Development and Application of Distributed Interactive Visualization Environment: Using Tamshui River and Baltic Sea Ruegen Terrain Models as an Example

Po-Huan Chiou
Institute of Information Management
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.
is83009@cis.nctu.edu.tw

Jaye-Jyh Tsai
Department of Computer and Information Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.
is85065@cis.nctu.edu.tw

Chen-An Huang
Institute of Computer Science and Information Engineering
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.
cahuang@csie.nctu.edu.tw

Shin-Jye Liang
National Center for High-Performance Computing
Hsinchu, Taiwan, R.O.C.
c00lsj00@nchc.gov.tw

Frank Molkenthin
Institut für Bauinformatik
Brandenburg University of Technology at Cottbus, Germany
frank@bauinf.tu-cottbus.de

Abstract

This paper presents the preliminary results of an ongoing joint Taiwanese - German (TaiGer) research project, which deals with the development and application of a WWW-based collaboration environment for distributed coastal engineering projects [1]. We focus on the Distributed Interactive Visualization Environment (DIVE) developments as well as the application of Geographic Information System (GIS) and 3D Virtual Worlds. The DIVE system consists two components: the display module and net module. Display module employs both OpenGL and VRML for visualizing the 3D terrain models and the physical behavior of a system. Net module uses the CORBA (Common Object Request Broker Architecture), Java and EAI (External Authoring Interface) for the distributed communications. The developed system integrates the interactive visualization of 3D terrain models and flow simulation scenarios with tools for net-based communications. The system has been applied for the visualization/navigation of Tamshui River basin and Baltic Sea nearby Ruegen for distributed multiusers via Internet. More details of our research can be found in Reference [1, 2]. In this way, the approach has a great potential for collaborative visualization and other applications in modern distributed engineering projects.

Key Words: Distributed Interactive Visualization Environment, Geographic Information System, Collaborative Visualization

Introduction

Collaborative visualization is an active area of research in computer science, and presents great potential for various applications, such as distance learning, tele-conference, tele-medicine and etc. [3, 4]. The distributed and interactive visualization environment (DIVE) is especially tuned to an internet-based multiuser visualization system where participants navigate in a 3D world and interact with other users and applications. Many different techniques and implementations have been proposed. In this study, a DIVE is developed using various computer graphic tools (OpenGL and VRML) and network communication techniques (CORBA, Java and EAI) [1, 2]. Their applications and performance are scrutinized.

Our objective is to create a WWW-based computational infrastructure, which integrates the geographic information system (GIS), hydrodynamic model (numerical modelling), and distributed and interactive visualization environment (DIVE) for visualizing and predicting events. We use Tamshui River and Baltic Sea nearby island of Ruegen as the application fields. The hydrodynamic model incorporates circulation, salinity, and temperature patterns in 3D, winds, tides, upstream inflows, and seawater-freshwater interactions. The developed WWW-based tools provide the basis for the remote researchers work on the collaborative projects with heterogeneous computer platforms by integrating the interactive visualization/navigation of 3D terrain models with flow simulation scenarios. More details of results can be found at and http://bauinf.tu-cottbus.de/taiger/ or http://www.nchc.gov.tw/RESEARCH/TaiGer/ [1] and http://140.110.34.5/tamshui/ [2]. This approach opens the great potential of the WWW for collaborative visualization and other distributed applications in science and engineering.

Application Fields

Two coastal regions, one in Taiwan (The Tamshui River basin) and the other in Germany (The Baltic Sea nearby island of Ruegen), are used as the application fields. The importance and physical complex of the problems are described as following:

<u>Tamshui River basin</u>

The Tamshui river which locates in the northwest of the island is the second largest river in Taiwan, as depicted in Figure 1. Its three main tributaries - Hsindtien Stream, Tahan Stream, and Keelung River - pass the metropolitan Taipei which is the center of politics, economy and culture of Taiwan. Owing to the dense population and rapid economic development, dirty water, air and soil pose a threat to public health in regular seasons. Furthermore, inundations and mudslides often cause death and losses of property during the typhoon and rainy seasons. Public concerns over these situations have prompted.

