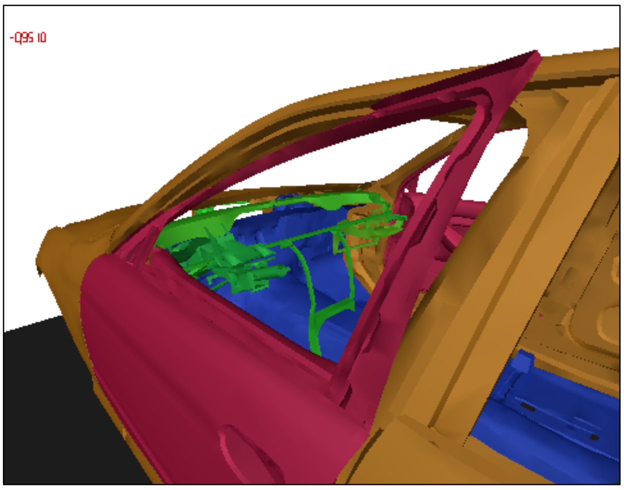


Figure 7: The door vibrates with the user defined amplification factor.

3.4 Acoustics

The noise level (or preferably the absence thereof) in the passenger compartment is a feature of relatively high customer value because it is easily noticeable. This noise occurs when the volume of air that occupies the passenger compartment is subject to a change in pressure caused by the excitation of resonating components (Figure 8). The acoustics simulation is calculated on basis of the vibration analysis results.

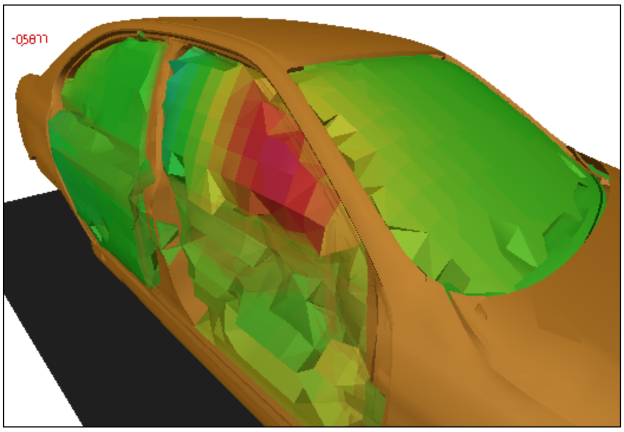


Figure 8: The air pressure volume inside the passenger compartment is colored by the scalar values.

Typically the volume of air consists of about 20,000 volume cells and as a component begins to vibrate a pressure wave is originating from it. An intuitive means of visualizing the corresponding noise level behavior is the calculation of an isosurface from the air pressure for every time step. The source of the change in air pressure and the resulting noise level can be localized and the propagation of the wavefront can be traced through the passenger compartment. Since our input volume consists of a variety of cell types (tetrahedra, pyramids, prisms, and hexahedra), the standard marching cubes algorithm [11] for computing isosurfaces has to be extended. We partition all volume cells into tetrahedra and apply a marching tetrahedra type algorithm which also avoids ambiguities in the surfaces.

4 Interactions

Our virtual environment offers a wide variety of different interaction mechanisms. Standard operations like picking, dragging, and changing the transparency of the components are realized and the animation can be controlled by the 3D menu. Furthermore, scalar values calculated by the simulation are displayed either as color maps on the picked geometry or as a function of time and location drawn on a virtual sketch pad appearing in the user's peripheral field of view (Figure 9).

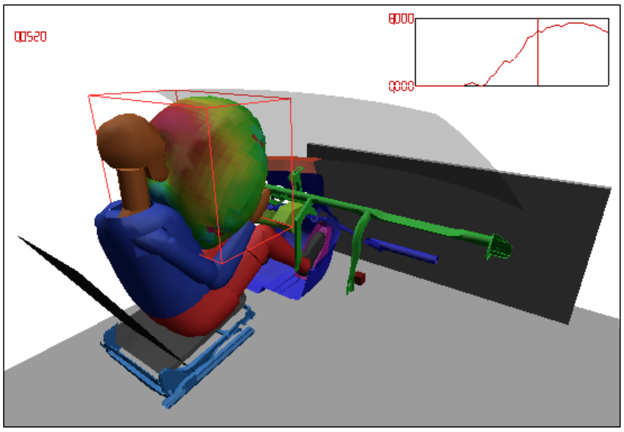


Figure 9: Display of a scalar value at one selected element as a function of time on the sketch pad.

