9-2005

Virtual Reality Interactive Design Utilizing Meshless Stress Re-Analysis

Kurt A. Chipperfield
Iowa State University

Judy M. Vance
Iowa State University, jmvance@iastate.edu

Virtual Reality Interactive Design Utilizing Meshless Stress Re-Analysis

Abstract

Interactive design gives engineers the ability to modify the shape of a part and immediately see the changes in the part's stress state. Virtual reality techniques are utilized to make the process more intuitive and collaborative. The results of a meshless stress analysis are superimposed on the original design. As the engineer modifies the design using subdivision volume free-form deformation, the stress state for the modified design is computed using a Taylor series approximation. When the designer requests a more accurate analysis, a stress re-analysis technique based on the pre-conditioned conjugate gradient method is used with parallel processing to quickly compute an accurate approximation of the stresses for the new design.

Keywords

VRAC, Stress, Design, Virtual reality

Disciplines

Computer-Aided Engineering and Design

Proceedings of IDETC/CIE 2005
ASME 2005 International Design Engineering Technical Conferences
& Computers and Information in Engineering Conference
September 24-28, 2005, Long Beach, California USA

DETC2005-84496

VIRTUAL REALITY INTERACTIVE DESIGN UTILIZING MESHLESS STRESS RE-ANALYSIS

Kurt Chipperfield
Iowa State University
Virtual Reality Applications Center
Mechanical Engineering Department
1620 Howe Hall
Ames, IA 50010

Judy Vance
ASME Fellow
Iowa State University
Virtual Reality Applications Center
Mechanical Engineering Department
1620 Howe Hall
Ames, IA 50010
jmvance@iastate.edu

KEYWORDS: virtual reality, re-analysis, free-form deformation, meshless method, parallel processing

ABSTRACT

Interactive design gives engineers the ability to modify the shape of a part and immediately see the changes in the part's stress state. Virtual reality techniques are utilized to make the process more intuitive and collaborative. The results of a meshless stress analysis are superimposed on the original design. As the engineer modifies the design using subdivision volume free-form deformation, the stress state for the modified design is computed using a Taylor series approximation. When the designer requests a more accurate analysis, a stress re-analysis technique based on the pre-conditioned conjugate gradient method is used with parallel processing to quickly compute an accurate approximation of the stresses for the new design.

INTRODUCTION

Finite element analysis has enabled designers to approximate product performance and predict product failure early in the design process before physical prototypes are built and tested. However, as products become more complex, longer computation times are needed to calculate these performance parameters. In addition, when a designer wishes to change the shape of a part in order to investigate multiple product design options, the entire finite element analysis must be recalculated. The intent of this research is to provide the designer with a design environment where part shapes can be modified and stresses can be examined interactively. We call this interactive design.

Virtual reality (VR) offers an ideal method for performing interactive design. The part is displayed in 3D and the designer can change the shape of the part using natural hand motions. Interactive stress analysis performed in a large display VR environment allows several people to simultaneously discuss and perform “what-if” scenarios on a proposed design. We believe the interactive design paradigm will result in better designs because the designer will develop an understanding of how changes in design affect the stress state and will use this knowledge to produce an optimum design. The addition of VR speeds up this process because natural hand motions make design modifications easy and position-tracking facilitates intuitive viewing of 3D geometry.

This paper presents the results of our research on interactive design in virtual reality. First, a short history of the use of VR for visualization of finite element results and VR used in interactive stress analysis is presented. This is followed by a detailed description of the design environment and the underlying theory and algorithms.

BACKGROUND

In 1994, Dai and Gobel first suggested using VR for visualization of finite element results [1]. Since then several researchers have followed this approach [2-5]. VR has also been used to visualize time-dependent results from car crash simulations performed by the finite element method [6]. In each of these cases, the virtual environment is used solely as a visualization environment. The user can only view the results of an analysis.

