

Advanced Visualization of Crashworthiness Simulations using Virtual Reality Techniques

Sven Kuschfeldt^{†‡}, Martin Schulz^{†‡}, Thomas Reuding[†],
Michael Holzner[†], Thomas Ertl[‡]

[†]BMW AG

[‡]University of Erlangen, Computer Graphics Group

Corresponding author:

Sven Kuschfeldt
kuschfel@informatik.uni-erlangen.de
+49 9131 85-9923

Key words:

Crash Simulation, Finite Element Analysis, Interactive Real-Time
Visualization, Polygon Reduction, Virtual Reality

Abstract

Regarding the increasing complexity of computer simulations on one hand and the development of computer technology in the last few years on the other hand, the demand on new visualization techniques to analyze the immense amount of data originating from scientific calculations has reached new dimensions. In the case of crashworthiness simulations, expanding the effectiveness of traditional post processing techniques is one key to achieve shorter design cycles and faster time to market. A very promising approach, that benefits from the recent progress in software and hardware technology, is the use of virtual reality techniques. At BMW, in co-operation with the University of Erlangen, we developed the software system **VtCrash**, that performs a three dimensional visualization and allows the user to observe, enter and manipulate the visualized crash in a virtual environment. A key feature of **VtCrash** is the possibility, not only to look at the simulation, but also to immerse into a scene that contains a true scaled virtual crash while using stereoscopic viewing technology. **VtCrash** takes the simulation output of the dynamic finite element solver PAM-CRASH and converts it into data structures suitable for the visualization using efficient data reduction and data ordering mechanisms. It creates a virtual scene, performs real time animations, and enables the user to interact with the visualized crash data using a data glove and a Binocular Omni-Oriented Monitor. Additionally, there is the possibility to use **VtCrash** with stereoscopic projection technology. The design and implementation of the system is presented in the paper.

INTRODUCTION

Numerical finite element simulations for automotive engineering applications have become increasingly complex as well as reliable and precise. The behaviour of a car body in frontal, side or heck impact collision scenarios can be simulated and predicted with such high degree of accuracy and confidence that computer simulation results have become a standard evaluation tool in the vehicle development process.

The analytical process in the development of prototype vehicles uses finite element analysis with the primary goal to do as much development as possible with analytical tools and to minimize hardware-prototype testing. At BMW, the nonlinear, transient, dynamic finite element code PAM-CRASH is used to simulate the crash behaviour of the total vehicle.

In the case of vehicle crash worthiness simulations the effectiveness of traditional FEA postprocessing methods can be significantly improved through use of interactive real-time visualization techniques. In this paper we outline the design and implementation of a virtual crash testing environment that provides an immersive and interactive analysis of crash simulations.

The paper is structured as follows:

Section 1 outlines the current status of visualization in crash simulation. Section 2 discusses the demands on new efficient visualization systems. Section 3 describes the design of our virtual environment **VtCrash**. Section 4 shows the effectiveness of **VtCrash** with respect to the application area. Finally, in section 5, we present our conclusions and fields of interest for further investigations.

STATUS OF VISUALIZATION IN CRASH SIMULATION

The crash simulation procedure can be divided into three main stages:

1. Preprocessing. During this stage the finite element models are created from the CAD data representing the car components.
2. Analysis. The shape and physical properties of the models are calculated for several time steps of the simulated crash.

3. Postprocessing. The calculated models are investigated and decisions are made about improvements of the CAD models to get the desired crash behaviour.