Another interactive feature is a cutting plane which can be moved freely in virtual space and which slices through virtual objects (Figure 10). Two methods of calculating a cut are used: position-oriented (Euler) or geometry-oriented (Lagrange). The computation of the Euler cut involves performing polygon intersection tests for the geometry for each time step. Computationally, the Lagrange cut is the better solution because the desired object is cut in only one time step and the resulting intersection vertices are then transfered to the other steps, giving the cut geometry the appearance of truly deforming during the simulation. Both methods compute a time dependent cutting plane, which is visible throughout the animation.

Since the data varies in time, the visualization will, in general, be in motion. In the case of impact visualization moving objects can be difficult to analyze, and users must have the opportunity to lock onto what they want to see. We provide the ability to attach oneself to arbitrary objects in the scene, thereby observing only relative motion. This can provide the user with, for instance, the crash test dummy's perspective of driving the vehicle. In addition, individual objects can be selected and moved freely during the animation. In the case of a side impact the engineers are interested in the depth of intrusion caused by the impacting vehicle. A plane can be attached to any point of the original shape of the vehicle. For each time step the depth of intrusion is calculated in reference to this plane and mapped as colors onto the component's surface (Figure 11).

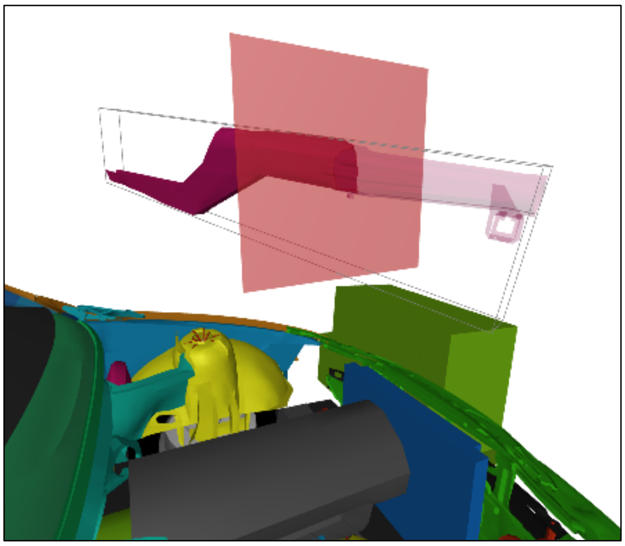


Figure 10: Cutting the engine mount with the transparent plane.

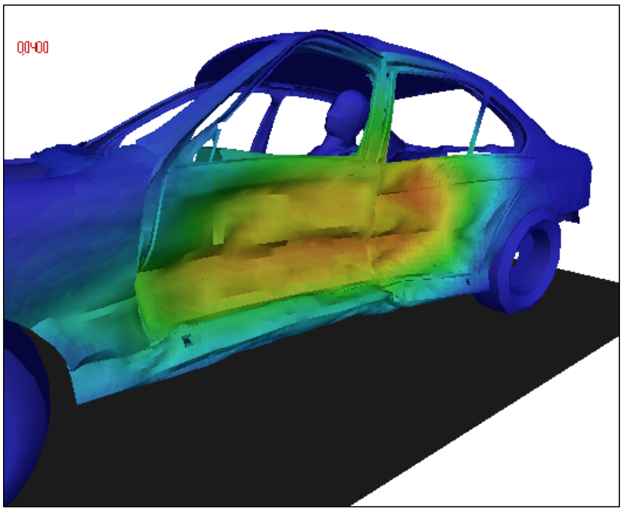


Figure 11: Side impact: the depth of intrusion calculated in reference to an interactive plane colors the geometry.