<u>Wind Wadden Area in the Baltic Sea</u>

The German coast at the Baltic Sea is shaped by morphodynamic processes due to wind-induced currents and waves. Different directions and intensities of sediment transport generate complex shore profiles. In the region of

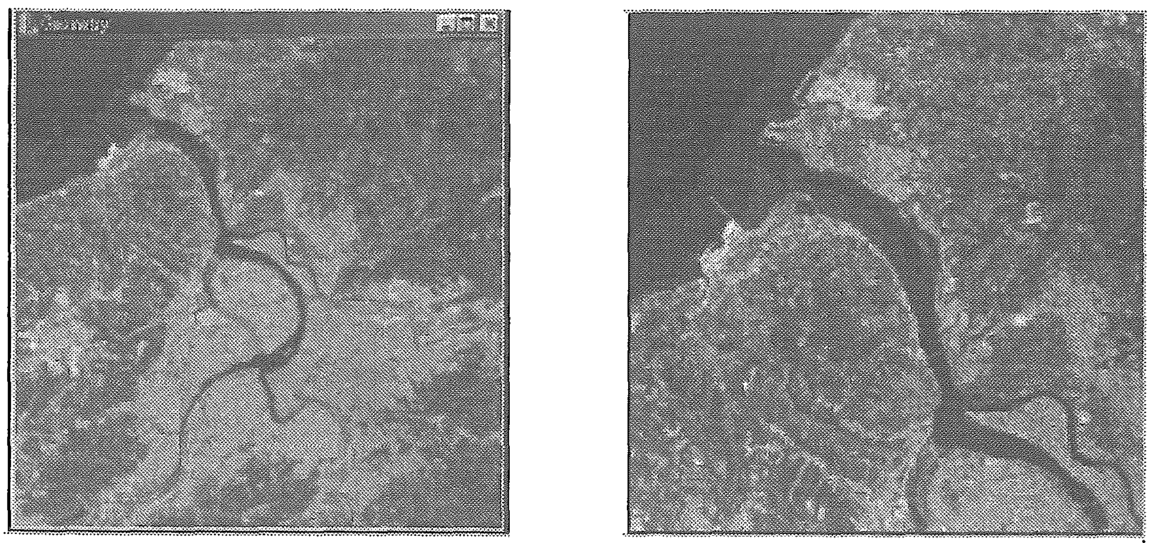


Figure 1 The SPOT satellite images of Tamshui River basin

the island Ruegen, significant sediment accumulation near the Strelasund and the lagoon area Darß/Zingst forms shallow sand bars of only a few decimeters of depth but several kilometers in extension. These morphodynamic bodies, called wind wadden, are either submerged or dry depending on the weather conditions and the dynamics of the Baltic Sea itself. Figure 2 depicts the Satellite images of wind wadden area nearby the island of Ruegen. The morphodynamic behavior of these areas is not sufficiently understood up to now. Thus numerical modeling of the morphodynamic processes in space and time is a key research for the coastal management in this region.

Software Technologies

To develop the WWW-based tools for remote visualization and collaboration, three technologies has been selected from the wide spectrum of Information Communication Technology (ICT) and basis software applications:

GIS Technology: ARC/INFO I/O interfaces
Visualization in 3D Worlds: OpenGL and VRML
Network and WWW Technology: HTML, Java Script, Java, CORBA, and EAI

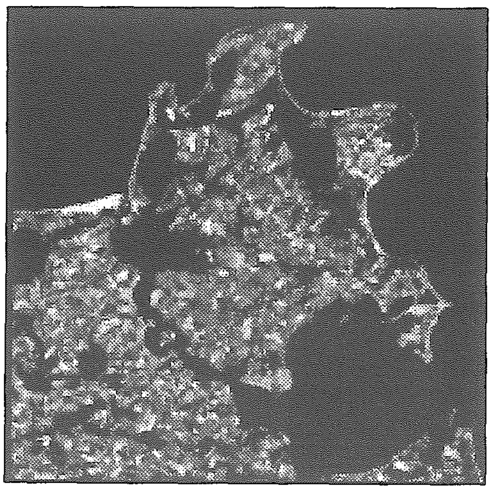
The application of these technical backgrounds in respect to the tasks in coastal engineering is described as follows:

GIS Technology

Geographic Information Systems (GIS) can be regarded as the high-tech equivalent of the map. It represents the means of locating ourselves in relation to the world around us. Conventional GIS focuses on the 1D/2D applications and visualization due to the computer hardware and software resources constraint. It's outlooks and operations are relatively static and passive compared with the modern GIS. Modern GIS supports several kind of information from scalar values up to 3D world and multimedia objects. Besides the cathegrapical feature for maps and satellite images, GIS offers a wide range of dynamic and interactive analysis, visualization and documentation features for spatial information. ArcInfo is today's premier GIS software. Several tools are designed to facilitate the GIS raw data from ArcInfo, such as, processing a large amount of geographic information, texturing the satellite image, and embedding watercourse and physical behavior into the terrain coordinates. In this way, we present a new way to visualize 3D terrain models which contain rich and detailed GIS information, and integrate it with the analysis and visualization of simulations from physical modelling.

