

# Discrete Event Simulation Implemented in a Virtual Environment

Jason Kelsick

Judy M. Vance

e-mail: jmvance@iastate.edu

Department of Mechanical Engineering,  
Virtual Reality Applications Center,  
Iowa State University,  
Ames, IA 50011

Lori Buhr

Cheryl Moller

e-mail: cmw@iastate.edu

Department of Industrial and  
Manufacturing Systems Engineering,  
Iowa State University,  
Ames, IA 50011

*Virtual reality (VR) technology provides a human-computer interface that allows participants to interact naturally with digital objects which are represented as three-dimensional images that occupy positions in a three-dimensional world. Related to problems of engineering design and manufacturing, this new technology offers engineers the ability to work with computer models in a three-dimensional, immersive environment. This paper describes a virtual reality application where the results of a discrete event simulation of a manufacturing cell are integrated with a virtual model of the cell to produce a virtual environment. The program described in this paper, the VRFactory, combines results from a commercial discrete event simulation program, SLAM II, with a virtual environment. This allows the user to investigate, using three-dimensional computer models, how various changes to the manufacturing cell affect part production. This investigation is performed while immersed in a computer-generated three-dimensional representation of the cell. Existing discrete event programming software allows only two-dimensional views of the factory as the parts progress through the simulation. Parts are shown only as primitive geometric shapes on the computer monitor and instantaneously move from one station to the next. The virtual environment implementation of the SLAM II results allows users to experience the simulation in a fully immersive three-dimensional digital environment. The virtual environment used here is a CAVE™-like projection screen-based facility called the C2, which is located at Iowa State University. This paper describes the creation of the VR model of the manufacturing cell, the animation of the environment and the implementation of the results of the discrete event simulation. [DOI: 10.1115/1.1587745]*

## Introduction

Virtual reality techniques have been used to model manufacturing facilities and operations for a wide variety of applications [1]. These manufacturing applications can generally be categorized into four areas: *process simulation*, *factory layout*, *assembly methods prototyping* and *part flow simulation*. Maiteh et al., [2] have developed methods of using swept-volume computations to generate tool paths in NC machining simulations to facilitate virtual process simulation. Virtual tools have been developed by Kesavadas and Subramaniam [3] and Kesavadas and Cannon [4] to aid in operator programming of complex machinery. These tools carry with them attributes related to the desired manufacturing task. Ebbesmeyer et al., [5] have used virtual reality for layout design of a complex European power plant to facilitate design early in the design process. Bullinger et al., [6] have developed a virtual assembly planning tool that is coupled with a CAD system for easy input/output of part models.

Assembly methods prototyping holds great promise for the use of virtual reality. Jayaram et al., [7] have developed VADE, the Virtual Assembly Design Environment, to prototype assembly methods. Using VADE, a designer can examine how humans interact with parts in the assembly process. Physically-based modeling underpins the behavior of the virtual parts as they are assembled so realistic assembly methods prototyping can be performed in the virtual environment. Fernando et al. [8] are also evaluating the use of constraint-based virtual environments to support assembly methods prototyping. McDermott and Bras [9] as well as Evans et al. [10] are evaluating the use of haptics to provide more realism to the user when performing a virtual assembly task.

The work most closely related to the research presented in this

paper has been done by Kesavadas and Ernzer [11], at the University of Buffalo. They have developed the VR-Fact! program which provides an immersive virtual factory environment that can be used to investigate part flow through the factory. Our work provides similar functionality but the approach is different in two ways. First, we chose to develop an interface to existing commercially available discrete event programming software instead of incorporating the discrete event logic within the virtual environment software. The rationale for this decision is that the commercial discrete event programming software has been modified and improved over many years such that it is very robust and versatile. Instead of re-writing this functionality we chose to use the discrete event programming software to its best advantage. In this way, both software applications--the discrete event software and the virtual environment--can be optimized to their full potential. The other difference from VR-Fact! is in the implementation in the virtual environment. While both VR-Fact! and the VRFactory can be run using a head-mounted display or stereo glasses with a computer monitor, only the VRFactory can also be run in the C2, the CAVE™-like projection system located at Iowa State University. Our experience has found this environment very conducive to collaborative engineering. Several people can enter the C2 and investigate the results of the virtual simulation while discussing possible design changes.

