European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Computational Fluid Dynamics Conference 2001
Swansea, Wales, UK, 4-7 September 2001

VIRTUAL ENVIRONMENTS FOR FLOW SIMULATIONS

Lakshmi Sastry[*], Venkat V. S. S. Sastry[†], Michael D. Wilson[*] and David Boyd[*]

[*]Information Technology Department,
CLRC Rutherford Appleton Laboratory,
Chilton, Didcot, OX11 0QX UK
email:{m.sastry|m.d.wilson|d.r.s.boyd}@rl.ac.uk, web page: http://www.itd.clrc.ac.uk

[†] Applied Mathematics and Operational Research Group, Engineering Systems Department
Cranfield University, RMCS Shrivenham, Swindon, Wilts, SN6 8LA, UK
email:sastry@rmcs.cranfield.ac.uk, web page: http://www.rmcs.cranfiled.ac.uk

Key words: Virtual reality, flow simulation, human computer interaction.

Abstract. *This paper describes a software framework, based on virtual reality, for real-time immersive visualization of and interacting with flow simulation data. The objective of applying virtual reality techniques to flow simulation data is to explore the efficacy of combining the real-time interaction afforded by virtual reality based interactions with conventional visualization algorithms to gain better insight into flow features. Initial results from our earlier studies indicated a requirement for interrogative tools for the user-centred exploration in such environments. Such an interaction framework is afforded by the instantiation o an interaction toolkit within the software environement for visualization. The components of the virtual reality based visualization framework is illustrated using two examples from computational fluid dynamics. The dynamic use of the 3D interactions are demonstrated.*

1 INTRODUCTION

Virtual reality (VR) is characterized by

- its technology to interface to novel and more intuitive input and output devices which facilitates the user and his application to share the same data space.
- more intuitive and natural multi-modal metaphors of interaction that form seamless, common and continuous communication between the user and the application
- real-time visualization that supports the natural capabilities of the human visual information processing

Together, the above three characteristics hold the promise to provide better engagement with the application rather than with the human-computer interface[1]. Indeed the increasing use of VR to real world experiential applications such as navigation through cyber cities[2], therapeutic treatments[3] and a host of training applications[4, 5] and some evaluation studies bear this out. Application of VR interaction technology in industrial contexts such as automotive and engineering product life cycle has been reported to show productivity gains[6]. Hence, the raised expectation is that the application of VR to non-experiential, non-real world situations such as scientific data visualization can render better understanding and more productive exploration of the application domain.

In scientific data visualization, the data is determined by the underlying natural phenomena being studied. These are typically dynamic in nature and undergo rapid and complex changes over short time and space. The associated spatio-temporal relationship in the data is closely coupled and is inherently non-linear. Given the results that virtual environments (VE), user interfaces based on VR interaction techniques, convey strong spatial cues, previous studies have attempted to exploit VR to explore computational data spaces[7, 8]. However, the limitations of such studies have been that they are application or toolkit specific, lacking in generality. Moreover, the initial focus had been primarily on exploring the feasibility of the application of VR technique for such applications and addressing the ensuing data management and presentation issues. Our earlier attempts[9] have identified a clear need for enriching these environments with interactions based on VR techniques. These interactions are beneficial for the successful deployment of such applications because the user not only need to navigate through the data but to be able to effectively control and query the data parameters dynamically or to issue commands to compute relevant derived data values as and when appropriate. In addition he would have to be able to switch between egocentric and exocentric navigation depending on the task and the scale of the domain. What is required is a generic toolkit framework to facilitate these interactions. In this paper we describe an instantiation of the INQUSITIVE interaction toolkit[10] for developing virtual environments for flow simulation applications.

Section 2 describes the components of the framework, Section 3 describes two examples drawn from computational fluid dynamics. Observations on future work are highlighted with a summary in Section 4.

2 GENERIC FRAMEWORK FOR SCIENTIFIC VISUALIZATION

VR, as a visualization and interaction metaphor facilitates real-time interactive exploration of data in 3D. Within a virtual environment the user is an essential participant rather than an observer and, by being directly immersed in the application database, has a truly user-centred view of the data (Figure 1). VR systems also provide the ability to visualize 3D data in real-time with rich spatial and depth cues. There is inherent support for generating geometrical objects in a format suitable for efficient rendering. By tracking the user's position and orientation, it is possible to adopt techniques such as level of detail and view portals, to make substantial economies on the quantity of data presented visually as geometric objects in each frame without loss of context.