Visualization

OpenGL is the premier environment for developing portable, interactive 2D and 3D graphics applications.



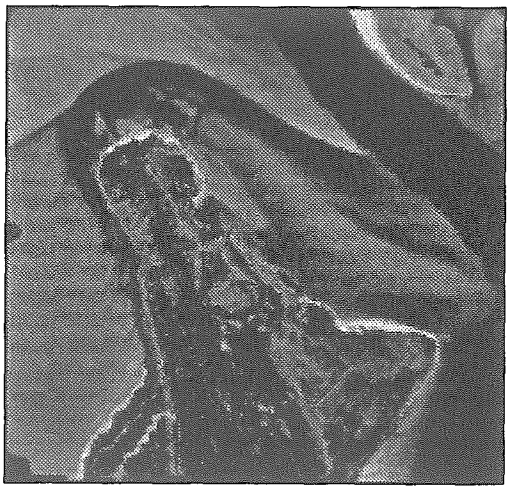


Figure 2 Satellite images of wind wadden area nearby the island of Ruegen.

Since its introduction in 1992, OpenGL has become the industry's most widely used and supported 2D and 3D graphics application programming interface (API), brings thousands of applications to a wide variety of computer platforms. OpenGL fosters innovation and speeds application development by incorporating a broad set of rendering, texture mapping, special effects, and other powerful visualization functions. Though it is easy to represent the 3D terrain models with OpenGL, OpenGL is written with C++ language, and therefore platform-dependent.

VRML (Virtual Reality Modeling Language) is an open, extensible, industry-standard scene description language for 3D worlds on the Internet. With the VRML browser, we can view distributed and interactive 3D objects that are rich with text, images, animation, sound, music, and even video. VRML 1.0 supports worlds with relatively simple animations, while VRML 2.0 supports complex 3D animations, simulations, and behaviors by allowing Java and JavaScript programmers to write scripts that act upon VRML objects.

We use both OpenGL and VRML to visualize/navigate the 3D terrain models and the physical behavior of a system. The applications and performance of both methods are scrutinized.

Network and WWW Technology

The WWW is an open and unstructured nonlinear hypermedia information system. It is based on nonlinear linked HTML documents, in which different types of WWW objects can be embedded. Besides multimedia objects like pictures, videos and audio sequence, interactive components can also be integrated. Three kind of interactive components has been used in our study: Java-Script, Java application and VRML objects.

JavaScript is an interpreter language direct embedded in HTML pages. Using the browser interfaces, JavaScript has directly access to the browser components and the document pages content. Together with the form tags of HTML, JavaScript is a powerful environment for dynamic interactive documents on the client side.

Java is an object-oriented programming language with all necessary features for developing a complete hydroinformatics system [3]. Several class packages for graphical user interfaces, 3D visualization, inter-process communication like RMI and COBRA, data base access and others are part of the standard Java development kit (JDK). Main advantage and feature of the Java technology is the platform-independence. Java programs run under all operation systems as well as inside WWW browser. (concept of *the virtual machines* [4]). Java programs can be easily embedded as applets in HTML pages. The server/client-technology is implemented by Java applets and Java servlets, and enables an efficient development of WWW-based science and engineering applications. Interfaces for VRML models allow an efficient control of 3D objects.

These WWW technologies (HTML, JavaScript, Java and VRML) allow the development of modern WWW-based software systems, designed for a distributed application in the WWW. This leads to an integrated analysis, visualization and documentation as an interactive WWW documents. Figure 3 shows a typical example for such kind of information system for hydroinformatics applications.

Methodology

In this study both OpenGL and VRML are employed to visualize/navigate the topography/bathymetry as well as the physical behavior of a system in a 3D world. OpenGL is a popular graphics library because of its good rendering performance and visualization quality. However, OpenGL is written in Visual C++ language, therefore platform-dependent, and different software for different.

