Title

Towards Interactive Finite Element Analysis of Shell Structures in Virtual Reality

Authors

Liverani, Alfredo
Kuester, Falko
Hamann, Bernd

Publication Date

1999

Towards Interactive Finite Element Analysis of Shell Structures in Virtual Reality

A. Liverani*, F. Kuester**, B. Hamann**

* DIEM – University of Bologna, Italy
a.liverani@mail.ingfo.unibo.it
** Center for Image Processing and Integrated Computing (CIPIC)
Department of Computer Science
University of California - Davis, CA 95616, USA
(kuester,hamann)@cs.ucdavis.edu

Abstract

A first step towards a semi-immersive Virtual Reality (VR) interface for Finite Element Analysis (FEA) is presented in this paper. During recent years, user interfaces of FEA solvers have matured from character-based command-line driven implementations into easy-to-use graphical user interfaces (GUIs). This new generation of GUIs provides access to intuitive and productive tools for the management and analysis of structural problems. Many pre- and post-processors have been implemented targeting the simplification of the man-machine interface in order to increase the ease of use and provide better visual analysis of FEA solver results. Nevertheless, none of these packages provides a real 3D-enabled interface. The main objective of this project is to join state-of-the-art visualization technology, VR devices, and FEA solvers into the integrated development environment VRFEA.

Keywords

Virtual Reality, Interactive Modeling, Finite Element Analysis, 3D Modeling, Simulation.

1. Introduction

Primarily driven by product quality, cost, and time-to-market considerations, most automotive and aerospace companies heavily invested into the development and implementation of new VR technology during the last five years. These newly created synthetic environments have matured into valuable tools in the areas of human factors and usability studies, manufacturing, and simulation-based design. Nevertheless, most of these applications use immersive or augmented VR technology primarily to visualize or interact with pre-defined data neglecting some of its powerful features in the area of content creation. As companies focus on streamlining productivity in the pursuit of global competitiveness, the migration to computer-aided design, computer-aided manufacturing and computer-aided engineering systems has established a new backbone of modern industrial product development. While most of these technological advances are of high benefit to the engineering and design community they still lack some of the important visual and haptic features crucial to efficient human computer interfaces that can be addressed using the available VR hardware.

1.1 Motivation

In classic FEA environments engineers usually spend 40% of their working time on 3D shape modeling, 30% on mesh generation (the definition of boundary conditions and simulation parameters), 20% on result analysis and review and only 10% on running the FEA solvers.

The observation that modeling, mesh generation and adjustment, specification of simulation parameters and result analysis tasks are responsible for 90% of the workload indicates that further technological advances are required. A significant number of the areas prone to improvement require the visualization of scientific data, and intuitive 3D-display technology could be successfully involved in a solution approach.

Furthermore, VR devices can aid the designer in content creation and design verification during the pre-processing and the evaluation and interpretation of results during the post-processing phase. The primary focus of this paper is on those tasks that can be performed in real-time. This paper is intended as a first step towards the development of a fully integrated modeling and simulation environment, combining VR and FEA.

The design goal was to provide a visualization front-end, which easily interfaces with either, the traditional command line driven FEA applications or newer plug-in based technology. Furthermore, the visualization component had to enable smooth transitions between standard on-screen and VR-enabled modeling tasks. Therefore, support for stereo output devices, additional VR specific input hardware such as spatial trackers and data gloves had to be provided besides the standard workstation specific display modes.

1.2 Related work

Recent experiments performed at Iowa University [1] have used VR approaches in FEA to evaluate "parameter sensivity" and facilitate design. This means that VR is not really integrated into the FEA modeling and analysis procedure, but it is only used as a visualization tool to seek optimal design parameters.

Our environment, VRFEA, was designed to overcome the integration gaps between VR and FEA. In this environment the designer uses immersive VR tools to manage a FEA model, manipulate nodes or shells, create or adjust the mesh, specify boundary conditions, and visualize the results.