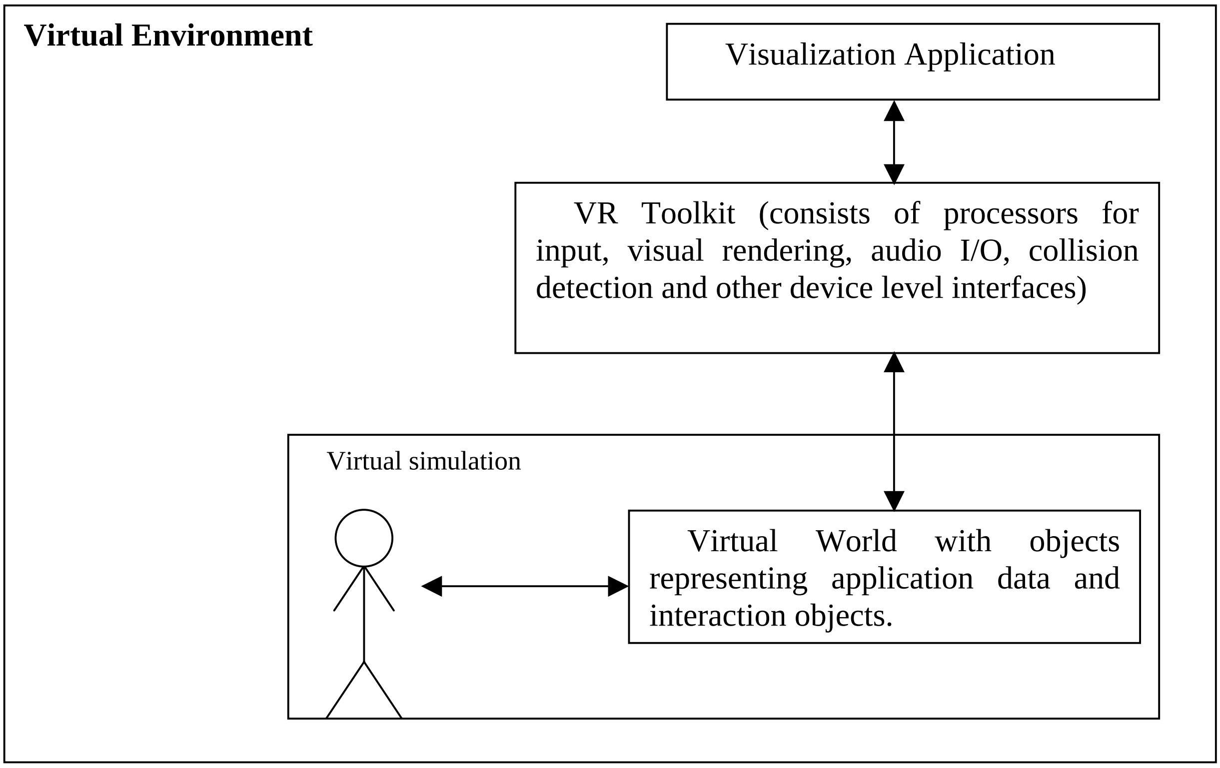

Figure 1. Schematic of a virtual environment for visualization

In general there are three approaches to the application of VR to visualization. These are: development of dedicated systems, adding VR modules to an existing visualization system or adding visualization modules to an existing VR system and the coupling of existing VR and visualization systems[7, 8, 9]. Our approach is to add visualization modules to an existing VR kernel. This approach enables us to select computationally efficient algorithms, to adopt appropriate rendering techniques as well as to customize novel interaction metaphors appropriate for the application and the user. This is done using a generic framework (Figure 2) using the Maverik VR kernel[11]. Such generic framework will aid the rapid development of interaction rich virtual environments for scientific visualization. This in turn may provide opportunities to conduct usability studies to explore issues further and indeed to be able to assess and quantify the utility of VR to such applications.

