3D Visualization and Animation – An Introduction

ULRICH LANG, UWE WÖSSNER, University of Stuttgart
JOACHIM KIEFERLE, University of Applied Sciences Wiesbaden

ABSTRACT

Visualization represents a support technology that enables scientists and engineers to understand complex relationships typically represented by large amounts of data. The visualization process chain forms the basis for explaining the characteristics of scientific visualization and how it is used in different application fields. The combination of various visualization techniques enables to explore simulation models and analyse datasets. Additionally engineers can judge complex geometries and use visualization and virtual reality techniques to communicate complex content and support decision processes. An application case is used to illustrate this.

1. INTRODUCTION

With the fast advances in hardware technologies the data volumes resulting from measurement and computing devices increase very fast. Data as intermediate carrier of information can not be immediately understood by humans. Visualization is the process to convert different forms of information and represent it in a visual manner, thus allowing humans to recognize states, structures and behaviour. The term scientific visualization was introduced in 1987 (McCormick, 1987). Since then visualization is evolving as an own discipline which has been structured into scientific and information visualization. While information visualization focuses on the visual representation of non-spatially structured information, scientific visualization is mainly oriented towards the visualization of data being defined on multi dimensional domains. Visualizing data distributed in 3D space enables humans to make use of their evolutionary developed capabilities to discern structures at certain locations or see spatial transitions in structures.
Many engineering disciplines focus on the development of products that are mainly characterized by their physical shape like cars, buildings, satellites or bridges. In addition their behaviour and properties is of importance. Bridges need to be stable, cars should have low fuel consumption and need to be safe, buildings should be energy conserving, etc.. While the visual appearance of such objects can be directly visualized, the behaviour and properties first have to be mapped into a visual representation that can be easily and intuitively understood. As properties in reality don't have a visual representation a visual metaphor is required that allows an intuitive understanding. This is further complicated if the relationship between different parameters should be understood.

2. THE VISUALIZATION PROCESS CHAIN

The visualization process chain describes the sequence of processing steps and their relationship as shown in figure 1. It consists of the source process, which either generates or reads data. Instead of the simulation it could also be a measurement process or the reading of data that has been produced earlier. Examples of measurement-based data are satellite born images or computer tomography datasets in medicine. The filter process, either selects or samples data, corrects measurement errors or produces deducted information. In fluid flow simulations the vorticity could e.g. be computed from the basic simulation parameters. The mapping step represents the core of the visualization process. The selected visualization method converts data into abstract visual representations. There is a multitude of different visualization algorithms that implement different types of mappings, each of them having their specific capabilities. In most cases the mapping leads to a collection of geometric primitives such as triangle lists, line lists, point clouds, etc. This is combined with information such as textures and material properties of surfaces. Figure 2 shows two example visualizations for a fluid flow field. In the left image particle paths visualize the velocity field of water flowing through a water power plant. The right image shows the colouring of turbine blade surfaces due to pressure distribution as well as an isosurface of enthalpy in the flow field of a water vapour turbine.

