FAVEum
Framework Architecture for Virtual Environments Applied to Urban Modelling

J. O. Erdal, T. Langeland, D. Patel, I. K. Eliassen

Christian Michelsen Research (CMR) (www.cmr.no)
P. O. Box 6031 Postterminalen, N-5892 Bergen, Norway
{jone, tor, daniel, ingekr}@cmr.no

KEY WORDS: Virtual environment, 3D user interaction, rapid prototyping, configurable framework, authoring system, urban modelling, new paradigms for 3D GIS, innovative visualization techniques

ABSTRACT:

FAVE (Framework Architecture for Virtual Environments) is a generic software framework for developing VR applications. It contains a toolkit of user interface widgets, navigation techniques, interaction features and optimised visualisation algorithms. Non-programmers can specify application behaviour. Applications developed in the framework can be used on different platforms, from immersive stereographic VR-environments down to desktop computers.

In this paper we present development and tests of a 3D urban modelling application based on FAVE. We have selected a few functionalities important to urban modelling relevant to user demands, such as modular scaling, silhouette analysis, design comparison, as well as safety and risk analysis. In addition, we present three functionality examples in our application prototype (FAVEum) based on the rapid prototyping qualities within FAVE. The chosen functionalities have been tested using a simple 3D model of a part of Bergen, Norway.

1. INTRODUCTION

Immersive Virtual Reality (VR) is an aid to mastering spatial complexities, by facilitating improved understanding of 3D data and offering new modes of interdisciplinary collaboration. Immersive VR has gained usage within a range of application areas, e.g. design, medicine, oil, and gas exploration, as well as production. The authors of this paper are expecting an increasing use of immersive VR also within urban planning and modelling.

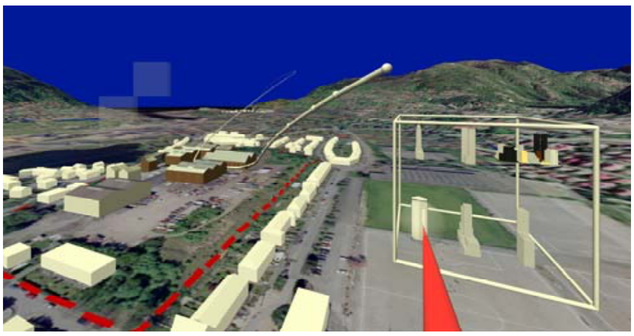

Figure 1. A model of the Bergen area with buildings at Kronstad (near) and downtown (far). To the right is a menu for importing buildings. Also shown is a spline with control points representing an editable flight path

We present a highly configurable, rapid prototyping framework for immersive virtual environments, FAVE (Patel, 2003). The framework is object oriented, event driven and has been designed with collaboration over networks in mind. It consists of several abstraction layers and application behaviour is separated from graphical data and is specified using XML. This enables development by non-programmers and facilitates comparisons of different user interaction strategies.

Issues regarding 3D user interaction have been investigated, including an efficient, context-sensitive, hierarchical extension to the command and control cube. The framework has been used to prototype an urban modelling application, FAVEum (Figure 1).

One important motivation for developing a system facilitating easy configuration of user interface and application behaviour is that we are interested in comparing user interaction methodologies for VR in general. Challenges related to developing efficient user interfaces for virtual environments (VE) delay the utilization of the full potential of the VR technology. User interface design is held to be one of the great challenges within computer science (Brooks, 2003). The FAVE framework facilitates easy and rapid comparisons of different user interaction strategies. In the urban modelling prototype we have used these capabilities to experiment with different user interface elements.