In order to analyze the acoustic volume we provide a dynamic pressure sensor, either in the shape of a plane or a bar. The sensor can be moved inside the volume to sample and display pressure values at a given location changing with time (Figure 13). While the sensor just visualizes local information next to the probe, do isosurfaces give global information of the volume. A dynamically positionable and propagating isosurface

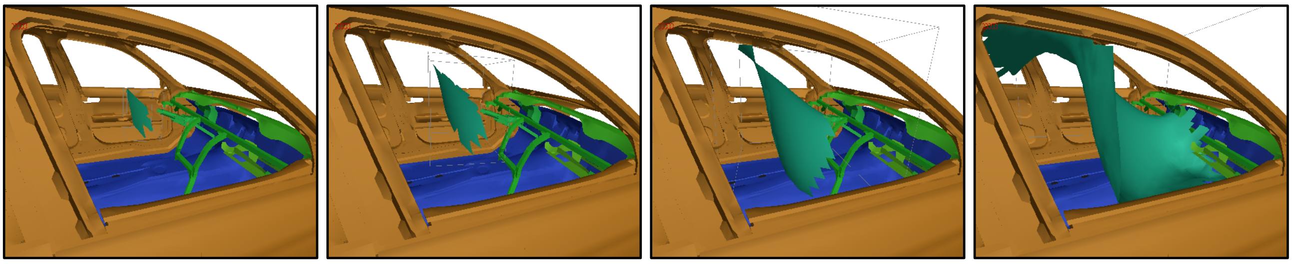


Figure 12: The interactive isosurface probe propagates through the volume.

is available to explore the entire volume. From the given starting point of the probe the isosurface propagates through the passenger compartment (Figure 12).

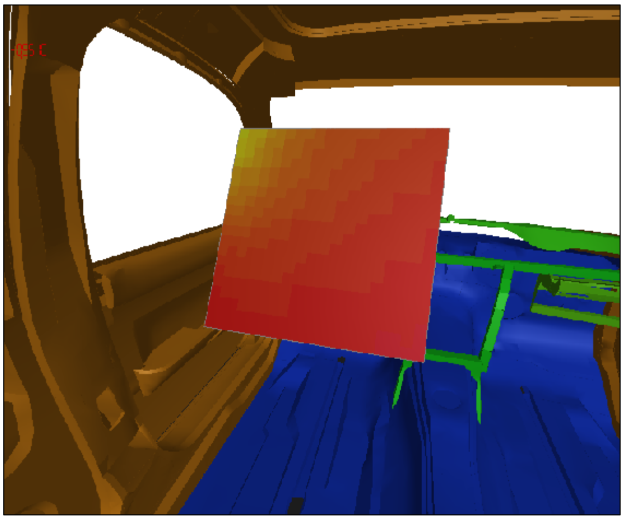


Figure 13: The plane probe explores the scalar volume inside the passenger compartment.

5 Conclusions

We presented a virtual environment for car-body engineering applications which provides insightful, intuitive visualization of virtual prototypes and allows effective communication between engineers and management. Our system is being used at *BMW* for the analysis of a wide variety of time-dependent numerical simulations, ranging from crash-worthiness to sheet metal forming, vibrations and acoustics. The main advantage of the virtual environment is the intuitive navigation and interaction with different types of FE models relevant to the car body development process chain. The option to generate *VRML* files allows low cost workstations and the Internet to be used. This opens the door for widespread communication networks to be established, a highly desirable feature within a collaborative and multi-disciplined simulation environment.

One key feature of our system is, that we are able to load simulation files into the virtual environment directly and efficiently without any special preprocessing and with on-the-fly polygon reduction. An average model consisting of 200,000 finite elements for each of up to 60 time steps can be visualized with at least 10 frames per second using a one-processor *SGI ONYX Infinite Reality* with 2 GB of main memory. Culling to the viewing frustum increases the frame rate to about 20. Larger groups of engineers prefer a large-screen stereoscopic projection for design reviews while the fully immersive environment using a *Fakespace BOOM* is optimal for detailed evaluation of data involving only one to three individuals.

In the future, we will continue working on the integration of different simulation types into the system, more intuitive interaction metaphors (GUI's) and scalability of the system to run on midrange computing hardware. Furthermore, texture mapping and iconic techniques such as "force tubes" [12, 13] are also being explored as improved means for visualizing scalar and vector quantities.

6 Acknowledgments

Many of the initial ideas elaborated in this paper were inspired by fruitful discussions with Sven Kuschfeldt, Daniel Heiserer and Michael Holzner of *BMW*. We would like to thank Edith Zimmermann and Manfred Weiler of the University of Erlangen for their contribution to the software development and to the preparation of this manuscript. Conceptual inspiration came from the work of Steve Bryson et al. on the *Virtual Windtunnel* project at *NASA Ames Research Center*. Finally, we gratefully acknowledge the continuing support of the University of Erlangen and *BMW*.