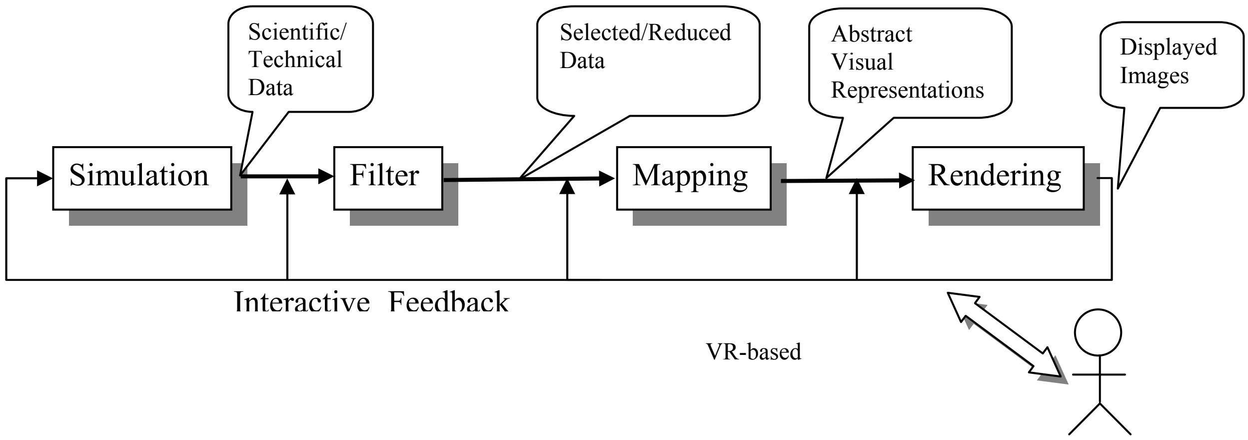


Figure 1: The visualization process chain

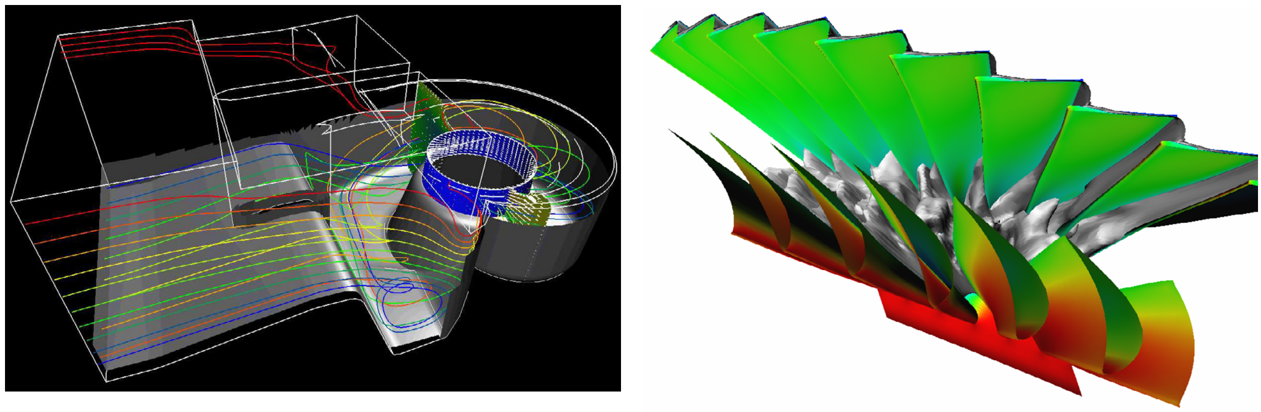
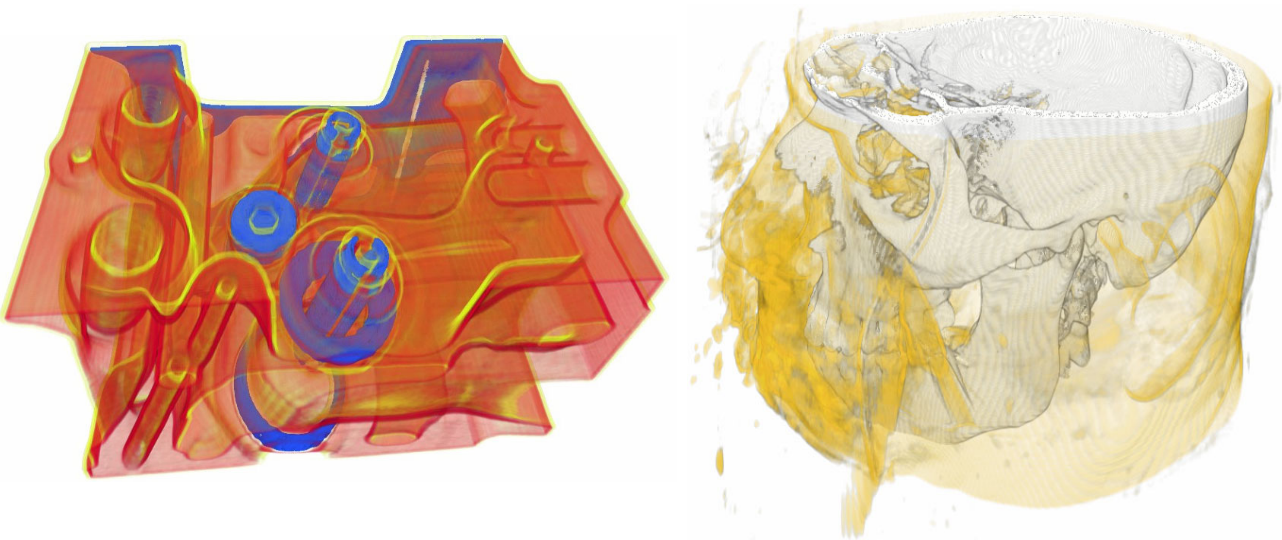