**Factory Simulation.** Defined by Pritsker [12], "in its broadest sense, computer simulation is the process of designing a mathematical-logical model of a real system and experimenting with this model on a computer." In discrete event simulation, the dependent variables must change at distinct times, thus forming events. In other words, the state of the simulated system can only change at event times. In use with manufacturing, discrete event simulations model part flow through a manufacturing process. The part flow is divided into a series of events with event times. The

Contributed by the Design Automation Committee for publication in the JOURNAL OF MECHANICAL DESIGN. Manuscript received February 2002; revised March 2003. Associate Editor: J. E. Renaud.

simulation can determine bottlenecks, machine tool usage, material handling problems and other issues before they occur.

Computer simulation was realized as a potentially useful tool for industry in the late 1950's and early 1960's. It allowed industries to test configurations of manufacturing systems before purchasing and implementing the actual equipment. Since then, many simulation programming languages (SPLs) have been created and improved. In 1961, GASP (General Activity Simulation Program) was developed by Philip J. Kiviat at the Applied Research Laboratory [13]. Two major SPLs used today are descendants of GASP. They are SLAM II (Simulation Language for Alternative Modeling), produced by Pritsker and Associates, Inc. [12], and SIMAN (SIMulation ANalysis), developed by C. Dennis Pegden. Both SPLs were developed in the late 1980's and have since then become major components of analysis and research for many industries.

The past few years have seen computer simulation development take a new direction. Previous versions of simulation software produced only text-based output. The increased capabilities of today's computers enable display of complex graphical simulations. Pritsker and Associates, Inc. have created AweSim, an interface tool that creates a graphical user interface for SLAM II and also allows for the integration of outside programs and databases. Included in this version is the capability to create two-dimensional graphical animations. Animation helps the user visualize the simulation, although it is limited to a two-dimensional display. Another computer simulation package, developed by Delmia, is called QUEST® (QUEuing Event Simulation Tool). QUEST® includes a three-dimensional graphical animation that allows for a more immersive environment than the two-dimensional animation provided by AweSim.

The VRFactory program described in this paper is an animated, immersive, three-dimensional model of a manufacturing work cell. A manufacturing process is simulated in the VRFactory by implementing a discrete event simulation of the process. A simulation model of the work cell, created using AweSim, provides the discrete event simulation data to the virtual environment. The topic of this paper is how the VRFactory was created and how the results from AweSim are used in the VRFactory. The steps required in the creation of the VRFactory include the modeling and animation of the machines, tools and parts, and the implementation of the results of the discrete event simulation.

## The VRFactory Environment

**Peripherals.** The VRFactory can be viewed in several different environments: in a head-mounted display (HMD), on a single stereoscopic projection screen, or in Iowa State University's C2. The C2 is a 12'×12'×9' room where stereo images are projected onto three walls and the floor. When viewed on the projection screen or in the C2, stereo vision is obtained through CrystalEyes stereo shutter glasses. Interaction with the virtual environment (created by any of the display devices mentioned above) is achieved through a Fakespace PINCH Glove. The Fakespace PINCH Glove records contact between a user's fingers so that various hand gestures can be used to control movement and virtual menu selections. The user's viewpoint and hand position are tracked with Ascension Flock of Birds magnetic trackers. When VR peripherals are not available, the user can interact with the program using a standard monitor and mouse. Figure 1 shows a user interacting with the VRFactory in the C2.

**Navigation and Interaction.** Since the factory floor space is larger than the workspace of the virtual environment, tracking the viewpoint position with the Flock of Birds alone does not allow the user to examine the entire work cell. Extra navigation is needed to move to a viewpoint outside the physical workspace of the virtual environment. The PINCH Glove gestures not only en-



Fig. 1 VRFactory in the C2

able this extra navigation; they also allow navigation to be independent of head tracking. The user can look off to one side, yet use the PINCH Glove to navigate forward.

To navigate, the user reaches out, touching the index finger to the thumb, and pulls inward while holding the finger and thumb together. A good analogy would be grabbing a rope and pulling yourself along (Fig. 2). Since the C2 only displays on three projection walls, turning around and navigating in the direction behind the users is not possible. To overcome this difficulty, rotating the world was implemented. Touching the middle finger to the thumb activates a constant clockwise rotation of the world. Touching the ring finger to the thumb rotates the user's viewpoint counter-clockwise.