In addition to visualization of results, stress analysis techniques have been used in VR applications to compute both object deformations and forces as users interactively modify the displacements and/or forces applied to a virtual object. Force computation is required by VR programs that utilize haptics to provide force-feedback as objects are manipulated in the VR environment. Other VR applications use stress analysis techniques to compute the shape an object takes as users apply deformations. The finite element method has been used to calculate the shape of an object after deformation and the force required to perform the deformation [7-12]. Another stress analysis technique, boundary element analysis, has been used for similar applications [13, 14].

These researchers used various techniques to increase the speed of the deformation calculations: Bro-Nielsen used condensation of the finite element model to speed up the computations [9, 10]. Hui and Ma used finite elements with parallel computation to calculate the deformation of a virtual object as the user interacted with it[15]. Impelluso also parallelized the finite element process which allowed him to achieve interactive visualization of the deformation of a virtual object [16].

None of the methods described thus far allow the user to modify the initial object geometry – these methods only provide the ability to visualize analysis results and allow boundary conditions to be modified. In 1989 Vance and Bernard developed an interactive design method for designing systems subject to constraints on natural frequency and mode shapes [17]. The method allows users to interactively change the design while new vibration modes and frequencies are approximated quickly using a Taylor series approximation. Then in 1997, Yeh and Vance were the first to couple interactive part modification, virtual reality, and stress approximations [18-20]. Ryken and Vance followed with an industrial application using the VR interactive stress analysis program to redesign a lift arm from a John Deere tractor [21]. These applications allow the initial shape of the object to be changed while users are in the virtual environment.

The disadvantage of the method introduced by Yeh and Vance lies in the accuracy of the stress approximations and the requirement that the area to be modified must be chosen before the interactive design session. Their method utilizes a linear Taylor series approximation that computes the new stresses as the design is modified. This approximation is very fast but is only accurate for small design changes. In addition, in order to calculate the stress derivatives required for the linear Taylor series, the design area that is to be changed must be selected before the interactive VR design session starts. This inhibits truly interactive design. Our work seeks to overcome these limitations by utilizing an improved method of approximating the stresses as the shape of the part is changed.

VIRTUAL REALITY INTERACTIVE DESIGN ENVIRONMENT

A virtual reality application has been developed to demonstrate the interactive design method. This application will be referred to as M3D and is similar to the application written by Yeh and Vance [20].

In order to provide more accurate stress approximations for larger design changes, the meshless method coupled with pre-conditioned conjugate gradient (PCG) re-analysis was selected. The meshless method was chosen so that re-meshing could be avoided when large shape changes produced distortion in finite elements, which would alter the analysis. Recently, Chen, et al., has developed a fast meshless solver, which reduces the time to perform meshless analysis[22]. A re-analysis method uses the results from a stress analysis of the original design to quickly compute the stresses for a modified design. Several re-analysis methods were evaluated and the PCG method was found to be the most efficient [23].

In order to estimate the stresses fast enough as the shape is being changed, a two-step analysis procedure is used. While the user is changing the shape, a linear Taylor series stress approximation is used. These linear Taylor series approximations are the same ones used by Yeh and Vance and are only utilized while the user is interactively performing free-form deformation (FFD) on the part. When the user stops modifying the part shape, the meshless method and PCG re-analysis methods are used to quickly re-compute accurate stresses and sensitivities for the new design. The stress re-computations are performed faster through the use of parallel processing.