Image Courtesy of IHS, University of Stuttgart   Image Courtesy of ITSM, University of Stuttgart

Figure 2: Particle paths and isosurfaces to visualize fluid flow behaviour

In the rendering step the scene descriptions together with lighting information and camera positions is used to generate images of the scene which are then displayed. The rendering step can be conceptually split into the image generation and the display step. In the display step series of images can be collected and viewed in fast sequence thus appearing as a continuously animated representation of the selected content.

Volume rendering as a special visualization method conceptually integrates the mapping and rendering steps. Its input is a scalar field defined on a three-dimensional grid which is interpreted as a semitransparent medium. Via transfer functions for transparency and colouring a mapping of the scalar values in each volume element (Voxel) is performed. These semitransparent coloured voxels are then superimposed to form an image of the overall volume. Volume rendering bypasses the geometric representations between the mapping and rendering step. A multitude of algorithms exist on how to define the transfer functions and how to accumulate the voxels. Aims are to detect subtle structures and reduce the processing time. Figure 3 on the left side shows internal structures of a metallic motor block. The data has been acquired via computer tomography. On the right the bone and skin of the visible human (Visible Human Project, 2003) is shown while all other materials are rendered transparently.



CT dataset of GE Motor Block   Visible Human Dataset

Figure 3: Volume rendering of measured and simulated data

2.1. Interaction and feedback

The visualization process is used for different purposes, accordingly one differentiates e.g. between exploratory visualization and presentation visualization. In exploratory visualization the scientist or engineer does not know before hand the structures or the behaviour of a system that is simulated. Therefore the visualization toolkit needs to support an incremental exploratory process to search for structures or behaviour. Such a process has a highly interactive character.
In the visualization process chain the interaction activities are separated into multiple feedback loops. The innermost loop feeding back into the rendering step allows modifying the observer position and orientation thus enabling a free roaming in the 3D scene. The modification of camera parameters additionally allows zooming into specific details of the scene. As many simulations produce time dependent results it is of equal importance to understand the dynamic behaviour of a system. The human perception requires that the dynamic behaviour is shown in an appropriate time scale. To allow this the scene content for the different time steps need to be stored in memory enabling a quick switching between them thus giving an observer the impression of a smooth change in structure respectively a smooth movement of objects. An exploratory visualization process is characterised by a repetitive display of the dynamic behaviour while changing viewer parameters while an interaction concept similar to a video recorder is required to slow down the animation speed, step through time or reverse the orientation of the animation time.

In the next outer feedback loop a user can interact with the parameters of a selected mapper. Real process chains typically consist of multiple filters and mappers as shown later. Typical interactions with mappers are e.g. the repositioning of a cutting plane, the definition of new starting positions for particle traces in a flow field or the definition of a new isovalue of an isosurface. Such types of interactions are applicable to the 3D visualization of figure 4. Thus a user can locate a specific region where a certain effect appears or an unusual behaviour is determined.