2. RELATED WORK

VR frameworks cover many different domains. Frameworks like CAVELib (CAVELib) and VR Juggler (Bierbaum, 2001) offer support for basic VR functionalities, such as stereoscopic viewing and device sampling. The Quanta (previously CavernSoft) framework (He, 2003) is an API for network communication, such an API is necessary for extending a single user VR application to a collaborative application. Many higher-level VR frameworks use these or similar building blocks accompanied by specialized graphics libraries or commercial ones like OpenGL Performer (OpenGL Performer). In addition some frameworks come with a scripting environment, making them more flexible and configurable. Frameworks supporting scripting often also support the dataflow approach seen in SGI's Open Inventor (Open Inventor) and in VRML 2.0 (Carey, 1997). In Inventor and VRML 2.0 the flow of data between virtual world entities is established by creating channels or connections between the different entities' attribute values. Avango (Springer) is an example of a framework that supports scripting.

3D urban models can cover a wide range of topics, and the focus areas are naturally depending, both in size and scope, on the role of the users – the general public (car navigation (SONY, 2005), and web based information (NMM)), urban planning commissions, urban planners, site developers (urban design guidelines (Mak, 2004)), large scale project proposals and presentations (Olympic Games & Urban Development of Beijing, CrystalCG), emergency operators (safety analysis and training (Jafari, 2003)), to mention a few.

Observing different discipline-specific approaches, we note that usage of 3D models are well integrated with visual assessment methods widely applied within landscape architecture (Lange, 1999, Ervin 2001), in addition to the more database-oriented and multi-applicable approaches often found within geomatics (Zlatanova, 2000). Within computer-aided architectural design, references are also rich (CumniCAD), showing a developing terminology (Martens, 2001) and a globally well organized professional field (Architectural Computing Organisation).

3. INTERACTION TECHNIQUES

The toolbox of controllers in FAVE implements a wide range of interaction techniques. Interaction techniques are typically divided into four groups: navigation, selection, manipulation, and system control.

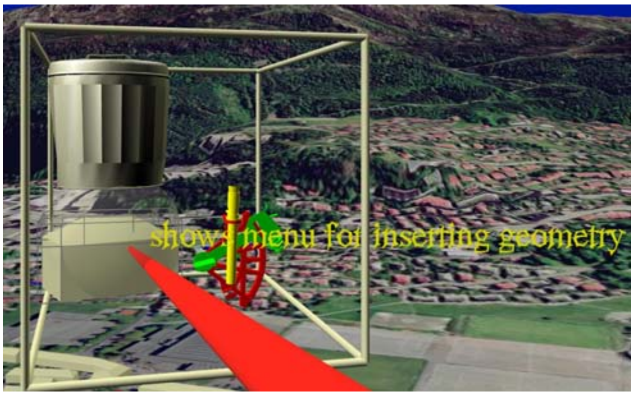

Figure 2. Main menu as C3

System control is defined (Bowman, 2001) as the task of changing the interaction mode or the state of the system by issuing commands. These commands are issued in our system by means of interacting with controllers. The Controller class is used to implement simple widgets, such as buttons, sliders and menus, as well as more complicated tools, e.g. for managing editable flight paths.

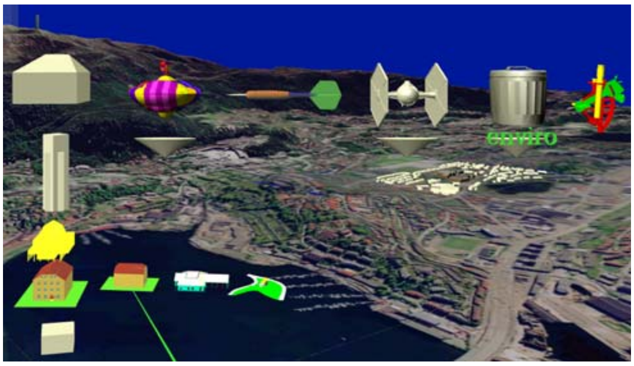

Figure 3. Main menu as a menu bar

The distinction between widgets and tools is not always as well defined in virtual environments as in traditional desktop WIMP environments. Some controllers in the scene can be multi-functional and act as both widgets and tools. One difference, though, is that the controllers acting as widgets tend to provide more elementary behaviour and will probably be reused across different application domains, whereas the more advanced tools are generally more bound to specific application domains.