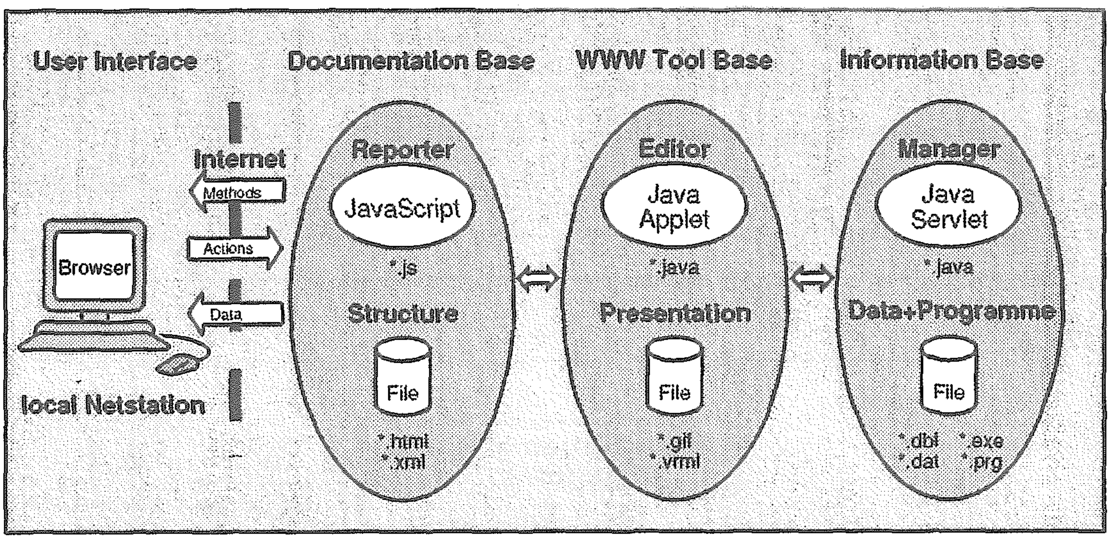


Figure 3 Interactive WWW document for analysis and visualization.

platform is needed. Whereas the 3D terrain models are easy to convert to VRML files, and the interactive visualization can be operated consistently with the standard web tools for VRML. The performance of VRML is generally not as good as OpenGL in terms of speed (frames per sec.) and visualization quality. Therefore two versions of DIVE has been developed: the DIVE–OpenGL and DIVE–VRML. The application and performance of both methods are investigated. Main features and functionalities are briefed in the following:

DIVE–OpenGL

We used the OpenGL to develop the *display* module, and CORBA for the distributed and interactive communications in the DIVE-OpenGL. The state of view observed mainly determined by (1) the position, (2) the direction of view point, and (3) the normal direction of the view point of the observer. We use *X*, *Y*, and *Z* to represent the position of the observer; *Heading* and *Pitch* for the direction of the view point of the observation; *Roll* for the normal direction of the view point of the observer. Therefore, the relationship between the state of view of the observer and the above six control parameters can be expressed as follow

$$I(x, y, z) = I(X, Y, Z) \quad (1)$$
$$\sin(Heading)x\cos(Pitch)$$
$$+ \quad \sin(Pitch)$$
$$\cos(Heading)x\cos(Pitch)$$

By changing the above six parameters, we can control the state of view, and therefore, navigate the 3D terrain models. This action is done by the **setView** function, which is demonstrated with program in the Table 1:

```
!
X = 0.0;
Y = 400.0;
Z = -60.0;
Roll = 0.0;
Pitch = -90.0;
Heading = 0.0;
setView(Heading,Pitch,Roll,X,Y,Z);
OpenGL1->Repaint();
!
```

Table 1 Main body of the setView function.

When the client with token changes the control parameters, and passes the information to other users through the server, then the users can update the state of view by the **updateView** function. This action is done by the **updateView** function, which is demonstrated with program in the Table 2:

```
!
X = 0.0;
Y = 400.0;
Z = -60.0;
Roll = 0.0;
Pitch = -90.0;
Heading = 0.0;
updateView(Heading,Pitch,Roll,X,Y,Z);
OpenGL1->Repaint();
!
```

Table2 Main body of the updateView function.

It should be noted that server only broadcasts the information of changing of the control parameters to all users, and all the computations are performed locally. Therefore, the communications load between the server and clients is very low – the actual size of data for communications is less than 100 bytes. By repeating the above setView/updateView actions, as depicted in Figure 4, the distributed and interactive visualization becomes feasible.

Figure 5 illustrates the relationship between clients and server as well as *display* and *net* modules of the distributed interactive visualization environment.