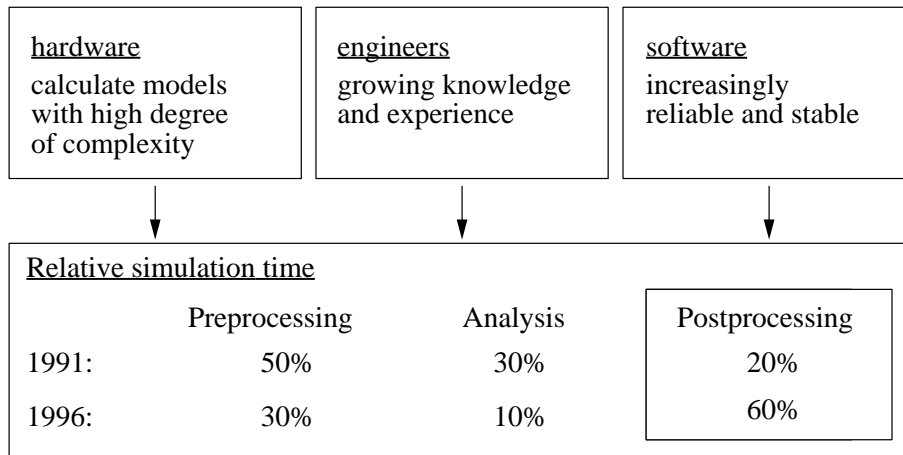


Figure 1: Development of relative simulation time

Today, the hardware is able to calculate models with a high degree of complexity. Regarding the growing knowledge and experience of the simulation engineers and the increasing reliable and stable calculation software the absolute simulation time for modelling, computing and investigating a complete crash model could be decreased. Another important result of the development mentioned above is an adjournment of the proportions between preprocessing, analysis and postprocessing. As shown in Figure 1 the postprocessing stage turned out to become the most time consuming activity performed by the simulation engineers.

During the last ten years, the number of elements contained in a finite element crash model has reached a new dimension. Whereas in 1986 a model consisted of 3000 elements, crash models today comprise more than 200000 elements. Between 1986 and 1991 the visualization techniques of crash simulations have been improved significantly by polygonal representation of the finite element models and implementation of new lighting techniques. During the last 5 years the postprocessor systems exhibit more features and new techniques have been implemented to use additional video support [KHE95]. Much research activities were performed in developing new visualization techniques in different areas such as CFD analysis [BL92] as well as digital prototyping [DG94]. However, in the case of crashworthiness simulations the basic philosophy behind the visualization remained the same. The simulation engineer watches the visualized crash model from outside. An immersion into the model is not possible. Much effort must be undertaken to separate and investigate parts of the structure.

One of the most important problems is the complexity of traditional postprocessing systems and the non-intuitive interfaces which make it impossible to achieve a sufficient investigation of the calculated models without spending a lot of time getting used to the system. Another penalty is the incapability to provide a three dimensional visualization of the simulation.

The changes in the partitioning of relative simulation time, the disadvantages of traditional postprocessors and the development of computer graphics technology in the last few years have resulted in the need for new visualization techniques to facilitate the analysis of crashworthiness simulations. On one hand, there is a high potential to optimize the product development process by saving engineering time. On the other hand, the quality of postprocessing will be improved significantly by implementing an efficient and advanced visualization system.

DEMANDS ON NEW VISUALIZATION TECHNIQUES

The idea of a virtual crash test environment is to allow the user to enter a three dimensional scene that contains geometry data of the vehicle plus the results of all stages of a crash test. The experience of a realistic simulated crash event presumes the meeting of three basic requirements:

Three dimensionality.

In order to achieve spatial perception and motion it is necessary to support a human interface that provides a stereo view of the crash. Such stereo viewing technology can be divided into immersive (i.e. Binocular Omni-Orientation Monitor) and non-immersive devices (i.e. stereo projection, shutter glasses). Another requirement is to keep the response time of the system with respect to actions of the user lower than 0.1 seconds. This ensures an accurate motion and modification of the viewpoint within the virtual scene [BF95].

Intuitive interaction.

There are several interaction features that could be of interest for crash simulation. Structural parts of the vehicle could be separated for investigating details. Car components could be made semi transparent or invisible. The user could use virtual tools like a cutting plane and slice through the vehicle. Because these interactions should be performed in a three dimensional environment it is necessary to support devices which provide three dimensional input. This is important to avoid the problem of mapping the two degrees of freedom of the traditional mouse to the three dimensional control and display interface. Most popular three dimensional input devices are space mouse and data glove. In order to select between the interaction features a tool is required which can be affected by the chosen input device. A three dimensional menu which is part of the virtual environment meets this requirement. The combination of BOOM, data glove and menu or space mouse respectively provide the capability of interaction being simple enough to be denoted as intuitive.