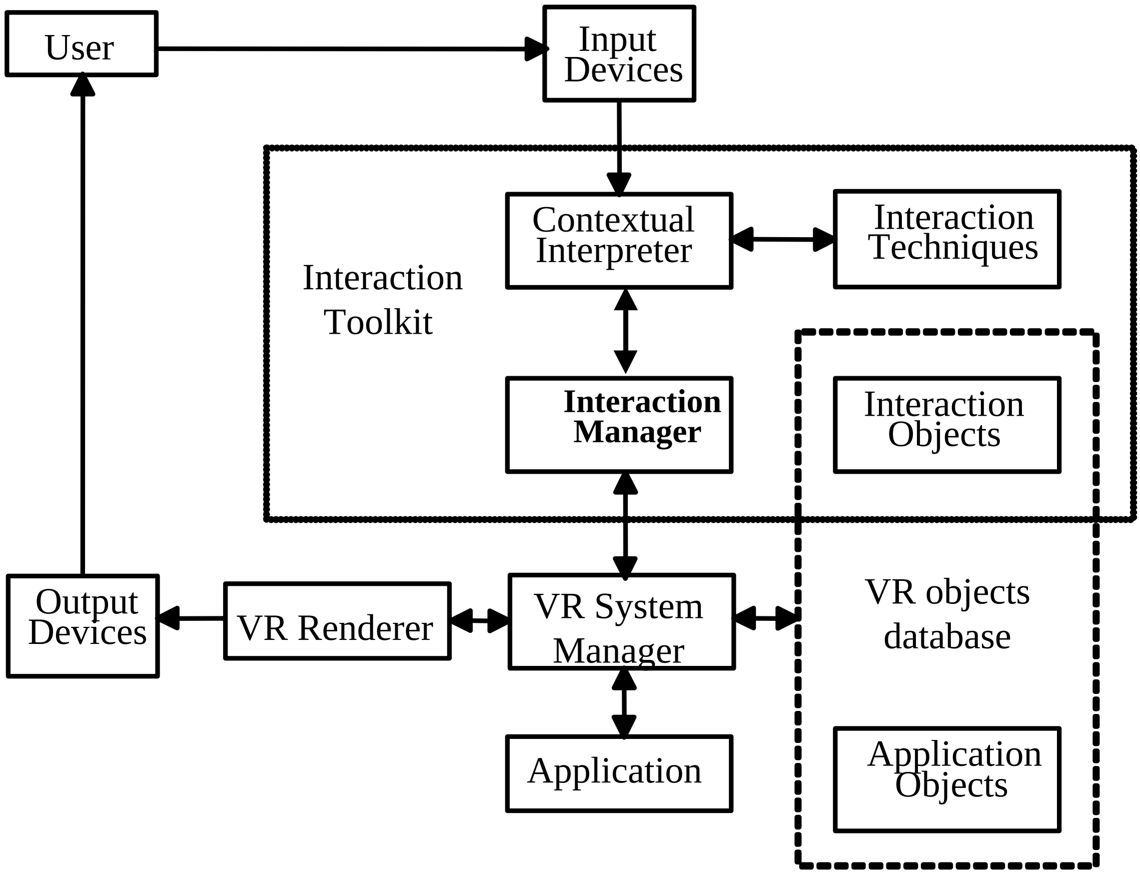

Figure 2. Framework with embedded visualization objects within a VR kernel and the interaction toolkit.

2.1 Components of a VR-based framework for visualization

Components of such a generic framework include modules for

- Geometric representation
  - Contextual geometry of the problem domain as well as relevant objects such as stream tubes, isosurfaces.
- Methods of Interaction, for example[12],
  - Navigation techniques
    - Ego-centric vs exocentric
    - Point-fly vs gaze-directed
  - Communication with application

    - Define/set parameters (e.g. specifying volume of interest)
    - Query data (e.g. what is the velocity value at this point)
    - Issue application commands (e.g. compute derived data such as vorticity)
- Real-time visualization support
  - Data handling strategies, especially important for large and time-varying data sets.

Figure 2 provides a diagrammatic overview of the above components and their inter-relationships, based on the Maverik VR kernel and the INQUISITIVE interaction toolkit. The first component of the runtime interaction framework is a *Contextual Interpreter* which

- obtains the measures and triggers from devices,
- convert these into the VE co-ordinate system, taking into account modality, device tolerance,
- takes account of the dynamic constraints and current state of the interaction objects,
- interprets the measures and triggers in that context,
- calls the appropriate interaction techniques to generate the event tokens,
- The Contextual Interpreter is independent of the host VR system.

The second component of the runtime interaction framework is an *Interaction Manager* which

- monitors the changing state of user interaction within the VE,
- receives the event tokens from the Contextual Interpreter,
- Queries the current state within the VR system's runtime object database,
- Communicates the update required,
- The operation of the Interaction Manager must be customised for each host VR system.