References

[1] Steve Bryson and Steven Feiner. Virtual Environments in Scientific Visualization. In *Virtual Reality for Visualization, Course Notes of Tutorial 5 at Visualization 95*, 1995.

[2] S. Bryson and C. Lewit. The Virtual Windtunnel. *IEEE Computer Graphics and Applications*, 12(4):25–34, 1992.

[3] W. Ye and J.M. Vance. Visualization of Structural Impact Problems in a Virtual Environment. In *Proc. SCS Simulation MultiConference, Atlanta, GA.*, pages 325–330, 1997.

[4] T.P. Yeh and J.M. Vance. Combining Sensitivity Methods, Finite Element Analysis, and Free-

Form Deformation to Facilitate Structural Shape Design in a Virtual Environment. In *Proc. 23rd ASME Design Automation Conference, Sacramento, CA.*, 1997.

[5] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of Triangle Meshes. *Computer Graphics (Proc. of ACM SIGGRAPH 92)*, pages 65–69, 1992.

[6] K.J. Renze and J.H. Oliver. Generalized Surface and Volume Decimation for Unstructured Tesselated Domains. In *Proc. IEEE Conference VRAIS'96, Santa Clara, CA.*, pages 111–121, 1996.

[7] S. Kuschfeldt, M. Schulz, T. Reuding, T. Ertl, and M. Holzner. Visualization of Crashworthiness Simulations Using Virtual Reality Techniques. In *Proc. International Conference on High Performance Computing in Automotive Design, Engineering and Manufacturing, Paris*, pages 455–462, 1996.

[8] S. Kuschfeldt, M. Schulz, Th. Ertl, Th. Reuding, and M. Holzner. The Use of a Virtual Environment for FE Analysis of Vehicle Crash Worthiness. In *Proc. IEEE Conference VRAIS'97, Albuquerque, NM*, page 209, 1997.

[9] M. Schulz, Th. Reuding, and Th. Ertl. Crashing in Cyberspace - Evaluating Structural Behaviour of Car Bodies in a Virtual Environment. In *Proc. IEEE Conference VRAIS'98, Atlanta, GA*, pages 160–166, 1998.

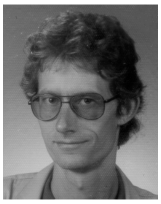
[10] M. Schulz, E. Zimmermann, Th. Reuding, and Th. Ertl. From High-End VR to PC-based VRML Viewing: Supporting the Car Body Development Process by Adapted Virtual Environments. In *Proc. CGIM'98, IASTED, Halifax, Canada*, pages 231–234, 1998.

[11] W.E. Lorensen and H.E. Cline. Marching Cubes: A High Resolution 3D Surface Construction Algorithm. *Computer Graphics (Proc. of ACM SIGGRAPH 87)*, 21(4):163–169, 1987.

[12] S. Kuschfeldt, Th. Ertl, and M. Holzner. Case Study: Efficient Visualization of Physical and Structural Properties in Crash-Worthiness Simulations. In *Proc. IEEE Conference Visualization'97, p. 487-490, Phoenix, AZ*, 1997.

[13] S. Kuschfeldt, M. Holzner, O. Sommer, and Th. Ertl. Efficient Visualization of Crash-Worthiness Simulations. *IEEE Computer Graphics and Applications*, 18(4):60–65, 1998.

Martin Schulz is a PhD student in the computer graphics group of the University of Erlangen, Germany where his research interests are in the field of interactive scientific visualization of finite element and fluid dynamic simulations. His work is supported by BMW through a dissertation scholarship. He received a “Diploma” in computer science from the University of Erlangen in 1996.



Thomas Reuding is a project manager and senior research engineer in the car body development section of BMW's Research and Development Center in Munich, Germany. He is currently working on visualization as well as virtual environment application relevant to car body engineering. He received a “Diploma” in naval architecture and marine engineering from Technical University of Berlin in 1983 and he was a visiting scholar at University of Michigan in 1984. He is a member of the SIAM Activity Group on Geometric Design.

Thomas Ertl is a professor of computer graphics and visualization in the computer science department of the University of Erlangen where he leads the scientific visualization group. His research interests include volume rendering, flow visualization, multi-resolution analysis, parallel and hardware accelerated graphics, large datasets and interactive steering. He received a MS in computer science from the University of Colorado at Boulder, and a PhD in theoretical astrophysics from the University of Tübingen, Germany.