Real time animation.

Because the visualization of crash simulations deals with time varying data sets, the achieved animation rate is very important. Although a high animation rate showing the crash test is desirable, a steady and regular motion is more important than a smooth animation. However, the animation rate must be high enough to prevent the behaviour of perceiving a series of still pictures rather than a continuous evolution or movement. Experience has shown that most users consider an animation rate of 10 frames per second as adequately [BF95].

DESIGN OF THE VISUALIZATION SYSTEM VTCRASH

Description of the application area

The study of realistically simulated crash scenarios involves very large data sets. After triangulation, FE models typically consist of 60 time steps with about 400.000 triangles each. Our system, *VtCrash* provides novel computer-human interface techniques for intuitive and interactive analysis of large amounts of crash test simulation data. *VtCrash* takes geometry data and physical properties data like stress, strain, acceleration or velocity as input and performs real time animations of the vehicle's behavior in a collision. The user can enter the virtual environment, directly manipulate the simulation, as well as examine and interact with any part of the vehicle.

Relevant system implementation issues

Data structuring and system design. Currently, the system uses Sense8's WorldToolkit as graphics interface. WorldToolkit is based on OpenGL. *VtCrash*

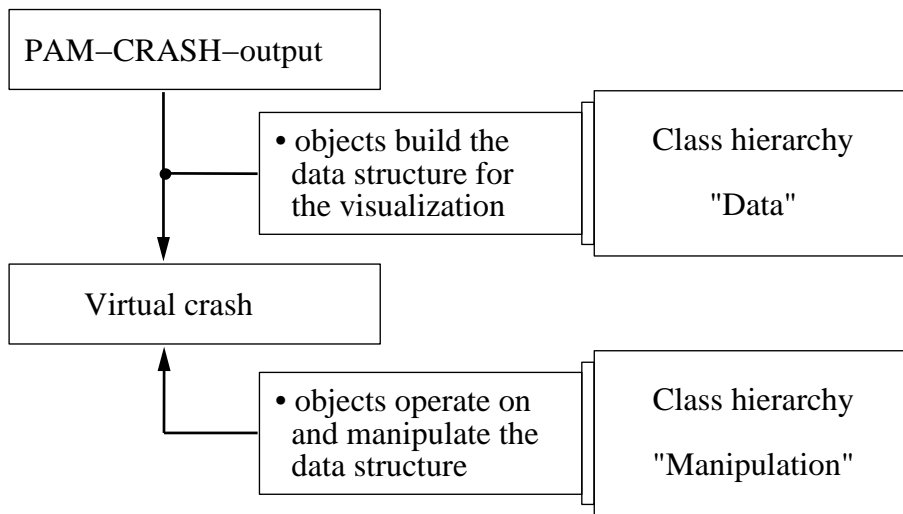


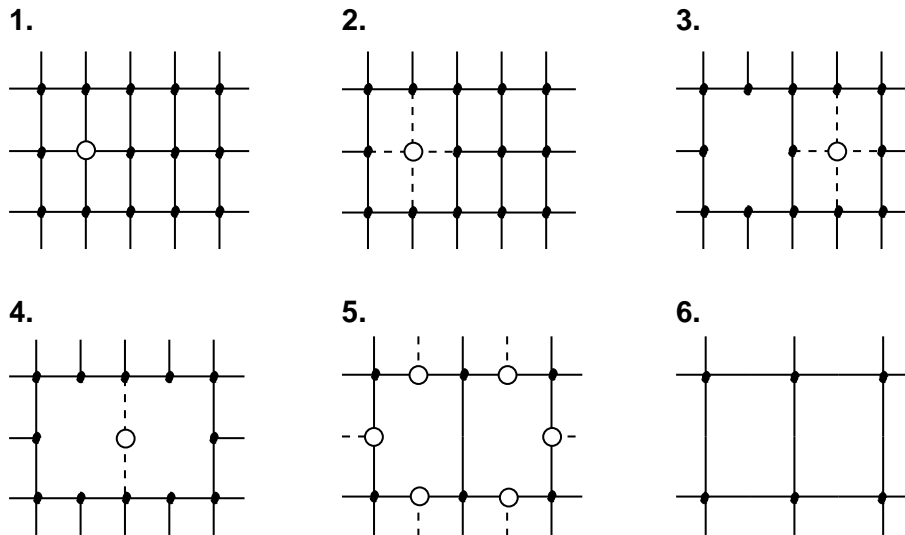
Figure 2: Object oriented design of `VtCrash`