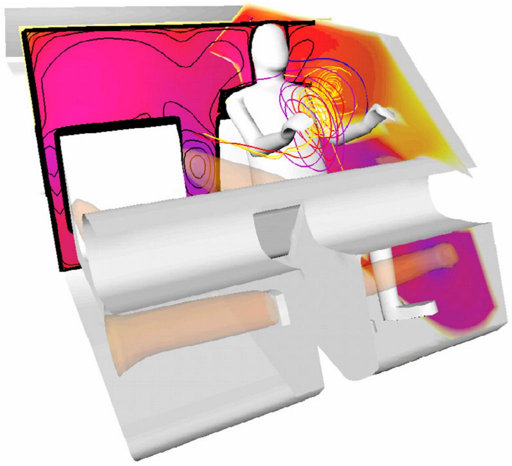


Figure 4: Climate simulation in a car cabin(data courtesy DaimlerChrysler Research): coloured cutting plane, particles paths and isosurface used for visualization

The next outer loop allows interactions with the filtering steps. Filter parameters enable to select subdomains of a region together with the values defined on this subdomain. An alternative approach to reduce the volume of data to be processed is to sample it down. This speeds up the processing and thus raises the interactivity. When a peculiar behaviour is located in a certain spatial region, the sampling can be removed and an extraction of a region of interest can be applied.

Finally the feedback into the input of the simulation step is called simulation steering. Here a user can see the simulation results change as the simulation evolves. With the immediate feedback the user can modify boundary conditions and with a certain delay see how the behaviour of the simulated system changes. Also here the time scale of the human perceptual system is of importance. Changes need to happen within seconds to be perceived as a dynamic behaviour.

From the inner to the outer loop the timing requirements for the reaction of the system become less demanding. When moving through a scene or interacting with other animation parameters the system should ideally react within a 1/30 of a second. This requires an image update rate of at least 30 frames/s, which limits the complexity of the scene. Modifying mapper parameters can already take longer, especially if they need to be applied to a whole sequence of time dependent data. In such a case a new isosurface would e.g. have to be recalculated for all time steps.

Finally large-scale simulations can take hours or even days to finish. Waiting to see the modified behaviour of such a system after changing an input parameter seems to be rather inappropriate. Instead, the visualization system collects the result data as it appears over time and keeps it for a repetitive analysis. Thus the user can repeatedly look into the time dependent system behaviour. This enables him to interrupt the simulation very early when he recognizes an erroneous system behaviour. Alternatively he can provide other more appropriate input parameters.