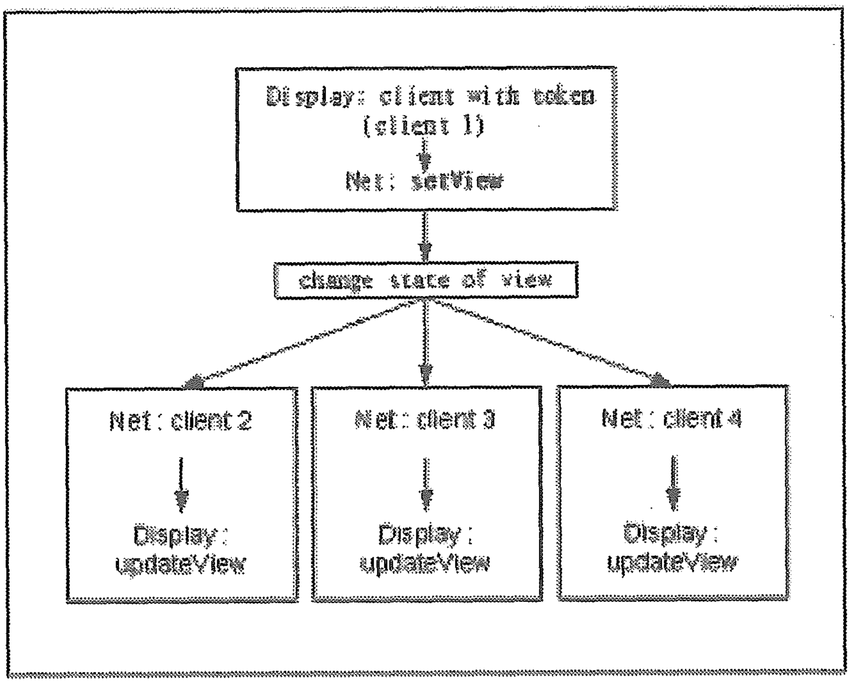


Figure 4 Relations between *display* and *net* module.

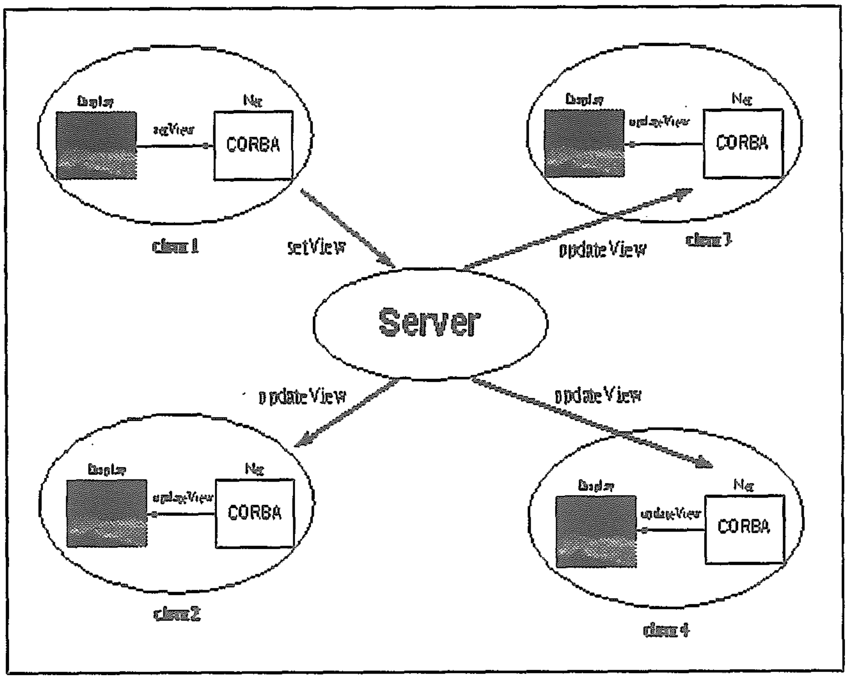


Figure 5 Architecture of distributed interactive visualization environment.

DIVE-VRML

3D terrain models are easy to convert to VRML objects. The interactive visualization can be operated consistently with the standard web tools. We use VRML for the visualization and navigation of 3D terrain model, Java for distributed communications, and EAI to control the contents of a VRML browser window in DIVE-VRML. Noted that DIVE-VRML is easy to maintain and extended for future development comparing with DIVE-OpenGL.

Results

Application of DIVE-OpenGL and DIVE-VRML for two coastal regions, one in Taiwan (The Tamshui River basin) and the other in Germany (The Baltic Sea nearby island of Ruegen), are described as following:

DIVE - OpenGL

Figure 6 presents the interface of the DIVE-OpenGL. It contains logging in, file managing, converting 2D image to 3D object, chatting, zooming and moving navigation functionalities. Local window denotes the state of view for the local user, and global window denotes the updated state of view for all users. By changing the control parameters, operating on the 3D object locally, we can collaboratively visualize/navigate the 3D terrain model.