Figure 4. Controller widgets, menu showing two buttons and tool-tip text

We have implemented controllers that can be used as a widget library. Figures 2, 3 and 4 show applicable usages of these widgets. Each instantiated widget will have a tag in the XML file. The central widgets in this library are buttons and toggle-buttons for performing discrete commands (the icons in Figures 2, 3 are buttons, and Figure 4 toggle buttons), sliders for specifying step-wise values (Figure 8), text windows for showing lines of text and menu bars (Figure 3) and C3 cuboids (Figure 2) as two different ways of presenting a hierarchy of commands to select among.

The C3 cuboid was inspired by ideas from the Command and Control Cube (Grosjean, 2002). It is a powerful metaphor for structuring and presenting collections of commands, possibly in hierarchies, using very little screen area and having quick and precise navigability. We have called the widget a C3 cuboid due to its shape; its edges are drawn as lines. The cuboid is a

3D matrix consisting of equally sized smaller cubes, each containing a geometry representing a controller.

4. FAVE APPLIED TO URBAN MODELLING

For the purpose of evaluation and gaining experience the FAVE framework has been used to prototype an urban modelling application. The application runs on Linux PCs and on SGI Onyx graphics computers, and can easily be ported to any other platform with support for the libraries CAVELib and OpenGL Performer. The application makes use of two different modes, a desktop mode (applying a mouse and a keyboard), and an immersive stereo mode with head tracking and a 3D mouse with three buttons and a joystick.

4.1 Data Sources

The data for this application consists of textured 3D terrain data of an approximately 10x10 km area and contains buildings from downtown Bergen and Kronstad 3 km from downtown Bergen, supplied by the City of Bergen. The terrain for Kronstad is also modelled in increased resolution, this is displayed as a higher level of detail when the user navigates sufficiently close. The buildings in the two areas have different origins. The selected buildings at Kronstad are recreated manually from aerial photos, while the downtown buildings are generated automatically from data in the SOSI format (Norwegian Mapping Authority).

4.2 User Interface

The application was developed with two goals in mind. It should be both a test case for the framework architecture, and an extensible prototype of an urban modelling application. Focus has been on user interaction techniques for navigation, selection, manipulation, and system control.

The application is configured with one global main menu and several context menus specific for various tools and objects. All menus are configured as C3 cuboids. The main menu has tools for navigation, for importing graphical objects, such as buildings, and for changing system settings. The context menus provide functionality for changing settings for tools and for manipulation of objects.

Objects in the scene, such as buildings, may have menus attached to them. Such a menu is accessible by selecting the corresponding object. Thus, a building is both a geometric object part of the scene and a button for accessing functionality specific for that building. This multifunctional role prevents a clear distinction between widgets and other objects – thus having both advantages and disadvantages: On the positive side we obtain a more natural flow of interaction since geometric objects can be interacted with directly instead of interacting on a GUI separated from the object, like for instance toolbars and menus. On the other side it might be difficult for the user to identify objects that can be interacted with and which cannot, since all are embedded in the virtual world and not separated out as menus/widgets. This is partially solved by giving the user a clue that an object can be interacted with in terms of the object being highlighted when pointed at. A complete solution would be also to always have a menu representation of all geometric objects so they can be interacted with directly or through a separate menu system.

The main menu is configured in two different versions for comparison reasons. One version is a C3 cuboid, the other is a menu bar. Both versions are accessible by selecting semi-transparent icons located in the upper left part of the screen. These icons are fixed in device space coordinates, making them always close to the user. Moreover, by being semitransparent and not in the central view area they don't disturb the sense of being immersed in the virtual environment. The C3 main menu is also a context menu attached to the ground geometry. This means that the main menu also can be reached by selecting the ground and right-clicking.