is designed in an object oriented fashion (See Figure 2). The data is structured into a class hierarchy (Car, Group, Poly, Vertex, GlobalVertex) derived partly from the element structure the FE models are built upon. The element library of PAM-CRASH includes, among other, solids, shells and beams, which are collected into groups defining the vehicle's components. Geometric data comprises labelled nodes with global coordinates for each time step of the simulation and labelled elements which reference the groups they belong to as well as their nodes. `VtCrash` employs efficient data sorting methods to generate new local polygon lists with bidirectional pointers between nodes and polygons, creating a data structure suitable for the animation of all time steps of a crash test. Shell and beam elements from the FE package are converted into polygons in a straightforward manner. In the case of solids, an algorithm has been designed that first builds polygons from the outer sides of each solid, then eliminates redundant polygons between adjacent solids. For all vehicle components originally modelled as solids, only those polygons which define the outer surface are finally retained. A hierarchically built scene graph encapsulates the graphics and visual simulation features. The tree is made up of a root-node and environment-control nodes which control the animation. Finally, geometry nodes contain the topological information of the vertices as well as graphic attributes of the polygons like color, transparency and lighting. Geometry nodes can be manipulated interactively at runtime.

Mesh simplification. In order to meet memory requirements and to maintain frame rates of at least 15 frames per second, the polygon mesh of the model needs to be simplified. Since it is necessary to keep the shape of the model consistent during the animation, the simplification algorithm is applied to all time steps, identifying and preserving those vertices relevant for the animation of the deformation and eliminating the rest. Our method operates strictly on the basis of the original vertices, without remeshing by creating new vertices. This is essential since we need to preserve the scalar or vector valued physical data computed in the FE analysis which is assigned to each vertex of the original set. The chosen polygon decimation criteria is geometric in nature and is based on general ideas of [SZL92] and [NN95]. We compare the angle of tangent plane normals of polygons sharing the same vertex. For differences of 5 degrees or less, this vertex is deleted and a new polygon is created dissolving the other polygons formerly belonging to this vertex. Next, adjacent vertices and their polygons are checked and if the normal vector criterion applies again and another new polygon is created, the vertex lying at the edge between both new polygons can be eliminated. The goal is to create larger

polygons with fewer vertices and to obtain as few triangles as possible after the triangulation of these multi-edged polygons. Our algorithm is fast enough not to delay the load time of the virtual scene unacceptably. We achieve reductions in the number of polygons of about 50 percent. The simplification algorithm is outlined in Figure 3.

User interaction. The virtual crash environment is immersive and creates an actual sense of presence for the user, achieved through head-coupled stereo displays and gestural input techniques. The equipment consists of a Fakespace BOOM, and a Virtual Technologies DataGlove with an Ascension tracker. *VtCrash* can also be used as a non-immersive system using spacemouse and traditional mouse as input devices and stereo projection or graphics console for output. The time evolution of the vehicle deformation can be controlled and manipulated in real time. Several car components can be picked with a data glove and isolated for evaluation of details. Occluding parts can be eliminated completely or made semi transparent to reveal the behavior of hidden components. The user can grab a cutting plane referencing either an Euler or Lagrange type coordinate system, translate and rotate it freely and slice through the vehicle to separate structural components and view dynamic cross sections.