2. Setup

2.1 Hardware

VRFEA was designed for a new generation of stereo projection systems currently marketed under names like *ImmersiveWorkbench*, *Responsive Workbench* and *ImmersaDesk* [21,22]. We used the *ImmersiveWorkbench* from Fakespace which allows stereo projection of 3D computer-generated images onto an approximately 2m*1.5m wide projection area. A four-processor SGI Onyx2 InfiniteReality (225MHz, R10000 processor) system was used as the rendering and computation engine. The basic hardware setup is illustrated in Figure 1. The user is wearing shutter glasses with integrated head tracking for stereoscopic viewing and uses a set of pinch gloves combined with a stylus device for interaction with VRFEA. The spatial data-set describing the user's head position and hand movements is fully incorporated into the environment. We briefly describe the input devices:

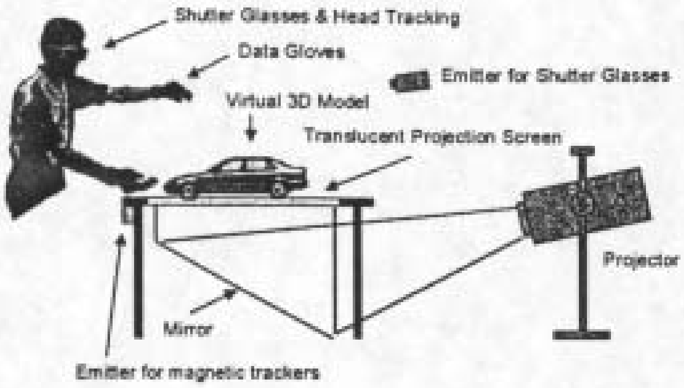

Figure 1: Hardware setup.

- **Stylus**. Using a fixed transmitter as reference, this pencil-like system accurately computes position (x, y and z coordinates) and orientation (yaw, pitch and roll) of a tiny receiver contained in the stylus. In addition, it provides an integrated button that can be used for picking actions.
- **Gloves.** The pinch system uses cloth gloves with electrical sensors in each fingertip. Contact between any two or more digits completes a conductive path, providing a variety of possible "pinch" gestures that can be associated with distinct actions. Additionally, an attached electromagnetic tracker captures the position of each glove.

2.2 Software

One of the governing development goals was to provide a platform-independent visualization front-end for FEA applications, which supports either plug-in technology or the traditional command line interface. In addition, a smooth transition between conventional and VR display modes had to be possible. To facilitate these requirements OpenInventor was selected as the implementation language of choice for the visualization component.

2.2.1 Open Inventor. Open Inventor is an object-oriented developer's toolkit that simplifies 3D graphics programming and developing high-performance interactive 3D programs. The rich Inventor set of pre-programmed building blocks defines a full-featured, extensible framework upon which entire applications can be developed.

It includes a wide variety of geometry, property, and group objects as well as manipulators for user interaction, and high-level viewers and editor components.

The underlying object hierarchy produces for each shape model change and motion an on-screen visualization. Additionally, OpenInventor establishes a file format standard for 3D data exchange that is the basis for the Virtual Reality Modeling Language (VRML).

2.2.2 Device Drivers under Inventor. A new set of device plug-ins was developed to provide OpenInventor with the real-time support for VR input devices, including the Fakespace pinch glove and the Polhemus Fastrack system. Furthermore, a new object selection method was introduced to bypass standard libraries for mouse-based object selection.

This modification was essential to enable real-time transition between keyboard-mouse and semi-immersive modeling metaphors. In fact, a generic event-based collision detection routine continuously checks for object collisions. Verified collisions trigger intuitive visual and audio feedback. These additions to the OpenInventor toolkit have substantially increased its VR potential.

3. Finite Element Code

3.1 Solver

The FE solver was implemented as a console-type application using ANSI C for portability reasons. Commands are passed to the solver through a command line interface (see Figure 2) and a generic file description language.

```
C:\IMP\FEA>fea.exe
NOME FILE INPUT :
   [illegible]
TIPO USCITA GRAFICA :
  0 = IGES
  1 = [illegible]
   [illegible]

 Lettura Dati.

MEMORIA ALLOCATA = 136.5? K

 Soluzione sistema lineare F = K*u

MEMORIA ALLOCATA = 2544.56 K

------ C A L C O L O   E S E G U I T O ------

C:\IMP\FEA>
```

Figure 2: FE solver command line interface.

The file format supports geometry information in mesh format, node coordinates, a variety of element types, boundary conditions and modeling parameters such as forces or displacements applied to particular nodes.