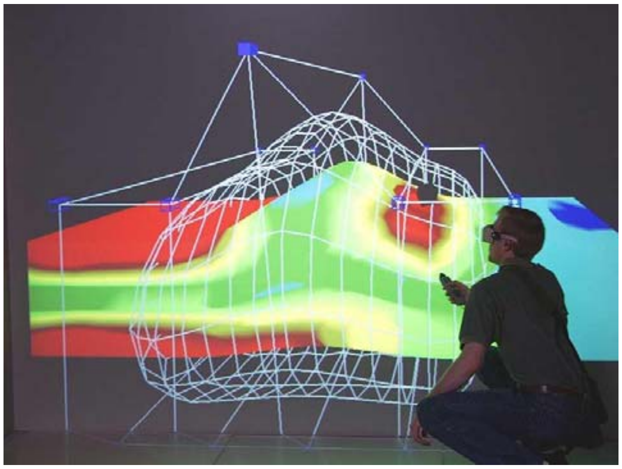

Figure 1: Interactive design using M3D in the C6

Virtual reality display is performed using VR Juggler, an Open Source VR software library developed at the Virtual Reality Applications Center at Iowa State University. Figure 1 shows the M3D application running in the C6 large screen projection VR system at the Virtual Reality Applications Center at Iowa State University. The C6 is a 10 foot square room consisting of three fixed walls, one movable wall, a ceiling and a floor, which all serve as rear-projection surfaces. Synchronized Barco projectors display stereo computer images on all six surfaces. User's head movements are tracked using Ascension Technology wireless magnetic trackers.

The initial geometry of the part is created using a finite element pre-processor or CAD package. An FEA pre-processor is also used to divide the geometry into a mesh of quadrilateral elements and to apply loads, to apply boundary conditions, and to assign material properties. Any common FEA program can be used, such as Abaqus, Ansys, or Nastran. The output from this pre-processor is then converted into an XML file format that has been defined for M3D. The XML file includes information about the geometry, mesh, material properties, loads, and boundary conditions.

Another XML file which describes the shape of the FFD volume and the location of the FFD control points is also created. The FFD volume is used to enclose the region of the part that the user wishes to modify and identify areas outside of the volume which won't be changed. FFD control points are located along the outside boundaries of the FFD volume. When the user moves a control point, the shape of the part within the FFD volume is changed. In this version of M3D, the control point that is used must be specified before execution of the program. In future versions of M3D, specification of control points and FFD volumes will be done within the interactive virtual environment. Figure 2 illustrates M3D being used to perform interactive design on an engine connecting rod. In this figure, the FFD volume is a rectangular volume surrounding the center section of the rod. The user can modify the design by moving any of the small cube-shaped handles along the edge of the FFD volume.

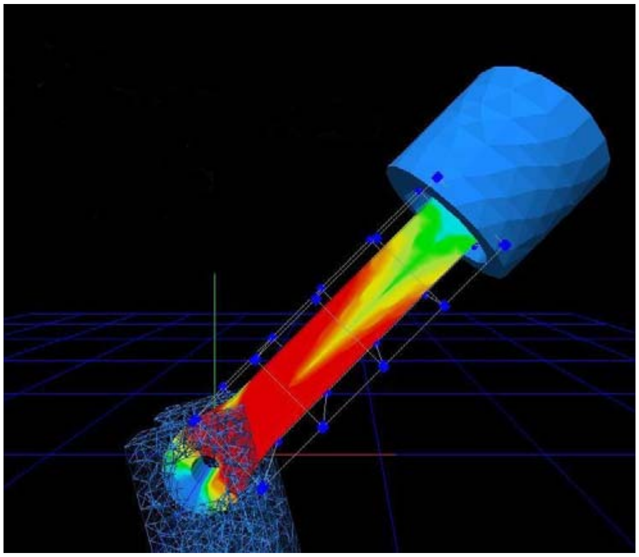

Figure 2: Interactive design of a connecting rod

An initial stress analysis on the part is performed in M3D and the sensitivities of the stresses in the part with respect to movement of the specified control point of the FFD volume are calculated. The stresses are displayed on the part in the VR environment as a color contour map. The user can then move the control point and view the change in the stresses based on a linear Taylor series approximation. After the designer is satisfied with the new design's stress state, he/she releases the control points and requests a more accurate analysis.