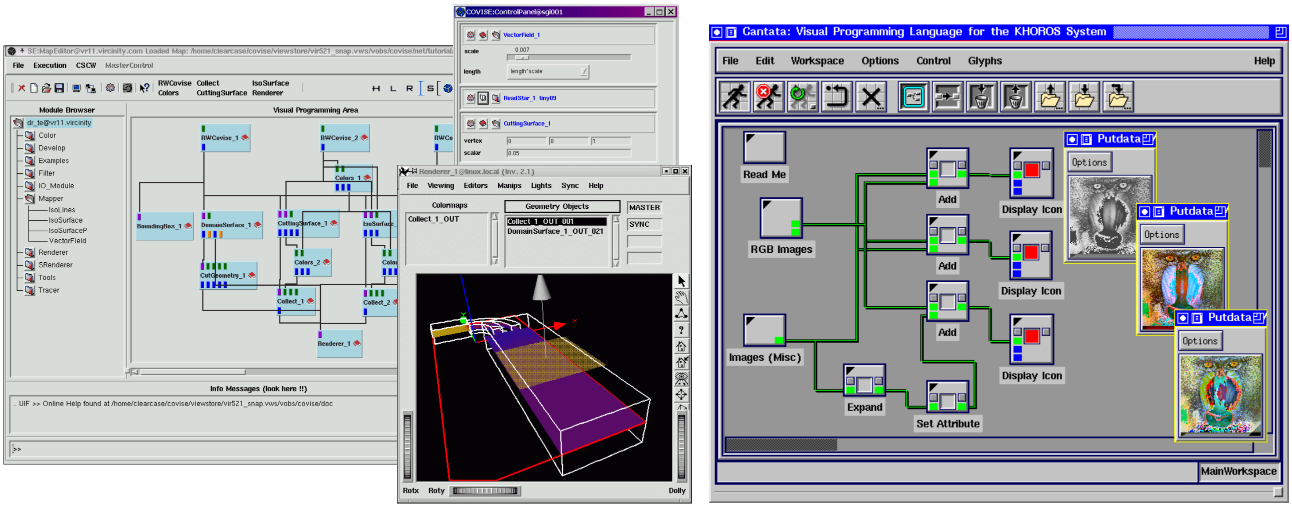
3. SCIENTIFIC VISUALIZATION PACKAGES

Since the introduction of scientific visualization multiple modular visualization packages have been developed that have strong commonalities but also clearly differentiate from each other. A common approach is the description of the visualization task via a data flow network paradigm. This reflects the concept of the visualization process chain. A visual program editor allows configuring the topological relationship of the processing steps graphically. Exchanged data is depicted as edges of a graph connecting the processing steps. Figure 5 shows two examples of such packages. COVISE (COVISE 2003) which has been developed at the High Performance Computing Center Stuttgart

(HLRS) and Khoros (Khoros, 2003) which has been developed at the University of New Mexico. Further packages like OpenDX, AVS or NAG Explorer have similar dataflow networking paradigms applied. Most of the packages allow executing a visualization process chain distributed across multiple machines in a computer network.

Compared to other software integration platforms, visualization packages typically have to be optimized for interactive work. There are different and complementing approaches to achieve this. OpenDX e.g. implements modules as subroutines in one executable. While this reduces the execution overhead it prevents the parallel and independent execution of modules within the same machine. Systems such as AVS, Khoros, COVISE and NAG Explorer implement the processing steps as separate processes that allow a high flexibility in the distribution of processes across different machines. While there is some execution overhead on the same machine the operating system automatically executes multiple modules in parallel. On multi processor machines this leads to an efficient handling of modules without additional effort of the user.

Large data flow networks can consist of more than hundred modules. To maintain an overview, many packages allow to collapse a whole set of modules into one macro and visually represent it by one icon. Data flow networks can be optimized for memory usage or execution speed. For high interactivity the later is the preferred option. Therefore most packages typically cache data objects, which can lead to a large data overhead. This is further aggravated if time dependent data is processes by a system. On the other hand the delay for any type of interaction in an exploratory visualization is minimized. OpenDX went even further in allowing a selective activation of this caching mechanism. When knowing which methods will be reused for parameter changes, one can enable the caching of the input to this module thus avoiding that the data has to be processed again from the beginning.



COVISE Khoros

Figure 5: Screen snapshots of modular visualization packages

COVISE is optimised to make efficient use of the high performance networking infrastructure of a typical high performance computing center by adapting buffer sizes, using asynchronous communication and assembler routines for data conversion between different machine platforms.

Most visualization packages focus on the visualization step assuming, that the simulation has been performed before. COVISE treats the coupling of visualization with an ongoing simulation as equally important. Therefore a specific communication library was implemented that allows an efficient coupling to ongoing simulations on remote supercomputers. Within the visual program editor such a remote simulation appears as a module and does not differentiate in its handling.

Many of the visualization packages have been extended toward collaborative working, allowing users at different locations to discuss visualizations as if they would be in one room looking on one workstation screen. While such an add-on functionality is often difficult to be integrated it has been a principle design concept of COVISE form its inception. It is inherently available in the extensions that are added to COVISE.
COVISE is taken as a sample application to describe a visualization system architecture. Figure 6 shows the core processes as well as the modules and how they interrelate. When a COVISE session is initiated, Mapeditor and Controller are started. Optionally an additional remote user can be invited to participate in the session. If accepted, a further Mapeditor is started on the remote machine. Data manager processes are started on all participating machines. Supercomputers, are added as further machines without a Mapeditor. The Mapeditor enables a user to establish and configure the module network he wants to execute. As soon as he brings up an icon representing a module the same icon also appears on the workstation screen of a collaboration partner. Additionally the process is started on the respective platform and switches into an event wait. Then the user connects the modules to define execution sequence and data exchange. Data flow networks are typically stored after having been set-up. Thus they can be loaded when they have to be reused.