An input file might look like this:

```
# NUM. MATERIALS
  1
# IND. MAT. FILE NAME
  0  MAT08.MAT

# NUM. NODI
  4
# IND.   X      Y      Z
0  0.00   0.00    0.00
1  1.00   0.00    0.00
2  1.00   1000.00   0.00
3  0.00   1000.00   0.00

# BEAM  SHELL  BRICK
  0   1   0
# BEAM ELEMENTS:

# SHELL ELEMENTS:
# ind  id_laminato
#      n0  n1  n2  n3
  0  0
      0   1   2   3

# NUM. CONSTRAINTS
   2
0  0  G
   6  1  2  3  4  5  6
1  3  G
   6  1  2  3  4  5  6

# NUM. LOADED NODES
   1
0  2
   0.000  0.000  0.000
   0.000  0.000  50.000

# NUM. GIVEN DISPLACEMENTS
 0
# GIVEN DISPLACEMENTS.
# id_nodo x  y  z
#       rx ry rz
```

The FE model can be described with either solid 3D elements as bricks, with shell elements for thin parts or a combination thereof. A classic frontal method was implemented to solve linear and non linear equation systems. This solver demonstrated good performance, solving small meshes (around 200 nodes) in less than two seconds and large models (around 3000 nodes) in about one minute. The command line-based interface style was chosen to simulate and test the visual front-end for its suitability with off-the-shelf products.

3.2 Input and Output Files

VRFEA provides two core modules for pre-processing and post-processing that encapsulate the FE solver. Like most commercially available pre-processor FEA solvers, VRFEA writes an input file (described above) and passes it to the solver. The files created by the solver include node displacements, written in ASCII format. This is an example:

```
      DISPLACEMENTS
node   Ux       Vy       Wz        Thx      Thy      Thz

 0 +0.000000E+00 +0.000000E+00 +0.000000E+00
      +0.000000E+00 +0.000000E+00 +0.000000E+00
 1 +0.000000E+00 +0.000000E+00 +0.000000E+00
      +0.000000E+00 +0.000000E+00 +0.000000E+00
 2 +0.000000E+00 +0.000000E+00 +0.000000E+00
      +0.000000E+00 +0.000000E+00 +0.000000E+00
 3 +0.000000E+00 +0.000000E+00 +0.000000E+00
      +0.000000E+00 +0.000000E+00 +0.000000E+00

    Von Mises stresses on elements:

    sigma1          sigma2          tau12

0   -8.129862e-01  1.793307e-01  1.418309e+00
1   4.189166e+00  9.180581e-02  6.915451e+00
2   7.480460e+00  3.414629e-02  -1.414779e+00
3   2.482450e+00  1.211878e-01  -6.898216e+00
4   1.519136e-17  4.326919e-17   3.223468e-17
5   -2.482450e+00  -1.211878e-01  6.898216e+00
6   -7.480460e+00 -3.414629e-02  1.414779e+00
7   -4.189166e+00  -9.180581e-02  -6.915451e+00
8   8.129862e-01  -1.793307e-01  -1.418309e+00
    .....
```

After the solver terminates, the results are passed to the post-processing stage of VRFEA. As mentioned earlier, information is passed through a file-based interface for compatibility reasons. Considering the real-time requirements for certain components, the use of multi-programming and shared memory has proven advantageous.

4. Static Simulations

The proof-of-concept implementation of VRFEA was tested by interactively applying specified node forces and node displacements.

The meshes for the given examples (Figures 3-5) were created with build-in VRFEA functionality.

Mesh generation was strongly simplified due to the chosen regular geometric design, which resulted in a mesh consisting of 150 nodes (magnified by colored spheres) and 126 shell elements (connecting the nodes). Constrained nodes are represented using red cubes, as shown in Figure 3.

4.1 "Forceless" Force Input

Following this new paradigm for integration among VR and FEA, the designer might be interested in testing and simulating the finger touch on a stereo deck or the strength of a drawer in a car console.

During the first implementation cycle, force feedback devices were not available and we had to define new metaphors for "*forceless-force-input*". This is the list of supported interaction modes:

- interactive node force displacement
- interactive node displacement
- force as implied by displacement
- force as a result of compressing a virtual medium
- force as implied by magnitude (distance between two points)
- force as implied by a visual queue (arrow)

Ultimately, the goal is to replace these concepts with actual force-feedback devices enabling the user to physically perceive model contents.

4.2 Interactive Node Force Application

The application of forces to mesh nodes may be considered a classical example of FEA. Standard 2D interfaces give a designer the option to select a node on the screen. VRFEA introduces a new interaction paradigm, a symbolic arrow, representing the force being applied.