When the user requests a more accurate analysis, the meshless method and PCG re-analysis are used to re-compute the stresses in the part for the new shape. This recalculation is performed quickly but not fast enough to approximate the stresses during the interactive design modification, therefore, we use the linear Taylor series approximation for an initial approximation. When the results of the PCG re-analysis are displayed, they can differ from the Taylor series stresses. This difference is due to inaccuracy in the Taylor series and varies depending on the design being analyzed. If the designer notices large changes in stress state during interactive design, he or she must request a PCG re-analysis more often.

After the new, accurate stress state is displayed, new sensitivities of the stresses with respect to the control point are computed. This allows the user to again modify the part and see the change in stresses using a new linear Taylor series based at the latest design. We have chosen to display von Mises stresses as a color contour map on the part. This process continues until the user is satisfied with the stresses and the shape of the part.

DETAILS OF PROGRAM ORGANIZATION

The application is written in C++ so that an object oriented design could be utilized, making the code easier to debug and modify. A flowchart outlining the steps in M3D is given in Figure 3. Each of the boxes in the program represents an object, implemented in C++ as a class. The VR_Display class was implemented using VR Juggler, which is an Open Source library of VR functions written at the Virtual Reality Applications Center at Iowa State University [24]. An application written with VR Juggler can easily be executed using many different types of VR hardware (pinch glove, wand, Cyberglove, etc.) and in many different VR environments (CAVE, HMD, C6, etc.). VR Juggler allows VR programmers to focus on developing their application and not on optimizing the hardware and software configuration associated with VR. All drawing in M3D is performed using Open GL [25].The vcModel and XML_Reader classes were written by Tsung-Pin Yeh.

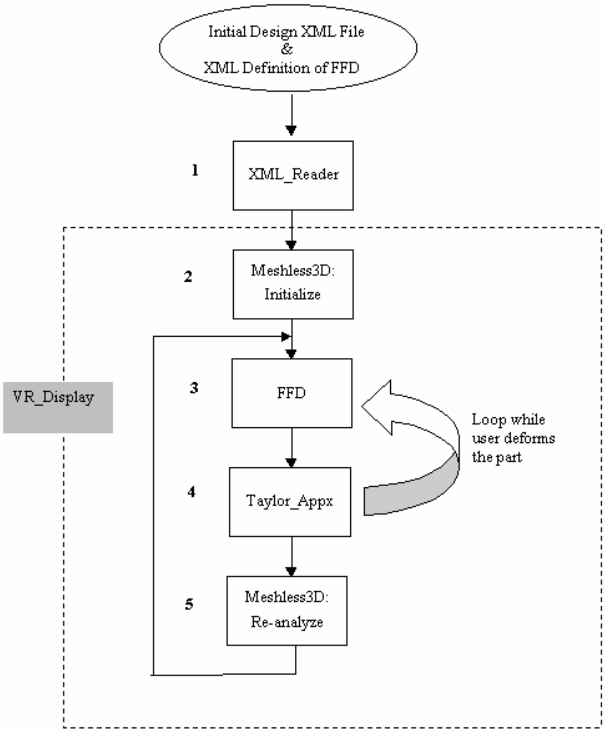

Figure 3: Program flowchart

The Meshless3D class is referenced in both steps two and five in Figure 3. Input to the Meshless3D class includes the geometry model created by the XML reader, the material properties, and the boundary conditions. The Meshless3D class then utilizes the meshless method to compute the stresses at each node of the geometry model. The initialization of the meshless class is shown in step 2 of Figure 3. The re-analysis step uses information stored after completion of the initialization step to quickly re-compute the stresses after the design is changed.

The initialization step for the 3D meshless class computes and stores information that allows fast computation of the re-analysis step. Initialization of the Meshless3D class includes the following operations.

1. Pre-processing. The input finite-element model is converted into meshless format and lists of influence are compiled for each node. The non-zero positions in the stiffness matrix are identified and a sparse data structure that stores only the non-zero positions is initialized.
2. Assemble the stiffness matrix (**K**).
3. Perform a factorization of the **K** matrix. The **K** matrix is first re-ordered so that fill in during factorization is minimized. The **K** matrix is then factored using a sparse direct method and the resulting factorization is stored.
4. Compute stress and strain at each node.