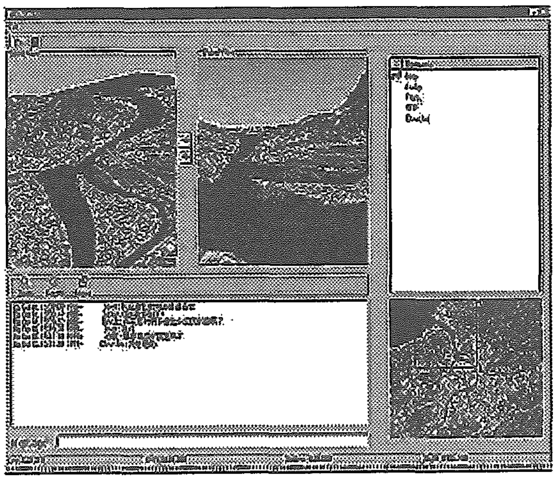


Figure 6 The interface of DIVE-OpenGL.

One of an example demonstrates the fly-by navigation of the 3D terrain model of Tamshui River basin is shown in Figure 7.

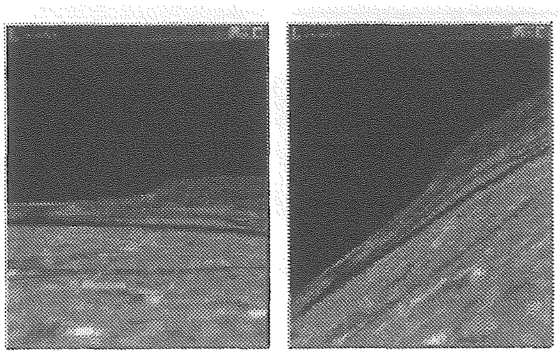


Figure 7 A demonstration of fly-by for Tamshui River basin terrain model.

DIVE-VRML

Figure 8 represents a conceptual application of the DIVE-VRML for Tamshui River basin. Using the standard VRML, visualization and navigation can be easily accessible with the WWW. We use EAI to control the contents of a VRML browser window embedded in a web page from a Java applet on the same page.

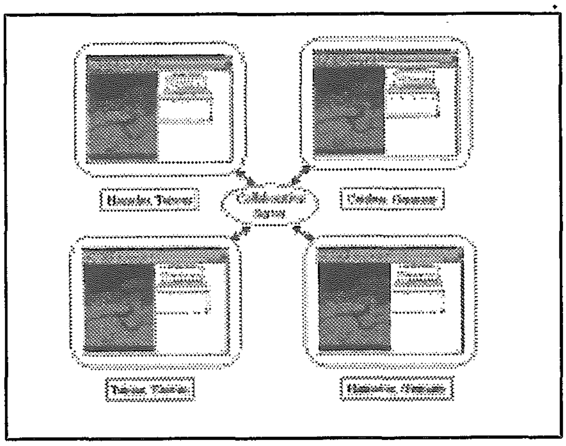


Figure 8 Application of DIVE-VRML for remote multiusers.

Figure 9 shows the integration of 3D terrain model with the flow simulation scenario. The hydrodynamic model incorporates circulation, salinity, and temperature patterns in 3D, winds, tides, upstream inflows, and seawater-freshwater interactions. The velocity field on the surface of the water is shown.

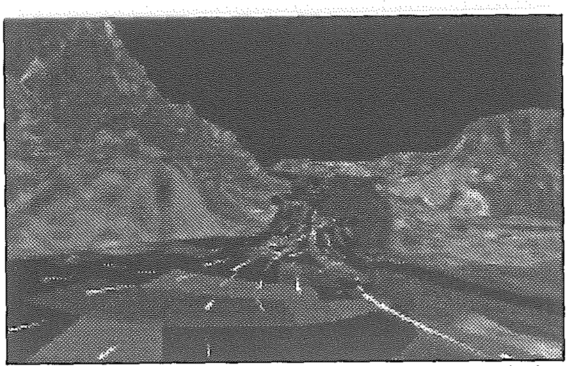


Figure 9 DIVE – VRML for a 3D world and flow simulation scenario.

XF4-VRML

XF4 is a pre- and post-processing tool for 2D hydrodynamic simulations (based on finite elements) developed for projects at the University of Hannover and the Brandenburg University of Technology at Cottbus, Germany [5]. This tool enables an efficient set-up of model runs including various features like grid modeling, time series analysis, analysis and visualization of 2D flow fields. Figure 10 shows the 2D X-Window based visualization of finite elements for Baltic Sea nearby island of Ruegen.