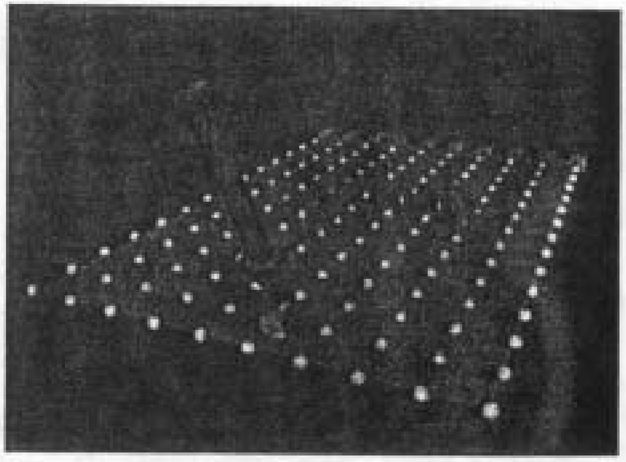

Figure 3: Increasing force value

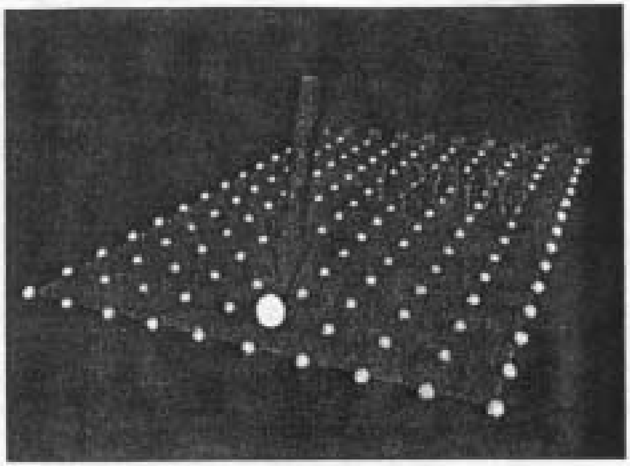

Figure 4: Decreasing force value.

The magnitude of these force vectors can be easily adjusted through associated gestures or actions provided by the stylus and glove interaction devices. If the stylus proxy is in the upside position, the force increases (Figure 2), otherwise it decreases (Figure 3).

However, this enhancement does not exclude the presence and use of multi-purpose menus, but it can dramatically reduce the frequency with which the menus have to be accessed and thus intuitive modeling efficiency is increased.

Object selection is based on a collision detection algorithm that continuously monitors the input device proxy in relation to the FE model. At any time the FE solver can be accessed to interactively update the model parameters. The surface is colored based on strain or displacement information obtained from the solver. For our test cases the solver latency was minimized to be lower than a three-second threshold.

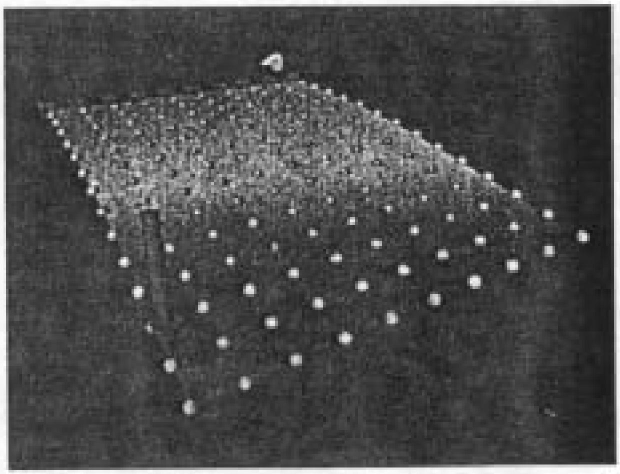

Figure 5: Color-coded results.

4.3 Interactive Node Displacement Application

To enable complete simulation in a VR environment, a very interesting FEA example could push VR and interface technology. In many test environments for ergonomic and usability studies, the designer needs to investigate the capabilities of the structure strength under displacements.

This test could easily be performed by simulating node displacement and computing the resulting surface deformation. To underlay the node displacements, a symmetric constraint set can be chosen (Figure 5).