The Interaction Techniques module provides the necessary algorithms to interpret the input data into required interaction techniques. The Interaction Objects are a library or toolbox of widgets, which have a visual representation with embedded objects behaviour.

For computational fluid dynamic applications, the application objects will typically consist of contour plots, isosurfaces etc. In Maverik these are defined as new application specific classes with their own computing, rendering and behaviour methods. The interaction with such an application would be to fly through the flow domain, select seed points etc. The interaction toolkit component allows the user to dynamically select the desired interaction techniques to carry out the above tasks. For instance, he may decide to select a “rake” object from the interaction toolkit using his virtual hand to place a line of seed points for generating stream ribbons. He may decide to use gaze-directed navigation for flying through the flow field.

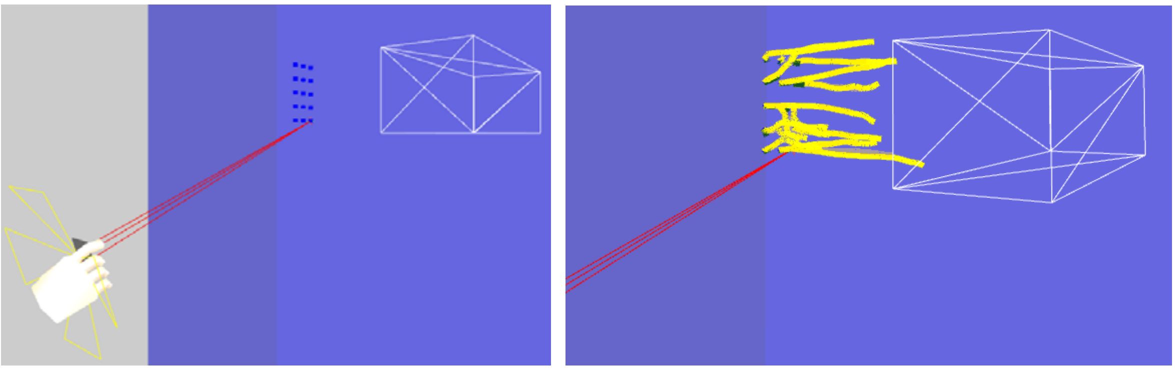

Figure 3. Virtual environment for large eddy simulation of the flow past a square cylinder, mounted transverse to the flow. Rake of seed points placed behind the cylinder by virtual-hand (left), and the stream ribbons emanating from the seed points (right).

The user can also interact with the raw application data. For example, he may select a desired flow domain and use a widget such as Data Reader to display computational quantities such as pressure etc. (Figure 4). The Data Reader is defined to handle both scalar and vector data, and display the quantities in an appropriate manner.

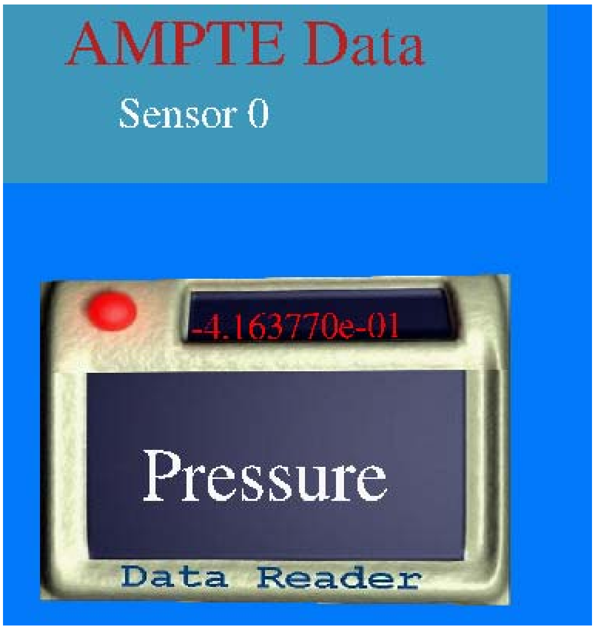

Figure 4. The Data Reader displaying a scalar field value as the user navigates through the flow field.

In the next section, we describe two applications from the domain of computational fluid dynamics that are implemented using the framework described above.