The context sensitive hierarchical C3 menu system has proven to be useful and effective. It is conveniently placed right in front of the user when activated, and is otherwise invisible and not cluttering up the view. The context sensitivity provides direct access to local functionality without having to navigate from the top of a menu hierarchy. The C3 selection system is easy to use since it does not require high precision pointing. Selection can even be performed without watching once the positions of the icons within the cuboid have been memorized. The hierarchical configuration of C3, i.e. icons in a C3 can lead to a new C3, provides a convenient way of grouping similar functions, and allows for an unlimited amount of commands. Extending the C3 to span more than 3 icons in one or more directions also allows for more commands. Although this impedes quick selection (Grosjean, 2002) it can be useful for listing many choices.

4.3 Scaling – Continuous versus Modular

An urban development area is normally characterized by a need, both by the planning authorities and by the developer, in an early stage to clarify the capacity of a certain lot or building block. Capacity analysis of this kind is important and normally also decisive for the financial realism and viability of a building project that is under consideration.

Modelling tools for such analysis have been modestly developed, and are often of a rather crude spreadsheet-based character, estimating floor space (2D, 2,5D) mainly, more than estimating the visual impact (3D) from the new building toward its surroundings. We first tested a simple tool for continuous scaling. Figures 5 and 6 show the view from a window before and during vertical scaling of a neighbouring building (note that the view to Mt. Ulriken in the background becomes barely visible after scaling). Continuous scaling seems to us not to be sufficient for doing proper capacity analysis for a building site.

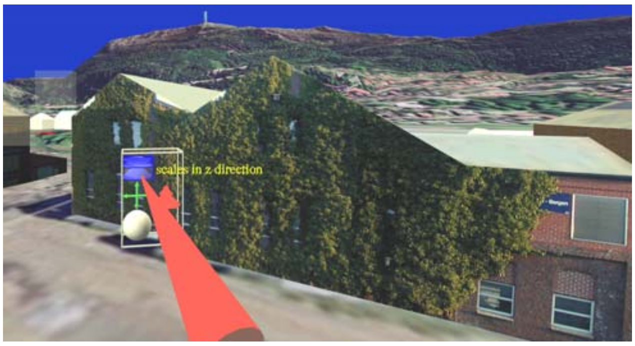

Figure 5. A view before scaling

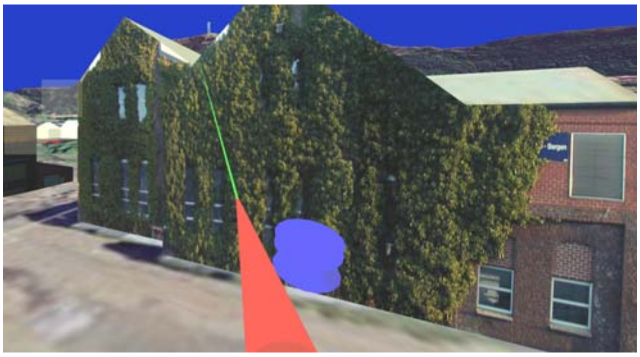

Figure 6. The same view during continuous scaling

Our proposal for a real time modular scaling is an attempt to obtain a more balanced – and volume oriented – building site analysis in the early conceptual stages of an urban design project. The way we intend modular scaling to be used is to cover a sketching function in these conceptual stages, without needing a full-scale CAAD application. Modular scaling, or 3D stepwise stretching, gives more direct analytic results than a function giving us only continuous scaling.

The modular scaling functionality is implemented by splitting a building into a set of modules (Figure 7). Three different types of module layers can be defined, one for the first floor, one for the top of the building, and one for all floors in between, i.e. the floors having the most general floor plans and elevation elements within the building. Each floor can contain up to nine different module types, one for each corner, one for each wall (i.e. containing the general elevation element), and a core module (i.e. a module without an exterior elevation). When the user scales the building vertically, the number of module layers is adapted (Figure 8). Scaling the building horizontally regulates the number of wall modules – adding modules either to the depth or to the front elevation of the building.