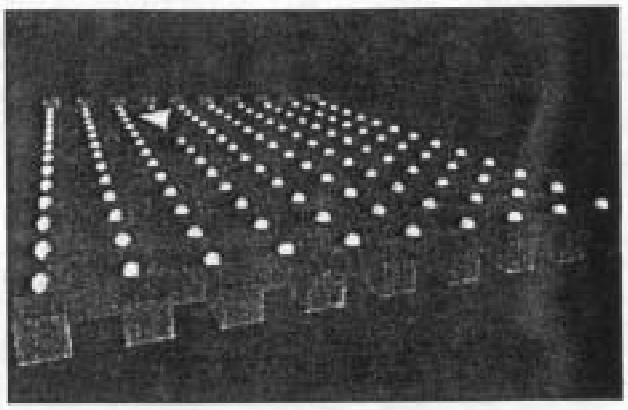

Figure 6: Symmetric constraint set.

As in the previous example, nodes or shell elements can be selected. However, in this setting forces are applied implicitly through the displacements applied to the elements. Figure 6 shows such a displacement test and the results obtained from the solver (Figure 7).

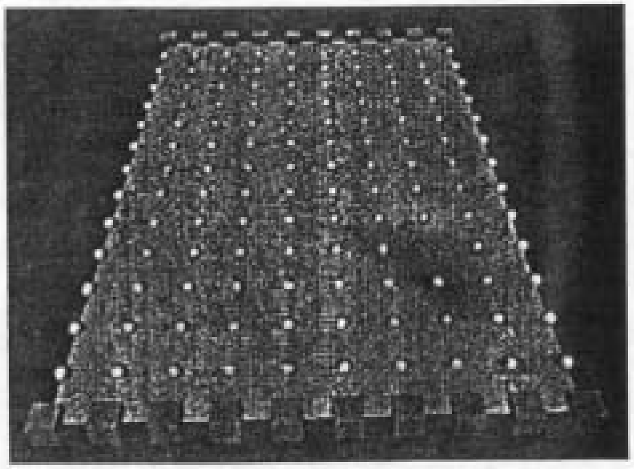

Figure 7: Applying node displacement.

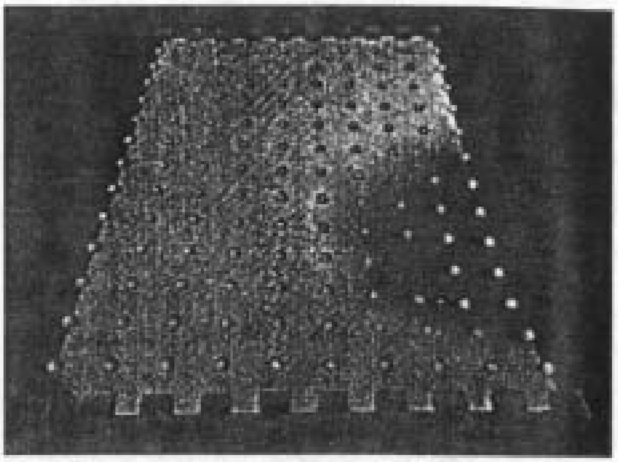

Figure 8: Color-coded results.

5. Conclusions

A new interaction method for FEA was introduced in this paper. While modern pre-processor and post-processor software for FEM/FEA primarily utilizes 2D interfaces, this interface prototype represents an essential step towards proving the benefits of VR to FE applications, and goes beyond visualization and navigation of results.

The use of 3D input devices and stereo rendering systems has the potential to transform how engineers interact with design environments enhancing model manipulation, examination and modification efficiency and product quality.

6. Future Work

So far, the current research focus has been on the integration of fully interactive 3D meshing techniques and the associated interactive meshing interface. With increasing system performance and the growing availability of a new generation of force-feedback devices we plan to develop an improved version of VRFEA in the near future.

7. Acknowledgements

This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), the Office of Naval Research under contract N00014-97-1-0222, the Army Research Office under contract ARO 36598-MA-RIP, the NASA Ames Research Center through an NRA award under contract NAG2-1216, the Lawrence Livermore National Laboratory through an ASCI ASAP Level-2 under contract W-7405-ENG-48 (and B335358, B347878), and the North Atlantic Treaty Organization (NATO) under contract CRG.971628 awarded to the University of California, Davis. We also acknowledge the support of Silicon Graphics, Inc., and thank the members of the Visualization Thrust at the Center for Image Processing and Integrated Computing (CIPIC) at the University of California, Davis.

8. References

[1] Tsung-Pin Yeh, Vance, Judy M.: "Combining MSC/Nastran, sensivity methods, and Virtual Reality to facilitate Interactive Design", MSC/Nastran Web site: http://www.macsch.com/tech/paper1/paper1.html

[2] Sastry, L, J. V. Ashby, D. R. S. Boyd, R. F. Fowler, C. Greenough, J. Jones, E. A. Turner-Smith and N. P. Weatherill, "Virtual Reality Techniques for Interactive Grid Repair", Numerical Grid Generation in Computational Field Simulations, Ed.