Interaction with the VRFactory is through a three-dimensional menu that can be positioned anywhere in the virtual space. To make the menu appear the user makes a fist (making contact with the tips of all four fingers to the palm of the hand). The options on the menu list different possible simulations of the same factory work cell (Fig. 3). To choose a particular simulation, the user must intersect the virtual hand with the menu option and make a gesture

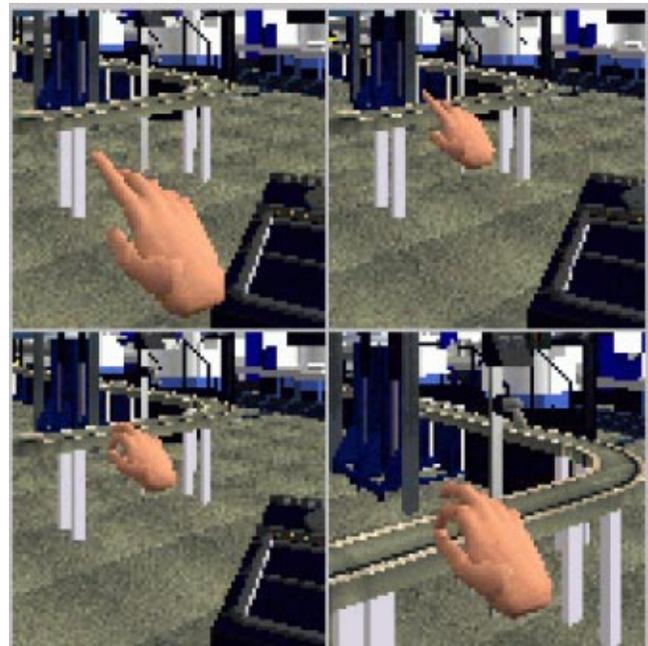


Fig. 2 Navigation in the virtual environment



Fig. 3 Virtual menu

with the index finger and thumb. Once the option has been chosen, the menu disappears, and the simulation begins.

Probably the most important and obvious aspect of the visualization of a simulation is the creation of the geometric models (three-dimensional representations) of the machines, tools, and parts involved in the simulation. The following section describes the process by which the geometric models for the VRFactory were created.

**Creation of the Factory Models.** The first step in creating the geometric models was acquiring all of the necessary measurements and dimensions. Floor plans and dimensional drawings of the machines were used as references, but since the VRFactory is modeled after a factory located in the area, dimensions were also acquired by measuring the actual objects. A portion of the geometry in the VRFactory is shown in Fig. 4.

CAD software (Pro/Engineer®) and modeling software (World Up™ Modeler, MultiGen® II) were used to build the models. The initial CAD files were saved in the native format of either Pro/Engineer® or MultiGen® II and then translated into a format best suited for the virtual environment management software. Material properties such as color, shininess, and emissive properties as well as textures were assigned to the models using the World Up™

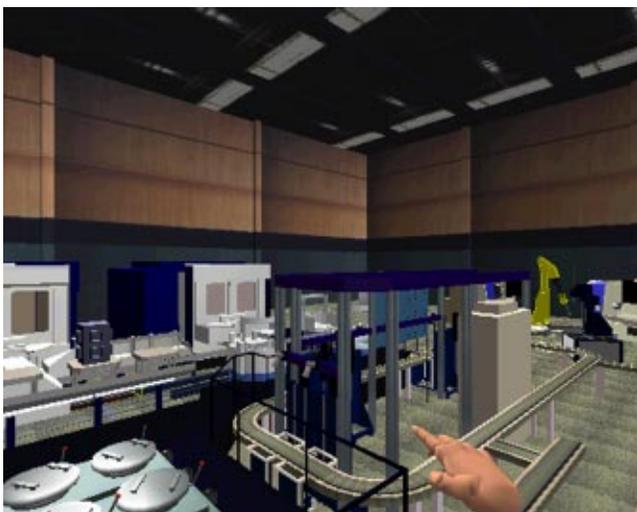


Fig. 4 VRFactory models

modeling software which was fully compatible with the virtual environment software. These properties were assigned to the models to make the models appear more realistic. The textures were created from digital pictures of the actual machines or created from scratch using Adobe PhotoShop. One consideration in preparing the models was to use the fewest polygons necessary in order to maintain real time display.

Once the models were created, they were loaded into the application and translated to either the position defined by the floor plan or, if that was not possible, to a position relative to other objects in the virtual factory. The software development system used to help create the application and manage the virtual environment was WorldToolKit from Sense8/Engineering Animation, Inc. (EAI). WorldToolKit is an object-oriented library written in C++ . It allows for the creation or importation of realistic models into a scene hierarchy that enables proper animation in real time.