3 VIRTUAL ENVIRONMENTS FOR COMPUTATIONAL FLUID DYNAMICS APPLICATIONS

There is a need in a wide rage of applications to understand the space/time interaction in the core of turbulent flows. For example propagation of shock waves, internal flows such as those in aircraft engines, turbulent fluid flow in channels such as riverbeds.

Numerical simulations of unsteady flow such as large eddy simulations (LES) produce gigabytes of data for every computational step which are hard to comprehend as numbers but still difficult using two dimensional visualization using particle path tracing, path ribbons, path lines, vortex tubes or isosurfaces of scalar values.

Three-dimensional visualization of the simulation from within a virtual environment allows users to fully navigate through the simulation results in both space and time. Conventional CFD simulations and visualization tools produce the data to be presented, while the virtual reality presentation tools handle the rendering with sufficient performance to support real time interaction. This allows users to:

- Use the human visual system's innate abilities to perceive and consider objects moving in three-dimensional space to their full capacity to facilitate insight into the phenomena being viewed.
- Navigate inside the flow field as it passes by the viewpoint.
- View the fluid flow from either the particle as it moves (Euler), or from a fixed point as the flow passes by (Lagrange).
- Stretch the time dimension to view the complexities of the flow in slow motion.
- Interactively vary the parameters controlling the simulation whilst observing the resultant flow to investigate their effects on particular views.

This flexibility allows users to move from the laminar flow into the turbulent flow, past the point of detachment as far back as the seed point of the eddy, experiencing each stage in the eddy's lifecycle as they do so.

3.1 Two case studies

The two data sets, (Figure 5) that have been used to demonstrate the interaction components are from computational models of

- sea sediment movement over rough sand ripples that cause erosion of potential flood defences[13]. The computational grid is of size 82x34x20.
- large eddy simulation which are responsible for the maximum momentum and energy transfer that characterizes turbulent flows[14]. The LES computational grid is regular and is of size 118x106x16.

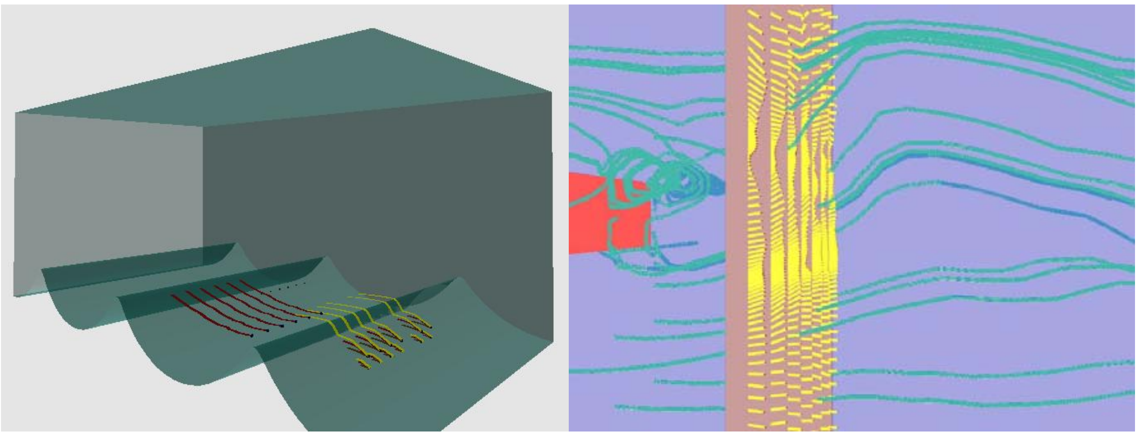

Figure 5. Turbulent flow developing on seabed starting at different times from the same position (left). LES flow past a rectangular cylinder. A cutting plane shows velocity distribution at a cross section (right). The cutting plane can be positioned in the flow field interactively by the user.

3.2 The virtual environment for the case studies

Typical visual components for the VEs for the above two studies consist of

- a backdrop of the boundary surfaces of the flow domain
- visualization widgets that implement derived flow quantities such as path ribbons, particle tracers, isosurfaces and vorticity tubes. These widget classes compute derived quantities, generate appropriate visual representation and pass them to the underlying VR system for rendering and object management

Typical interaction components for the VEs include, for example,

- interaction techniques to specify region/volume of interest (two-handed input to specify rubber banded volume)
- scales to convey parameter values, digital readers to display queried values, head-up displays for displaying meta data.