[3] J.F. Thompson, B. Soni, N. Weatherill, "Handbook of Grid Generation". CRC Press, 1999 (ISBN 0-8493-2687-7).

[4] Foley, J. D., Van Dam, A., Feiner, S. K. and Hughes, J. F., 1992, "Computer Graphics, Principles and Practice", Addison-Wesley Publishing second edition.

[5] Newman, W. M. and Sproull, R. F., "Principles of Interactive Computer Graphics", McGraw-Hill, 1979.

[6] Kalawsky, R.S., "The Science of Virtual Reality and Virtual Environments", Addison-Wesley, 1994, ISBN 0-201-63171-7.

[7] "HIT Lab Overview and Projects" Human Interface Technology Laboratory Seattle/WA/USA, Technical Report No. HITL-P-91-1, 1991.

[8] Hollerbach, J.M., Cohen, E.C., Thompson, W.B., and Jacobsen, S.C., "Rapid Virtual Prototyping of Mechanical Assemblies", Proc. 1996 NSF Design and Manufacturing Grantees Conf., (Albuquerque, NM), Jan. 2-5, 1996, pp. 477-478.

[9] Ballard, D., Brown, C.: "Computer Vision", Prentice Hall, 1982.

[10] Persiani, F., Liverani A.: "Virtual Reality CAD Interface", ADM Proceedings, Florence 17-19 Sept. 1997.

[11] Liang, J., Green, M.: "JDCAD: A Highly Interactive 3D Modeling System", Computers and Graphics, Vol. 18, No. 4, pp. 499 - 506, 1994.

[12] Butterworth, J., Davidson, A., Hench, S., Olano, T. M.: "3DM: A Three Dimensional Modeler Using a Head-Mounted Display", Proc.1992 Symposium on Interactive 3D Graphics, Cambridge, Massachusetts, pp. 135 - 138, March 29 - April 1 1992.

[14] Wang, Sidney W., Kaufman, Arie E., "Volume Sculpting" 1995, Symposium on Interactive 3D Graphics, Monterey CA USA, ACM Press, 1995.

[17] Williams, L. "3d paint" in: Computer Graphics, 24(2):225-233, March 1990.

[18] Murakami, T.: "Direct and intuitive input device for 3D shape design", DE-Vol. 83, 1995 Design Engineering Technical Conferences, Vol. 2, pp. 695-701, ASME, 1995.

[19] Sachs, E., Roberts, A., Stoops, D.: "3Draw: A Tool for Designing 3D Shapes", IEEE Computer Graphics and Applications, Vol.11, pp. 18-24, 1991.

[20] Sutherland, I. E.: "SKETCHPAD: a man-machine graphical communication system", MIT Lincoln Laboratory, Lexington, MA. Technical Report, TR-296, 1965.

[21] Krueger, W., Froehlich, B. "Visualization Blackboard: The Responsive Workbench", IEEE Computer Graphics and Applications, 14(3):12-15, May 1994.

[22] Rosenblum, L., Durbin, J., Doyle, Tate, D., "Projects in VR: Situational Awareness Using the Responsive Workbench. IEEE Computer Graphics and Applications 17(4):12-13, July/August 1997.

4. STATIC SIMULATIONS

The proof-of-concept implementation of VRFEA was tested by interactively applying specified node forces and node displacements.

The meshes for the given examples (Figures 3-5) were created with build-in VRFEA functionality. Mesh generation was strongly simplified due to the chosen regular geometric design, which resulted in a mesh consisting of 150 nodes (magnified by colored spheres) and 126 shell elements (connecting the nodes). Constrained nodes are represented using red cubes, as shown in Figure 3.

4.1 "Forceless" Force Input

Following this new paradigm for integration among VR and FEA, the designer might be interested in testing and simulating the finger touch on a stereo deck or the strength of a drawer in a car console.

During the first implementation cycle, force feedback devices were not available and we had to define new metaphors for "*forceless-force-input*". This is the list of supported interaction modes:

- interactive node force displacement
- interactive node displacement
- force as implied by displacement
- force as a result of compressing a virtual medium
- force as implied by magnitude (distance between two points)
- force as implied by a visual queue (arrow)

Ultimately, the goal is to replace these concepts with actual force-feedback devices enabling the user to physically perceive model contents.