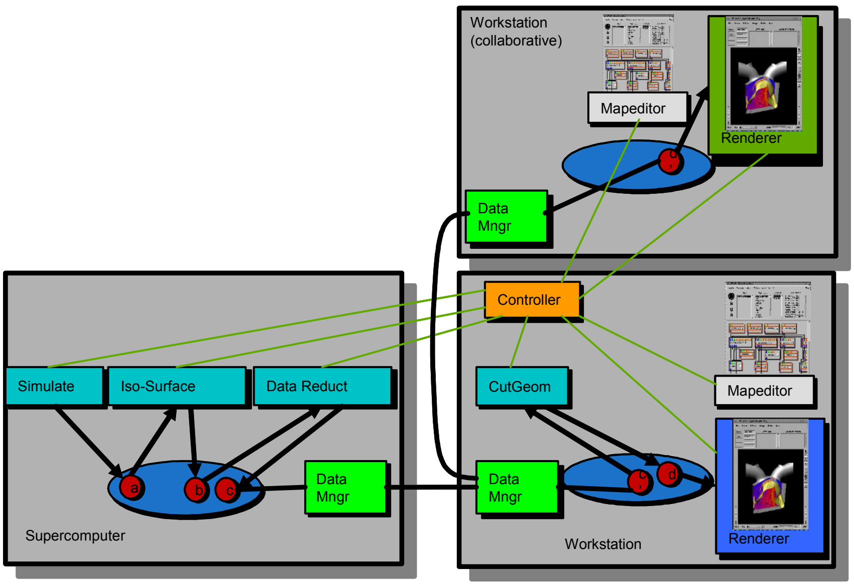


Figure 6: COVISE software architecture

When a data flow network is executed the controller sends messages to the modules in the topological sequence. The messages contain the names of files to be read, the names of data objects to be created or accessed as well as parameter values to be used by the module. Names of data objects are then sent to the data manager on the same machine together with characteristic sizes. This enables the data manager to allocate memory for the data object and pass a pointer back to the module to access it. When existing data objects need to be accessed, a pointer is immediately passed back. On machines with virtual shared memory data objects are mapped into the virtual address space of the modules thus avoiding the copying of large data objects. If data objects produced on one machine need to be accessed on another machine they are copied by the data managers from machine to machine.

4. 3D MODELLING AND TEXTURING

In the engineering sciences the modelling of object and part geometries is a process typically handled by CAD packages. Depending on the application field different CAD tools exist. The geometry forms the basis for the simulation of the physical behaviour of an object or part. Properties of interest could be the stiffness of a part, its thermal behaviour, its deformation when external forces are applied or its fluid flow behaviour. To simulate the behaviour a grid has to be defined which allows discretizing the domain of the physical behaviour. Based on additional initial and boundary conditions the calculation determines the time dependent behaviour of the respective object or part. To visualize the behaviour the shape of the part or object needs to displayed at the same time. In figure 4 this is e.g. the cut open part of the bounding geometry of a car cabin.

Whereas for mechanical engineering mostly different colours on the model surfaces are sufficient, architectural representations depend on the visual representation of surfaces like concrete, wood or plaster. As the atmosphere of a visualisation mainly depends on the mapping, one has to put a focus on it. To apply textures as well as to reduce the number of polygons to allow decent frame rates, further modelling and animation tools like 3D Studio MAX are used. As most of the CAD and modelling packages defined their own proprietary file format a common exchange format is required to pass the geometry on to the visualization package. VRML97 has evolved during the last years as this common file format describes 3D geometry and behaviour of models for the internet.

A VRML/VRML97 importer with extended capabilities is integrated into the COVISE renderer COVER to import VRML97 models to combine the imported geometry and mappings with the visualisation of measurement or simulation data. VRML97 supports interaction and animation that greatly assist the users in immersing into the scene.