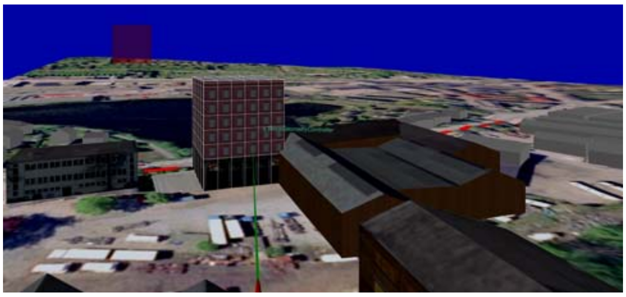

Figure 7. Modular scaling, modules highlighted (white lines)

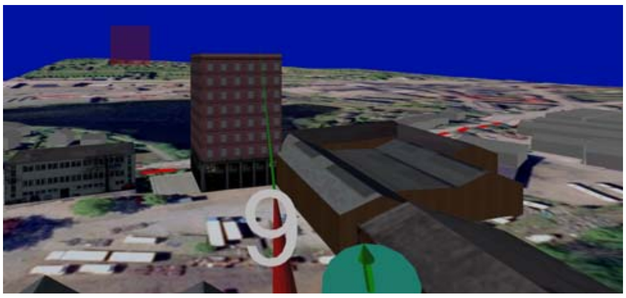

Figure 8. Modular scaling, extra modules have been added vertically. Both vertical and horizontal additions are possible

4.4 Visibility - Silhouette Analysis

Certain key areas, locations, and buildings (monuments) are of specific importance to a city’s identity. We consider them as vantage points, points that offer us identity-constituting views important to the city, and thus also have qualities that need safeguarding.

The objects, either man-made or nature-given (like ridgelines), which are visible from a vantage point – or any location in a 3D city model – are easily identified if we put a light source into our model in this very same location, and observe the objects that are protruding from the shadows in the surrounding area.

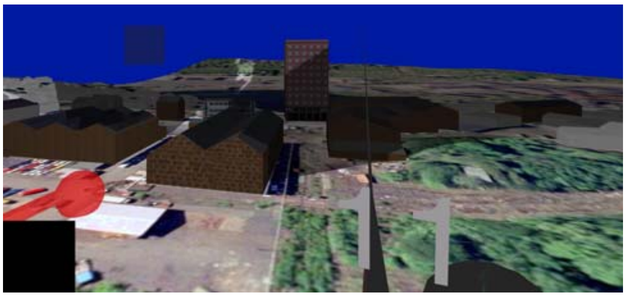

Figure 9. Eleven stories; the building is visible from chosen location (i.e. the light source, see semitransparent sphere to the left)

A new building or a development project that is not lit by the emission from the mentioned source does not create any silhouette seen from the chosen location – it is visually neutral to this location. So, if we want to shield certain locations from new visual impacts, the buildings under consideration must be kept within the shadow volumes.

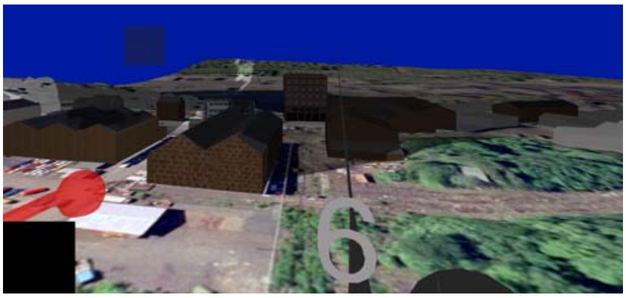

Figure 10. Six stories, the building is not visible from the chosen location

For enhanced use, analysis of this kind will have to be applied to a multitude of vantage points (Mak, 2004). Further, the analysis will be enhanced when used in combination with other forms of visual impact assessments (VIA).