4.2 Interactive Node Force Application

The application of forces to mesh nodes may be considered a classical example of FEA. Standard 2D interfaces give a designer the option to select a node on the screen. VRFEA introduces a new interaction paradigm, a symbolic arrow, representing the force being applied.

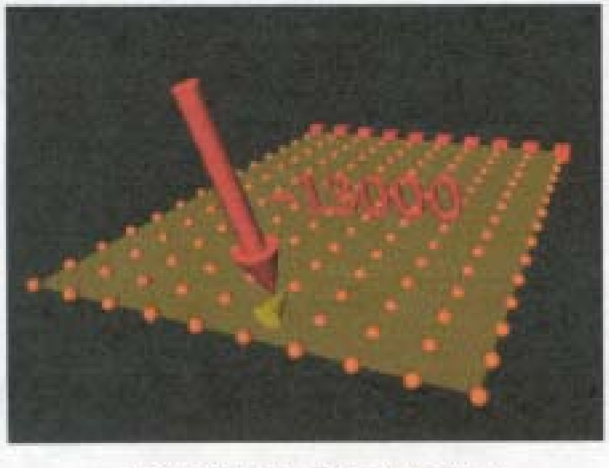

Figure 3: Increasing force value

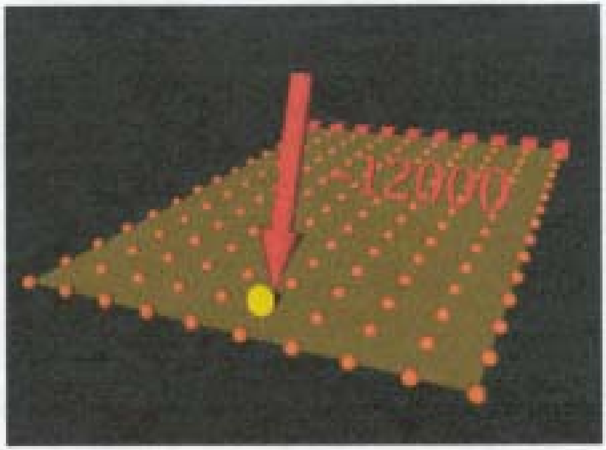

Figure 4: Decreasing force value.

The magnitude of these force vectors can be easily adjusted through associated gestures or actions provided by the stylus and glove interaction devices. If the stylus proxy is in the upside position, the force increases (Figure 2), otherwise it decreases (Figure 3).

However, this enhancement does not exclude the presence and use of multi-purpose menus, but it can dramatically reduce the frequency with which the menus have to be accessed and thus intuitive modeling efficiency is increased.

Object selection is based on a collision detection algorithm that continuously monitors the input device proxy in relation to the FE model. At any time the FE solver can be accessed to interactively update the model parameters. The surface is colored based on strain or displacement information obtained from the solver. For our test cases the solver latency was minimized to be lower than a three-second threshold.

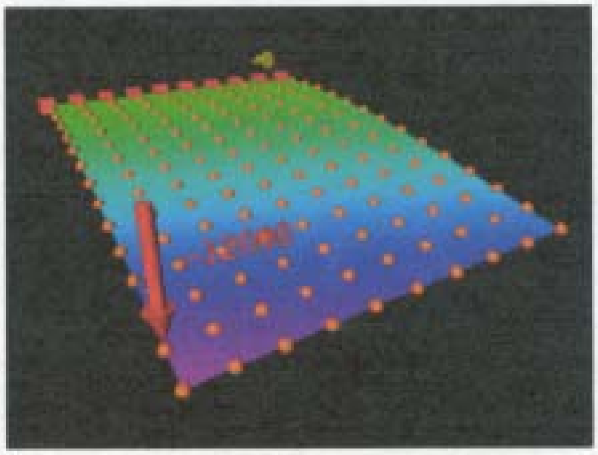

Figure 5: Color-coded results.

4.3 Interactive Node Displacement Application

To enable complete simulation in a VR environment, a very interesting FEA example could push VR and interface technology. In many test environments for ergonomic and usability studies, the designer needs to investigate the capabilities of the structure strength under displacements.

This test could easily be performed by simulating node displacement and computing the resulting surface deformation. To underlay the node displacements, a symmetric constraint set can be chosen (Figure 5).