5. VIRTUAL REALITY TECHNIQUES FOR 3D VISUALIZATION

A virtual reality as described here is produced by a combination of technologies that give a user the impression to be immersed in a computer generated scene. It is important to cover a large viewing angle, asthe peripheral view is an essential element of the human perception for having the impression of being inside the virtual world. This is best accomplished by setting up a CAVE like environment as shown in figure 7 consisting of at least 3 stereoscopic projection walls and a stereo

GIS and terrain model representation in VR

Temperature isosurface in a car cabin

Figure 7: Immersive virtual environment consisting of 3 stereo back projection walls and a floor

projection floor. To support this immersive impression the displayed world needs to react immediately to movements of the observer and allow direct interaction with the scene content. A user should be able to grab objects, move them around and perform other interactions directly and intuitively that fit to the displayed content. In the example of the climate simulation in the car cabin the scientist should e.g. be able to insert new particles in the air flow and see the paths they follow appear immediately. Figure 7 on the left shows GIS data of Zürich area layered over the terrain model. On the right two scientists discuss the temperature distribution inside a car cabin produced by a previous simulation.

Depending on the scenario further sensory information can be very supportive. For a medical specialist force feedback is essential during the training of an operation. For an architect or urban planner the auditory information within a larger building or street strongly improves the sensation of being there. In the real world objects can be moved with constraints, doors can be opened, etc. Users expect the same behaviour of objects like in a real world. Therefore time dependent event driven animations of objects are an essential element of a virtual environment. This e.g. allows calling an elevator by pushing a button, which opens and closes doors and caries users to different levels of a building.

6. 3D VISUALIZAZION AND ANIMATION TO REPRESENT CONCEPTS

Combining all the elements described above allows to present engineering design and development processes which not only focus on the layout and optimization of products but also on the development of usage concepts and profiles such as of buildings or cars. Layout concepts orient the focus of interest on certain areas of a building, or avoid distraction of a driver from a road while providing complementary information. In a virtual environment it can be shown how to guide humans through buildings or how to make users feel comfortable there.

Mostly combined approaches are used with maps and elements representing design concepts as well as animations explaining the concept.

7. SCALING RANGES OF 3D VISUALIZATIONS

The model size in 3D visualisation ranges from very small models like an air vent of a few millimetres diameter to models of urban or landscape size of several kilometres. Although a CAVE has a limited size of approximately 2.5 to 3 meters side length, all dimensions can be represented. Due to the reach of the human body the models are often scaled during interactions and then rescaled again. While the engineers scale up the vent to a size of a few meters to see single particles, the architects rescale their 1:1 model to architecture model scale, to make changes and then rescale to 1:1 again to judge the changes.

7.1. Sample: DC Autohouse

DaimlerChrysler is one of the leading applicants using interactive visualization in architecture. To design a new generation of autohouses virtual reality has been used from the very beginning of developing the general building concept to the final projects. Team meetings with many disciplines are held in the CAVE to discuss about the architecture and its impact in 1:1 scale. Architects, brand managers, sales specialists, event designers, marketing specialists, artists, simulation experts (e.g. airflow, temperature) and even potential customers discuss in the 1:1 project representations.

This way of working is part of the communication concept "MarkenStudio" (Drosdol et. Al. 2003) in which visualization plays a major role. It could be observed, that as soon as there is "something to look at" and ideas are being visualized, it is much easier to achieve a commonly agreed meeting result. The participants are more willing to change their "point of view" and to understand and accept the ideas of the others more easily. Additionally virtual reality assists in reaching a high degree of planning safety at an early stage.
To allow different users appropriate interaction with the model, different interaction methods like colour-picker (changing the colour of walls / floors / ceiling interactively), texture-picker (changing texture on the fly to judge the right material), exhibition-designer (creating, placing and modifying exhibitions interactively) or switching through variations have been implemented. As understanding takes a certain time, it became apparent that many ideas could be communicated much better with animations.