4.5 Urban Design Comparison

When we need to compare alternative design solutions for a specific building site, like in urban design competitions, a visual swapping or fading (merging) function is soon asked for. This need can partly be satisfied by showing the alternatives on different screens (or in several windows) when using a non-stereographic visualisation. In a VR stereographic mode a multi-window principle is likely to result in quite bewildering visual impacts when viewing the model. Translucency and visual morphing within the same scene seem to be more applicable functions in a stereoscopic mode. For handling the task with an acceptable quality a double representation might be needed, by both handling voxel and surface data.

Christian Michelsen Research is at present addressing this challenge, testing multi-attribute volume visualisation that has been more commonly used within medicine and geosciences. In addition, the current activity within urban modelling is based on experience and user demands expressed in an evaluation project, of VR applied in urban design competitions (Erdal, 2003), done by CMR for The Norwegian Directorate of Public Construction and Property, Statsbygg. Prior to this study digital 3D urban models had been used by the jury members while analysing the qualities of the various entries in an urban design competition in the city centre of Oslo, the Vestbanen project.

4.6 Safety and Risk Analysis

Another important aspect in urban modelling is safety and risk analysis. One possible risk factor is leakage of explosive gases in connection with road accidents. The terrain and buildings in the area will have important impact on the formation of dangerous gas concentrations as well as on the pressure build-up in case of a gas explosion.

CMR has developed a FAVE based application for enhancing safety and risk communication, as well as training (Patel, 2004). The customers are the Norwegian oil companies Statoil and Norsk Hydro, and the application has been targeted on offshore oil and gas platforms, and onshore process plants. As a result of this work a mode of operation has been developed whereby experts and non-experts can communicate relevant major hazard risk topics in a more efficient way by collaborative work practices utilising new media. A run time interface has been implemented between the FAVE application and Computational Fluid Dynamics (CFD) simulator codes. We have tested this functionality for urban modelling purposes. Appropriate use cases include investigating potential risk factors during the planning phase. It is also possible to utilize this solution for evaluating the impact of pollution, given different traffic scenarios.

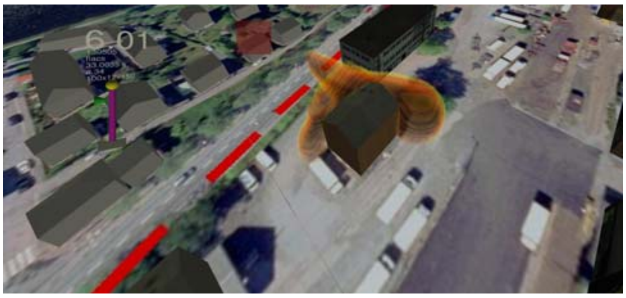

Figure 11. Visualisation of CFD simulation of gas dispersion

The CFD simulation tool used is FLACS (Flame Acceleration Simulator) (FLACS), a gas dispersion and explosion simulator developed by GexCon. The system makes it possible to assess the impact of the geometry of buildings both with regards to gas dispersion and to pressure build-up in case of a gas explosion. We chose an area at Kronstad, and defined a simulation grid covering an area approximately 330 x 50 meters. We defined a leak source close to the road passing by Kronstad to simulate propane leakage from a tank lorry. In addition to properties of the leak itself, like rate, temperature and gas composition, other conditions like wind can be controlled. The simulation can be controlled from within the VR system. The user can move the leak position to simulate a different scenario. The system also supports dynamic objects like doors that can be opened or closed during a simulation.

Several simulations can run simultaneously, and the user can switch between viewing the progress of them. A range of visualisation tools can be combined, including probes showing location specific values of variables, isosurfaces and direct volume visualisation. Figures 11 and 12 show the expanding gas cloud from the leak at Kronstad. The figures show a direct volume visualisation method using texture mapping on planes oriented perpendicular to the viewing direction. Colour and opacity mapping can be interactively controlled to get a better understanding of the simulation results.