The re-analysis function uses the information computed in the initialization step to quickly re-compute the stresses after the design is changed. Re-analysis is performed in the following steps.

1. Assemble the K matrix. This step is identical to step 2 of initialization. The list of influence for each node is assumed to remain the same. The support region at each node is "deformed" along with the rest of the geometry as FFD is used to change the design.
2. Re-solve the linear system of equations. The result can be computed quickly with the pre-conditioned conjugate gradient re-analysis method, using the factored form of the initial K matrix as a pre-conditioner.
3. Compute stress and strain at each node. This step is identical to step 4 of initialization.

Free form deformation (FFD) techniques allow users to modify part geometry by moving control points. Even though the FFD bounding boxes illustrated in Figure 1 and Figure 2 are rectangular; the bounding boxes can be any shape, which gives the user more control over the shape changes. Arbitrarily shaped bounding boxes are made possible with subdivision volume FFD [26]. Control points located on the surface of the bounding volume are used to change the shape of the underlying part. In the current implementation of M3D, the shape and location of this bounding volume is defined by the user in an input file before running the program. The user must also specify which control point will be used to modify the design. Future implementations of M3D will allow the bounding volume and control points to be specified in the VR environment. The subdivision volume FFD class used in M3D was programmed by Tsung-Pin Yeh.

The Taylor_Appx class stores the sensitivities of the equivalent stresses with respect to x, y, and z movement of the FFD control point. The Taylor_Appx class also stores the stresses at the initial design. The initial stresses and the sensitivities are used to quickly compute new stress approximations when the user moves the FFD control point. When the user pauses, an accurate stress analysis is performed using the meshless method and PCG. At the same time, new sensitivities are computed so that a linear Taylor series approximation based at the new design can be used to compute fast stress approximations when the user begins to deform the part again.

The Taylor_Appx class uses the finite difference method to compute sensitivities. The control point is perturbed 5% in the x direction and the Meshless3D class computes the resulting stress state. This stress state and the movement of the control point are passed into the Taylor_Appx class, which computes and stores the sensitivity in the x direction. This process is repeated for the y and z directions.

PERFORMANCE MEASUREMENT FOR EXAMPLE PROBLEM

The performance of M3D was tested for three nodal discretizations of a 3D cantilever beam. The beam was 10 meters long, 2 meters tall, and 2 meters wide. A shear force was applied to one end of the beam and the other end was fixed. M3D was tested for a beam containing 40, 625, and 2560 nodes. Uniform nodal distributions were used in each case. Performance was measured for the computation of the **K** matrix and the solution of

$$\mathbf{K}d=f \qquad [1]$$

Where **K** is the stiffness matrix, *d* is the displacement vector and *f* is the force vector. Computing the **K** matrix is the most time consuming step performed in M3D. An SGI Onyx 2 computer with twenty-four 400 MHZ R12000 Processors and 12288 Mbytes of main memory was used for the performance measurements.

Because the computation of the **K** matrix is the most time consuming section of M3D, a parallel implementation was written using MPI. MPI is a library of functions used to write parallel programs [27]. The parallel implementation to compute the **K** matrix divides the nodes up between the available processors. Each processor computes the contribution to **K** at each of its nodes. One processor then reassembles the **K** matrix so that the stresses can be computed. Only the non-zero elements in the sparse **K** matrix are stored and transferred back to the main processor. Table 1 gives the times in seconds that were required to compute the **K** matrix for the example problems. These values include the time required to pass the necessary data to the processors, compute a portion of the **K** matrix, and return the **K** matrix to the main processor. This data illustrates that above a certain number of processors, adding more processors doesn't decrease the running time. Eventually performance becomes worse as more processors are added. For the 625 node example, there is no reason to use more than six to eight processors. This type of performance is typical for parallel computing applications.