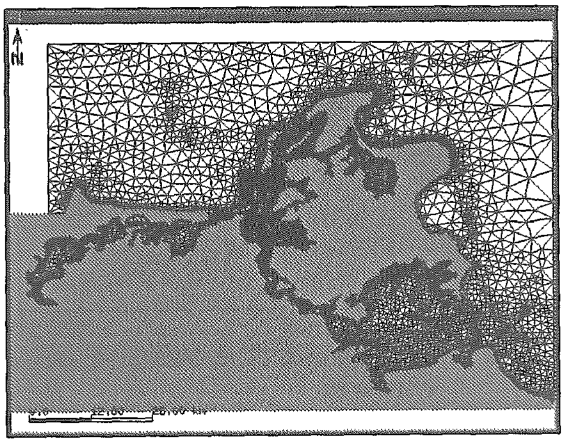


Figure 10 2D finite elements for the Baltic Sea nearby Ruegen.

For the purpose of 3D visualization, the XF4 was extended to a new module, XF4-VRML, which generates VRML files. This module includes interfaces to transfer the digital terrain models, the finite element grids, and the physical behavior of the water (e.g. water surface level and velocity fields) in 3D objects. It also provides the function to integrate additional 3D objects for the orientation of the end user like buildings, light houses and bridges. One such example is illustrated in Figure 11 for the area near Hiddensee Island which is located in the west of Ruegen.

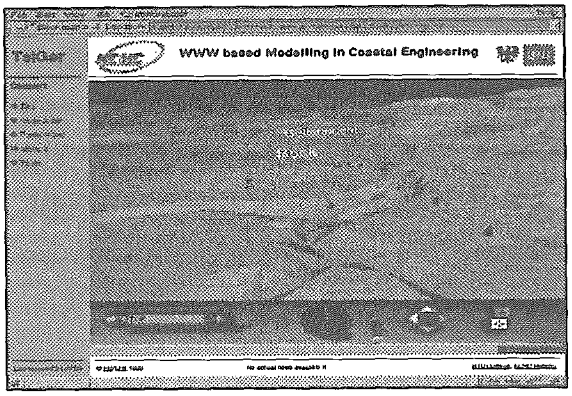


Figure 11 3D terrain model with light houses and text markers objects.

Figure 12 depicts the velocity vectors obtained by XF4 physical modelling. The integration of 3D topography, finite element grids and physical results in 3D virtual world is demonstrated in Figure 13.

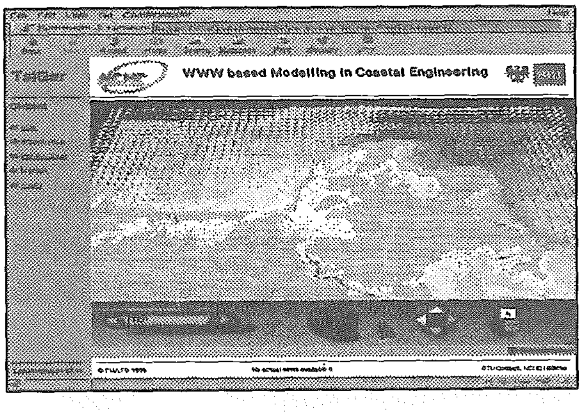


Figure 12 Velocity vectors obtained by XF4 physical modelling.

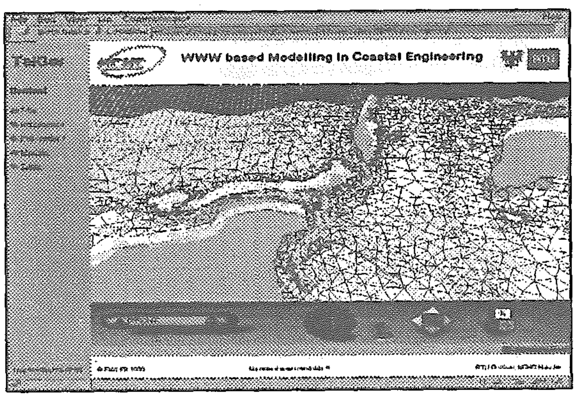


Figure 13 Integration of the topography, finite element grids and physical results in a 3D virtual world.

WWW-TGE

To provide the full advantage of modern Internet technology for the WWW, an existing software system has to be re-designed and re-implemented in an iterative process. The grid modelling module of XF4 has been transferred toward a WWW-based grid tool called WWW-TGE (triangle grid editor). This tool is completely developed in Java, and can be applied as applications as well as applets in the WWW. With such tool, an user can get access to grids for visualization and editing independent from his location and can easily share them for collaboration on WWW server. Figure 14 shows part of the WWW grid editor running in a Netscape WWW browser.