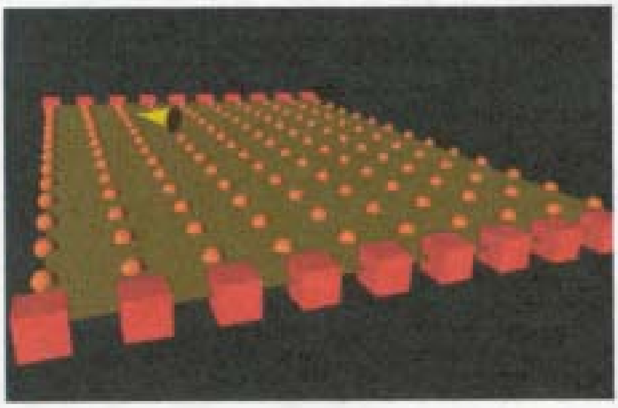

Figure 6: Symmetric constraint set.

As in the previous example, nodes or shell elements can be selected. However, in this setting forces are applied implicitly through the displacements applied to the elements. Figure 6 shows such a displacement test and the results obtained from the solver (Figure 7).

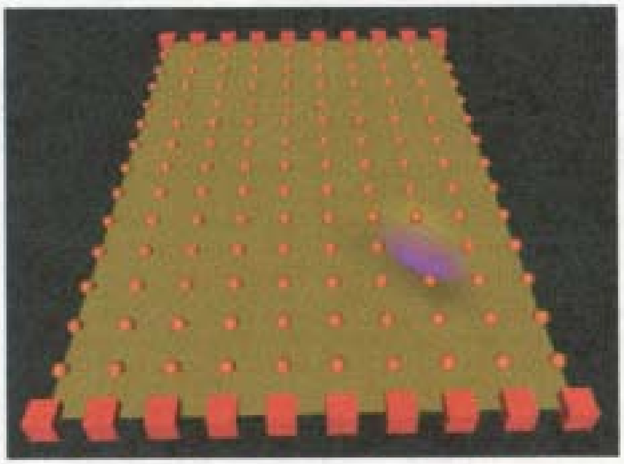

Figure 7: Applying node displacement.

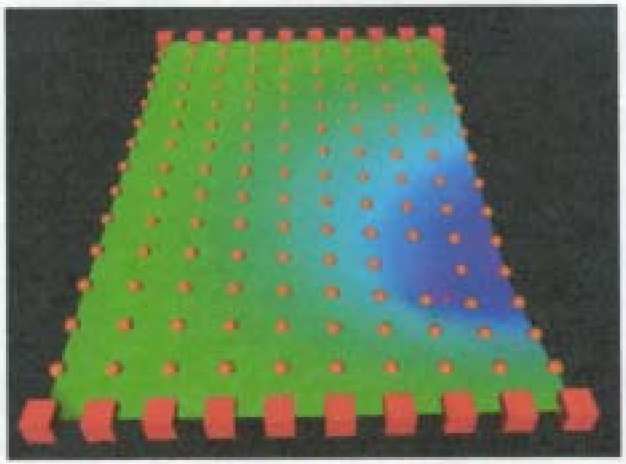

Figure 8: Color-coded results.

5. CONCLUSIONS

A new interaction method for FEA was introduced in this paper. While modern pre-processor and post-processor software for FEM/FEA primarily utilizes 2D interfaces, this interface prototype represents an essential step towards proving the benefits of VR to FE applications, and goes beyond visualization and navigation of results.

The use of 3D input devices and stereo rendering systems has the potential to transform how engineers interact with design environments enhancing model manipulation, examination and modification efficiency and product quality.

6. FUTURE WORK

So far, the current research focus has been on the integration of fully interactive 3D meshing techniques and the associated interactive meshing interface. With increasing system performance and the growing availability of a new generation of force-feedback devices we plan to develop an improved version of VRFEA in the near future.

7. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation under contract ACI 9624034 (CAREER Award), the Office of Naval Research under contract N00014-97-1-0222, the Army Research Office under contract ARO 36598-MA-RIP, the NASA Ames Research Center through an NRA award under contract NAG2-1216, the Lawrence Livermore National Laboratory through an ASCI ASAP Level-2 under contract W-7405-ENG-48 (and B335358, B347878), and the North Atlantic Treaty Organization (NATO) under contract CRG.971628 awarded to the University of California, Davis. We also acknowledge the support of Silicon Graphics, Inc., and thank the members of the Visualization Thrust at the