Example: Simplification of the knee bolster component group

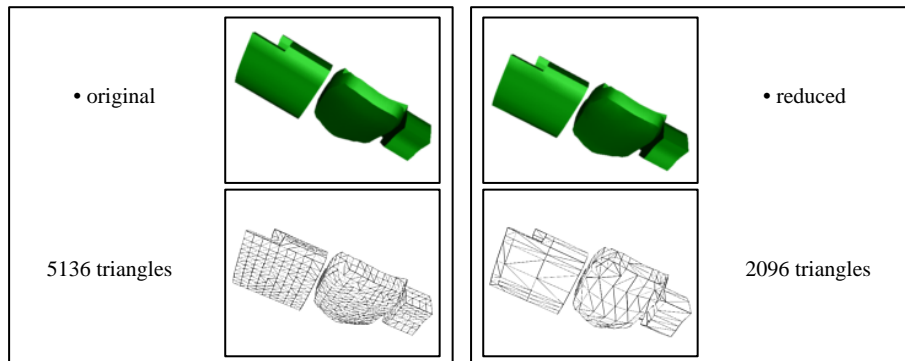


Figure 3: Mesh simplification

EFFECTIVENESS OF THE SYSTEM WITH RESPECT TO THE APPLICATION AREA

Our virtual environment provides a faster and much more powerful animation as compared to traditional postprocessors. Since it is user controllable it enhances the analytical insights into complex scenarios, which is important especially for communication between people with different expertise and background. **VtCrash** allows the investigation of crash simulations by engineers and executives without special knowledge of traditional postprocessing due to forming up intuitive interaction mechanisms. The user is able to fly through the crash test, to separate components, to observe their crash behaviour and to remove occluding parts. The components can be manipulated using virtual tools to get additional internal insights. Moreover, **VtCrash** can provide information not readily available in a real world testing scenario, i.e. make physical properties of the car components like stress or strain visible.

CONCLUSIONS

Three dimensional interactive real time visualization exhibits a high potential to optimize the product development process in the automotive industry and to improve significantly quality and functionality of the products. With our system **VtCrash** we take advantage from the evolution of computer technology. We verified the possibility and necessity of implementation of VR techniques in crash simulation. **VtCrash** is able to process any output of the finite element solver PAM-CRASH, thus it is applicable in a productive environment. The system opens the door for establishing a virtual crash laboratory which turns out to be a communication platform for the analysts performing the FE simulation as well as the product development engineers, thus supporting multidisciplinary concurrent engineering.

FUTURE WORK

More research is necessary in a variety of areas. In order to benefit from the recent development of hardware and software technology it is necessary to extend **VtCrash** to multiprocessing capabilities. Work on controlling shared and distributed virtual environments is needed to eventually succeed in realizing distributed virtual simulation. This could encompass not only different locations but also the integration of different types of simulations (acoustics, thermodynamics, etc.). Integrating data management into the visualization using web browsers is another interesting issue. Finally, merging video data of real world engineering field tests into a virtual simulation environment promises to enhance and augment complex engineering analysis tasks.

References

- [BF95] Steve Bryson and Steven Feiner. Virtual environments in scientific visualization. In *Virtual Reality for Visualization, Course Notes of Tutorial 5 at Visualization 95*, 1995.
- [BL92] S. Bryson and C. Lewit. The virtual windtunnel. In *IEEE Computer Graphics and Applications*, 12(4):25-34, 1992.
- [DG94] F. Dai and M. Goebel. Virtual prototyping - an approach using virtual reality techniques. In *Proc. of 14th ASME Int. Computers in Engineering Conference, Minneapolis(USA)*, 1994.
- [KHE95] Sven Kuschfeldt, Michael Holzner, and Thomas Ertl. Video integration of pam-view visualization results. In *Proc. of PAM 95, Fifth European Workshop on Advanced Finite Element Simulation*, 1995.
- [NN95] Daisuke Nishioka and Mikio Nagasawa. Reducing polygonal data by structural grouping algorithm. In *Lecture Notes in Computer Science 1024, ICSC 95*, 1995.
- [SZL92] William J. Schroeder, Jonathan A. Zarge, and William E. Lorensen. Decimation of triangle meshes. In *Proc. of SIGGRAPH 92*, 1992.