Table 1: Time to recalculate the **K** matrix for example problem (seconds)

	nodes		
# proc	40	625	2560
1	0.142	10.08	76.65
3	0.06	3.82	28.28
4	0.046	3.03	21.22
6	0.032	2.01	16.02
8	0.026	1.59	12.58
10	0.03	1.36	11.43
15	0.05	1.14	9.25
20		1.08	7.1

DIRECT SOLUTION VS. PCG RE-ANALYSIS

This section compares the time required to solve the linear system of equations using a direct sparse solver to the time required by the PCG re-analysis method. Table 2 and Figure 6 give the time in seconds to execute the direct solver and the PCG re-analysis. The PCG re-analysis is executed for five iterations in each case. Five iterations were found to produce very accurate results for example problems, even for changes up to 100% [23]. There is a noticeable difference between the direct solver time and the PCG time. Table 2 gives the asymptotic running time for each method in 3D. The time for the direct method appears to be increasing at a more rapid rate than the PCG method. This is expected from the asymptotic running times.

Table 2: Time to solve example problems with direct sparse solver and PCG (seconds)

Nodes	direct	PCG
40	0.018 s	0.002 s
625	1.22 s	0.35 s
2560	21.82 s	4.28 s
asymptotic	O($N^2 + N^{1.33}$)	O($5N^{1.33} + 5N$)

CONCLUSIONS

A method has been developed that allows a designer to modify the shape of a part and immediately see how the shape change affects the stress state. M3D couples FFD, VR, linear Taylor series approximation, meshless stress analysis, PCG re-analysis, and parallel processing to provide a useful engineering design tool. VR is used to provide natural interaction and viewing. FFD and the linear Taylor series are used together to allow the designer to understand how shape changes affect the stresses in the part. Parallel computation of the meshless method and PCG re-analysis provide accurate stress information while the designer is in the VR environment.

The following improvements could be made to M3D. First, more features could be added to the VR design environment. Users should have the ability to specify the FFD bounding volume and control points while in the VR environment. They could also be given the ability to change the boundary conditions and loads. Re-computation of the **K** matrix could be accelerated by only computing the section of **K** that changes. For example, if the FFD volume only includes a small portion of the part geometry, a large part of the **K** matrix will remain the same. Finally, other portions of the code, such as the PCG algorithm, could be computed in parallel to improve performance.

ACKNOWLEDGEMENTS

The authors would like to thank JS Chen and Sangpil Yoon for their help with the meshless implementation and Tsung-Pin Yeh for his help with the FFD method. This work was funded by NSF grant number DMI-0084876 and was completed at the Iowa State University Virtual Reality Applications Center.

REFERENCES

1. Dai, F. and Goebel, M., 1994, "Virtual prototyping - An approach using VR-techniques," *ASME Computers in Engineering Conference Proceedings*, Minneapolis, MN, pp. 311-316.
2. Plaskacz, E. J., Kulak, R. F., and et.al., 1996, "Use of high-performance computers, FEA and the CAVE automatic virtual environment for collaborative design of complex systems," *ASME Design Engineering Show and Conference*, Chicago, IL.
3. Bradshaw, S., et al., 1995, "An interactive virtual environment for finite element analysis," *High Performance Computing Conference Proceedings*, Tucson, AZ.
4. Kuhn, M. A., Plaskacz, E. J., and et.al., 1996, "Finite element visualization in the cave virtual reality environment," *ASME Design Engineering Show and Conference*, Chicago, IL.
5. Shultz, M., Reuding, T., and Ertl, T., 1998, "Analyzing engineering simulations in a virtual environment," *IEEE Computer Graphics and Applications*, 18, pp. 46-52.
6. Schulz, M., Ertl, T., and Reuding, T., 1998, "Crashing in Cyberspace - Evaluating Structural Behaviour of Car Bodies in a Virtual Environment," *Virtual Reality Annual International Symposium*, Atlanta, GA, pp. 160-166.
7. Astley, O. R. and Hayward, V., 1998, "Multirate Haptic Simulation Achieved by Coupling Finite Element Meshes Through Norton Equivalents," *International Conference on Robotics and Automation*, pp. 989-994 & 1050-4729.
8. Berkley, J., et al., 1999, "Real-Time Finite Element Modeling with Haptic Support," *ASME Design Engineering Technical Conferences*.
9. Bro-Nielsen, M., 1998, "Finite element modeling in surgery simulation," *Proceedings of the IEEE*, 86, pp. 490-503.
10. Bro-Nielsen, M. and Cotin, S., 1996, "Real-time volumetric deformable models for surgery simulation using finite elements and condensation," *Computer Graphics* Forum, 15, pp. 57-66.