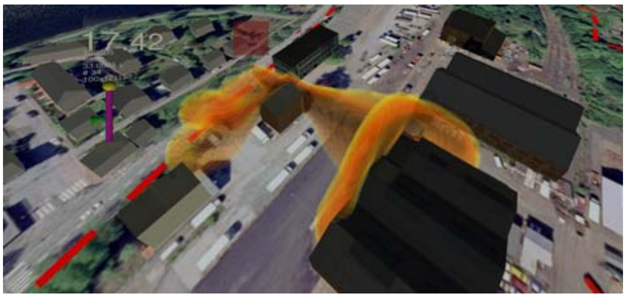

Figure 12. The same leakage as in figure 11 at a later time step

The interface to the CFD simulators is based on XML-technology. The scenario definition for the simulation is contained in an XML document. During the simulation, the VR application can send messages to the simulator for updating simulation parameters or defining new parameters. This is done using the XML Path Language, which is a language for addressing parts of an XML document. The CFD simulator notifies the VR application whenever new output is available, and the result data are communicated via a file interface.

5. CONCLUSION

In this paper we have presented further development and tests based on FAVE (Framework Architecture for Virtual Environments), a framework being developed by CMR. We have chosen 3D urban modelling as an interesting application area for these efforts. Further, we have made a selection of functionalities important to urban modelling, relevant to user demands, such as modular scaling, silhouette analysis, design comparison, safety and risk analysis. In addition, we have presented three functionality examples in our application prototype (FAVEum) based on the rapid prototyping qualities within FAVE. The chosen functionalities have been preliminary tested by using a simple 3D model of a part of Bergen, Norway.

Of specific relevance to safety and risk analysis, we have tested the ability of FAVE to handle VR simulations based on imported data from external software; in this case a simulation that makes it possible to assess the impact of the geometry of buildings both with regards to gas dispersion and to the consequences of a gas explosion.

Further steps in CMR’s efforts within the Urban Modelling field will be to develop additional functionality to the FAVEum prototype, and to apply the prototype to more complex data.

REFERENCES

Architectural Computing Organization (Website accessed 14 May 2005). http://www.architecturalcomputing.org/

Bierbaum, A., Just, C., Hartling, P., Meinert, K., Baker, A., and Cruz-Neira, C. (2001). "VR Juggler: A Virtual Platform for Virtual Reality Application Development." IEEE Virtual Reality, pp. 89-96.

Bowman, D., Kruijff, E., Joseph, J., LaViola., & Poupyrev, I. (2001). An introduction to 3D User Interface Design. *Presence*, Vol. 10, No. 1, Feb 2001, pp. 96-108.

Brooks, F. (2003). Three Great Challenges for Half-Century-Old Computer Science. *Journal of the ACM*, Vol. 50, No. 1, pp. 25-26.

Carey, R., Bell. G. (1997). *The Annotated VRML 97 Reference*. Addison Wesley Longman, Inc. Copyright 1997-1999 (Website accessed 13 May 2005). http://accad.osu.edu/~pgerstma/class/vnv/resources/info/AnnotatedVrmlRef/ch1.htm

CAVELib (Website accessed 14 May 2005). http://www.vrco.com/CAVELib/OverviewCAVELib.html

CrystalCG, Crystal Digital Technology Co., Ltd. (Website accessed 13 May 2005). http://usa.crystalcg.com/CRYSTAL.html

CumniCAD, Cumulative Index of Computer Aided Architectural Design (Website accessed 14 May 2005). http://cumincad.scix.net/

Erdal, J.O., Ulvang, D.M., Villanger, K.P, (2003). Evaluerings-prosjekt: VR-bruk ved arkitektkonkurranser og byutviklings-oppgaver (Report CMR-03-F50004).