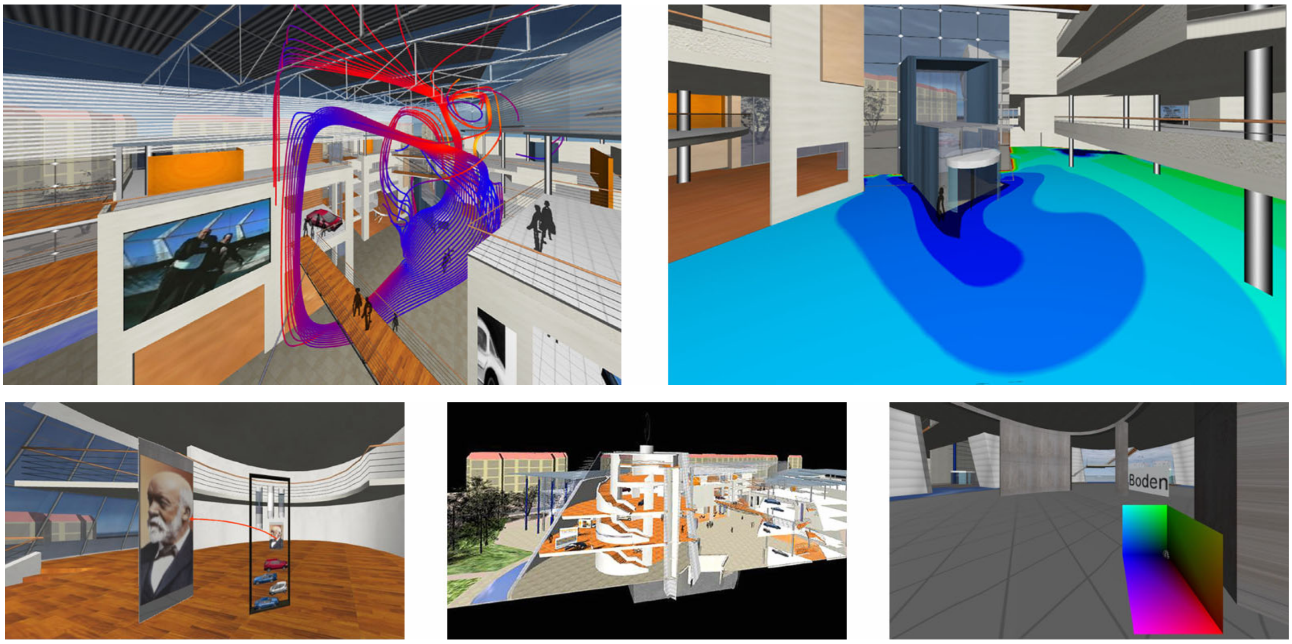


Figure 8: Snapshots of DaimlerChrysler Autohouse in a Virtual Environment

8. CONCLUSION AND OUTLOOK

In a world that becomes more and more interlaced with increasingly complex tasks, visualization provides assistance to understand complex structures and relations. Besides supporting only single users, virtual reality e.g. in a CAVE like environment assists the group discussion aspect. Due to the representation of planned objects "as if they were there" complemented by further property or behaviour information, specialists and lays can discuss very effectively.
Augmented reality represents the next step of information presentation by combining virtual objects and scenes with real world elements, overlaying them as required. This approach allows to move out of virtual reality laboratories into the real world and make use of virtual reality techniques there.

9. ACKNOWLEDGEMENTS

We would like to thank DaimlerChrysler for allowing using their datasets.

10. REFERENCES

COVISE (2003): www.uni-stuttgart.de/covise

Drosdol, Johannes; Kieferle, Joachim; Wierse, Andreas; Wössner, Uwe (2003): Interdisciplinary cooperation in the development of customer-oriented brand architecture. Proc. of Trends in Landscape Modelling, Dessau. p. 264-269.

McCormick, B.H., T.A. DeFanti, M.D. Brown (ed) (1987): Visualization in Scientific Computing, Computer Graphics Vol. 21, No. 6, November 1987.

Rantzau, Dirk (2003): Ein modulares Konzept für die interaktive Auswertung von Simulationsdaten in einer verteilten und immersiven Virtuellen-Realitätsumgebung, Promotion, Universität Stuttgart, Februar 2003

OpenDX (2003): www.opendx.org

Khoros (2001): www.khoral.com

The Visible Human Project (2003): http://www.nlm.nih.gov/research/visible/visible_human.html