11. Hutter, R. C. and Niederer, P. F., 1999, "Fast, accurate and robust finite element algorithm for the calculation of organ deformations," *American Society of Mechanical Engineers,* Bioengineering Division.
12. Rhomberg, A., et al., 1998, "Design of a FEW computation engine for real-time laparoscopic surgery simulation," *International Parallel Processing Symposium*.
13. James, D. L. and Pai, D. K., 1999, "Accurate real time deformable objects," *ACM SIGGRAPH Conference on Computer Graphics*.
14. Xu, M. and Tang, Z., 1996, "Boundary element method for simulation of deformable objects," *Journal of Computer Science and Technology*, 11, pp. 497-506.
15. Hui, K. C. and Ma, M. C., 1998, "Deforming Virtual Objects with an Instrumented Glove," *1998 Computer Graphics International*, Hannover, Germany, pp. 393-395.
16. Impelluso, T. J., 1998, "Physically-based Virtual Reality: Integrating FEM and Visualization," *Engineering Mechanics: A Force for the 21st Century, proceedings of the 12th Engineering Mechanics Conference*, La Jolla, CA, pp. 178-181.
17. Vance, J. M. and Bernard, J. E., 1989, "Interactive analysis of vibration modes using approximation algorithms and computer graphics," *Computer-Aided Design*, 21, pp. 430-434.
18. Yeh, T.-P. and Vance, J. M., 1997, "Combining MSC/NASTRAN, sensitivity methods, and virtual reality to facilitate interactive design," *Finite Elements in Analysis and Design*, 26, pp. 161-169.
19. Yeh, T.-P. and Vance, J. M., 1998, "Interactive Design of Structural Systems in a Virtual Environment," *Computer Modeling and Simulation in Engineering*, 3, pp. 166-169.
20. Yeh, T.-P. and Vance, J. M., 1998, "Applying Virtual Reality Techniques to Sensitivity-Based Structural Shape Design," *ASME Journal of Mechanical Design*, 120, pp. 612-619.
21. Ryken, M. and Vance, J. M., 2000, "Applying Virtual Reality Techniques to the Interactive Stress Analysis of a Tractor Lift Arm," *Finite Elements in Analysis and Design*, 35, pp. 141-155.
22. Chen, J. S., et al., 2001, "A Stabilized Conforming Nodal Integration for Galerkin Mesh-free Methods," *International Journal for Numerical Methods in Engineering*, 50, pp. 435-466.
23. Chipperfield, K., Vance, J. and Fischer, A., 2004, "Fast Meshless Reanalysis Using Combined Approximations, Pre-Conditioned Conjugate Gradient, and Taylor Series," *AIAA Journal*, (submitted).
24. Bierbaum, A., et al., 2001, "VR Juggler: A Virtual Platform for Virtual Reality Application Development," *IEEE VR 2001*, Yokohama, Japan.
25. Neider, J., Davis, T., and Woo, M., 1993, *OpenGL Programming Guide*, Addison Wesley.
26. MacCracken and Joy, 1996, "Free-Form Deformations With Lattices of Arbitrary Topology," *SIGGRAPH 96*, pp. 181-188.
27. Snir, M., et al., 1998, *MPI -The Complete Reference Volume 1,* The MPI Core 2nd ed., MIT press.