Ervin, S., Hasbrouck, H. (2001). *Landscape Modeling*. 289p. McGraw-Hill Professional Publishing

FLACS Startpage (Website accessed 14 May 2005). http://www.gexcon.com/index.php?src=flacs/overview.html

Grosjean, J., Burkhardt, J., Coquillart, S., & Richard, P. (2002). "Command & Control Cube: a Shortcut Paradigm for Virtual Environments," ICMI'2002, Pittsburgh, USA, 14-16.10.2002.

He, E., Alimohideen, J., Eliason, J., Krishnaprasad, N., Leigh, J. Yu, O., and DeFanti, T. (2003). QUANTA: a Toolkit for High Performance Data Delivery Over Photonic Networks. *Future Generation Computer Systems*, 19(6): pp.919–933.

Jafari, M. (2003). Technological Advances in Evacuation Planning and Emergency Management: Current State of the Art (Website accessed 14 May 2005). http://www.cait.rutgers.edu/finalreports/EVAC-RU4474.pdf.

K2Vi, Key to virtual insight (Website accessed 14 May 2005). http://www.k2vi.com/

Lange, E., (1999). *Realität und computergestützte visuelle Simulation*. 176p. ORL-Bericht 106/1999, ETH, Zürich.

Mak et al. (2004). Developing a city skyline for Hong Kong using GIS and urban design guidelines, URISA (Website accessed 13 May 2005). http://www.urisa.org/Journal/Under_Review/mak/developing_a_city_skyline.htm

Martens, B., Brown, A., Turk, Z., (2001). Automated Classification of CAAD-related Publications: Conditions for Setting-Up a Keywording System. *ACADIA 22*

NMM, Nordisk Multimedia AS, for MIR Visuals ANS and the City of Bergen (Website accessed 24 May 2005) http://nmm.no/vod/ba/byutviklingskonferansen_768.wmv

Norwegian Mapping Authority. SOSI – Systematic Organisation of Spatial Information (Website accessed 13 May 2005). http://www.statkart.no/standard/sosi/html/div/sosi_eng.pdf

Olympic Games & Urban Development of Beijing (Website accessed 13 May 2005). http://www.bjghw.gov.cn/english/index.asp.

OpenGL Performer (Website accessed 13 May 2005). http://www.sgi.com/products/software/performer/

Open Inventor (Website accessed 13 May 2005). http://oss.sgi.com/projects/inventor/

Patel, D., Eliassen, I.K., Langeland, T., (2003) FAVE - A Framework Architecture for Virtual Environments. The 3rd CAVE-Programming Workshop, Helsinki, Finland 2003 (Website accessed 14 May 2005). http://eve.hut.fi/cavews2003/favePerfAppPaper.pdf

Patel, D., Eliassen, I.K., Langeland, T., (2004). FAVE, Framework Architecture for Virtual Environments. Use case; visualisation of online CFD simulations (Website accessed 14 May 2005). http://www.cmr.no/avd50/doc/faveSafety.pdf

SONY (2005), Navigation à la Sony, *Sony Magazine* (Website accessed 24 May 2005) http://www.sony.at/PageView.do?site=odw_de_AT&page=MagazineWithArticleFirstSection&articlesection=1&article=1109586900097&section=de_AT_Magazine_News

Springer, J. P., Tramberend, H., Frolich, B. On Scripting in Distributed Virtual Environments. IMK.VE. German National Research Center for Information Technology (Website accessed 14 May 2005). http://www.avango.de/

Statsbygg. Directorate of Public Construction and Property (Websites accessed 14 may 2005). http://www.statsbygg.no/english and http://www.statsbygg.no/prosjekter/vestbanen/dbafile948.html

Zlatanova, S. (2000). *3D GIS for urban development*. 222 p. - PhD thesis Graz University of Technology. (ITC Dissertation; 69)