3.3 Real-time support and usability issues

The visualization steering needs to control the data management, data selection, data filtering, sampling and the mapping of data and derived quantities onto visual objects to real-time interaction rates. Of equal importance is an in-depth quantifiable understanding of usability issues associated with the application of VR interaction techniques to scientific visualization and indeed even the veracity of such applications. One of our goals behind the development of this framework is that it might aid rapid prototyping of virtual environments for scientific visualization, releasing much of the development effort to focus on these issues. These issues are to be quantified with user evaluation studies, which will inform future refinement of the framework.

4 SUMMARY

Creating interaction rich virtual environments for novel applications such as VR-based scientific visualization remains challenging, partly due to the non-availability of modular add-on interaction components. It remains to be seen whether the application of VR-based interaction techniques to scientific visualization results in more productive exploration and hence better understanding of such data. Rapid prototyping tools and components such as those described here may aid the ability to experiment, analyse and quantify such issues. This paper demonstrates the visualization and interaction components via two case studies.

Acknowledgements

The work reported in this paper was partly funded by the UK EPSRC through grant GR/L52406 to the INQUISITIVE (INcreasing the Quality of User Interaction for Strategic Interactive Tasks in Virtual Environments) research project. The authors express their thanks to P.R. Voke, University of Surrey, UK and D.J. Nicholas, formerly of RAL, UK for the use of their data.

REFERENCES

[1] D.A. Bowman, and L.F. Hodges, “Formalizing the design, evaluation and application on interaction techniques for immersive virtual environments”, *J. Visual Languages and Computing*, **10**, 37-53 (1999).
[2] Steed, E. Frecon, D. Pemberton and G. Smith, “The London Travel Demonstrator”, Proceedings of the ACM Symposium on Virtual Reality Software and Technology, 20-22 December, (1999)
[3] A.S. Carlin, H.G. Hoffman and S. Weghorst, “Virtual reality and tactile augmentation in the treatment of spider phobia: A case study. Behaviour Research and Therapy”, **35**, 153-158 (1997).
[4] V.V.S.S. Sastry, A.T. Edwards and J. Steel, “A Virtual Environment for Naval Flight Deck Operations Training”, NATO Workshop "What is essential for Virtual Reality to meet military performance goals?", Hague, The Netherlands, 13-15 Apr (2000)
[5] L. Sastry, and D.R.S. Boyd, ”EISCAT virtual reality training simulation: A study in usability and effectiveness”, *Proc. Interactions in Virtual Worlds*, University of Twente, The Netherlands, 19-21 May (1999).
[6] G. Wesche, “The responsive workbench for visualization of fluid dynamics”, *HPCN9*8, Netherlands, 21 – 23 April (1998).
[7] S. Bryson, D. Kenwright, M. Cox, D. Ellsworth, R. Haimes, “Visually exploring gigabyte data sets in real time”, *Communications of the ACM*, **42(8)**, 82-90 (1999).
[8] H. Hasse, “Symbiosis of virtual reality and scientific visualization system”, *Computer Graphics Forum*, **15(3)**, 443-486 (1996).
[9] L. Sastry, D.R.S. Boyd, R.F. Fowler, V.V.S.S. Sastry, “Numerical flow visualization using virtual reality techniques”, *Proc. 8th International Symposium on Flow Visualization*, I. Grant (Eds), Sorrento, Italy (1998).

[10] L. Sastry, D.R.S. Boyd, and M.D. Wilson, "Design review and visualization steering using the INQUISITIVE interaction toolkit", IPT-EGVE2001, 16-18 May (2001).
[11] R. Hubbold, J. Cook, M. Keates, S. Gibson, T. Howard, A. Murta, A. West, and S. Pettifer, "GNU/MAVERIK: A micro-kernel for large-scale virtual environments", *Proc. VRST'99*, ACM Symposium on Virtual Reality Software and Technology (1999).
[12] M. Mine, "Virtual environment interaction techniques", UNC Chapel Hill Computer Science Technical Report TR95-018, (1995).
[13] S.G. Sajjadi, J.N. Aldridge and D.J. Nicholas, "Dynamics of sediment transport over rough sand ripples", *Proc. Flow Modelling and Turbulence Measurements VI*, Balkema, Rotterdam, 821-830 (1996).
[14] P.R. Voke, "Flow past a square cylinder: Test case les2", *Direct and Large-Eddy Simulation 2*, J. Chollet, P.R. Voke and L. Kleiser (Eds). Kluwer Academic Press (1997).