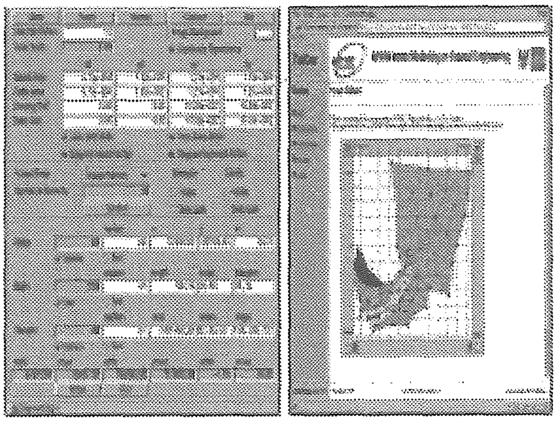


Figure 14 WWW-based Grid Editor as a Java Applet.

Conclusion

Result of the TaiGer project is a WWW-based tool set for a selected domain of coastal engineering applications. This kind of hydroinformatics systems enables a heterogeneous group of experts at different location to collaborate in physical modelling in coastal engineering over the Internet. The user is independent from his local computer facilities. The Internet becomes the computer, the WWW becomes the operation system. Both are available all over the world via Internet.

In this study, two versions of distributed and interactive visualization environment system are developed: DIVE–OpenGL and DIVE–VRML. DIVE-OpenGL has better display performance and visualization quality comparing with DIVE–VRML. However, OpenGL is written in Visual C++ language, therefore platform-dependent and different software for different platform is needed. CORBA is employed for distributed communications in DIVE-OpenGL. 3D terrain models are easy to convert to VRML files, and the interactive visualization is operated consistently with the standard web tools for the DIVE-VRML. We used Java for distributed communications and EAI to control the contents of a VRML browser window embedded in a web page from a Java applet on the same page. XF4-VRML is an example for the integration of modern WWW technology like VRML with existing software. The new WWW-TGE demonstrates the potential of the WWW for a new way of a net-based engineering collaboration.

Our OpenGL and VRML models enable interested individual and policymakers to observe firsthand the delicate interconnections among physical behavior and the 3D virtual worlds. The project concept and result can be used as a prototype for further WWW-based tools for supporting various engineering applications. These tools will be available for various distributed and interactive visualization cooperative projects. The developed systems will efficiently support scientist and engineers from different countries for common collaboration projects in science and engineering applications.

Acknowledgement

The described work in this ongoing project is conducted in a closed cooperation between the National Center for High-Performance Computing (NCHC), Hsinchu, Taiwan, R.O.C. and the Institut für Bauinformatik, Brandenburg University of Technology at Cottbus, Germany. We would like to thank the National Science Council of Taiwan, R.O.C. (Grant No. 8801PPP14 and NSC 88-2611-E-321-001) and the Deutscher Akademischer

Austauschdienst of Germany (Grant No. D19827246) for the financial support of our common research work. Many helpful discussions with Mr. Jiunn-Horng Lee of NCHC are also appreciated.

References

[1] Molkenthin, Frank, Liang, Shin-Jye, and Lee, Jiunn-Horng, WWW Based Collaboration in Coastal Engineering: Integrated Analysis, Visualization and Documentation, Proceedings of the 2nd German-Chinese Joint Seminar on Recent Development in Coastal Engineering – Sustainable Development in the Coastal Zone, Tainan, Taiwan, R.O.C., 153-169, 1999. http://www.bauinf.tu-cottbus.de/taiger/ and http://www.nchc.gov.tw/RESEARCH/TaiGer/

[2] Liang, Shin-Jye, Lee, Jiunn-Horng, and Huang, Chen-An, A GIS-Based 3D Hydrodynamic Modelling for Tamshui River System, 1999 Computer Technology Applications for Civil and Hydraulic Engineering Conference, Taichung, Taiwan, 1999. (accpeted) http://140.110.34.5/tamshui/.

[3] Abbott, M. B., Hydroinformatics: A Copernican Revolution in Hydraulics in IAHR, Journal of Hydraulic Research, Volume 32, 'Hydroinformatics', 1994.

[4] Molkenthin, F. and Holz, K.-P., Working Process in a Virtual Institute, Int. Conf. Hydroinformatics 98, Copenhagen, Denmark, pp. 941-948, 1998, A. A. Balkema, Rotterdam, The Netherlands. http://www.bauinf.tu-cottbus.de/publikationen/HIC98/HIC_8.html/.

[5] Holz, K.-P. et al., The TICAD-Toolbox Applied to Coastal Engineerings Problems, in Blain/Ouazar (eds.), Hydraulic Engineering Software Application, Computational Mechanics Publications, Boston, USA, 1990.