As more 3D models were loaded into the VRFactory application, the display speed decreased. To maintain real time display speed, level of detail (LOD) models were introduced. LOD is "modeling the same object at different detail levels and the appropriate one is chosen for display based on some viewing criteria and system performance" [14]. In other words, if the viewpoint of the user is close to a particular object in the factory, a detailed version of the object is displayed. If the user is far enough away from the object so that the smaller features could not be distinguished, a simpler version of the model is displayed [15]. With the combination of reduction of polygons, limited use of textures and implementation of LOD, an efficient model of the factory work cell was created.

**The Discrete Event Simulation.** The flexible machining work cell consists of four Makino Machining Centers, an Enshu horizontal machining center, a rail-guided vehicle, a work in process (WIP) conveyor, a Cascade deburring machine, a wash and hone center, and an automatic service and retrieval system (ASRS) (Fig. 5).

The function of this work cell is to machine housings for hydraulic pumps. Generally, the pump housings start in the ASRS, are transferred to the WIP conveyor, are fixtured and then transferred to the rail-guided vehicle. This vehicle takes the parts to a waiting station prior to their loading into a machining center. After machining, the parts are taken by rail-guided vehicle back to the

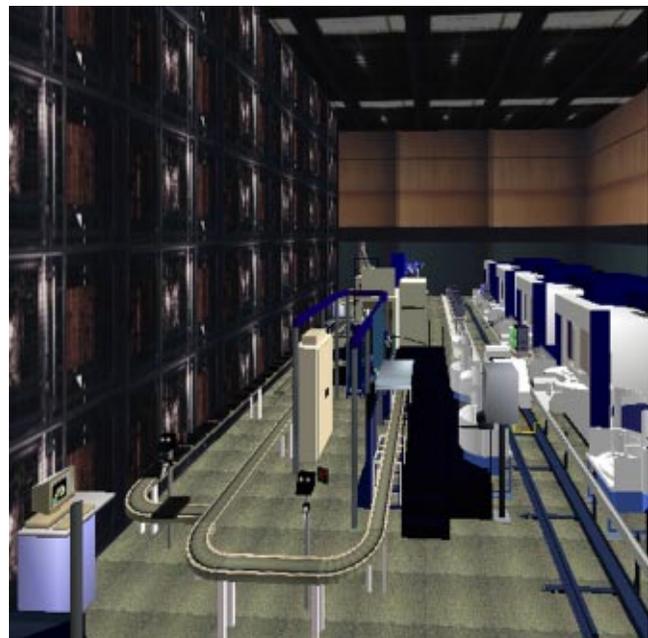


Fig. 5 The VRFactory environment

fixturing station where they are removed from the fixture and loaded back on the WIP conveyor. The parts are then transferred to the deburring machine, then to the wash and hone center, and are finally loaded back into the ASRS for storage. The simulation implemented in the VRFactory was limited to using only the ASRS, WIP conveyor, fixture station, rail-guided vehicle and machining centers. Every entity was not included in the final version of the VRFactory because the purpose of this project was just to develop a prototype VRFactory. The structure of the VRFactory program allows for additional modifications and could accommodate the deburring, washing, and honing machines in the future. In addition, there is a robot that loads parts into two lathes in the work cell area that could be included in the virtual environment.

There are 30 locations throughout the work cell which are used in the simulation to specify the location of the parts at any time throughout the work shift. In this application, the results of the discrete event simulation created using AweSim provide the results file which is the input to the virtual environment. Several different simulations can be run and then loaded into the virtual environment for examination. In this way the VRFactory is being used as a three-dimensional post-processing tool.

Time is the major independent variable in the simulation of the manufacturing process. In a typical day the Parts Manufacturing Technicians (PMTs) work three eight hour shifts. Each stage or movement of a part is defined by a specific starting time in the discrete event simulation file. Five part types were modeled in the simulation. The manufacturing personnel provided a data file with forty-five days of time study data from the work cell. ACCESS software was used to extract relevant data from this data file, and it resulted in eleven data sets of information. For each of the five part types, a data set was created for the Makino processing time and the fixturing time. The remaining data set contained information on the WIP conveyor time. ExpertFit software by Averill M. Law and Associates was used to develop a probabilistic model of part flow through the factory based on the collected data. This capability allows the simulation to more realistically represent the real factory results.

After the simulation was completed, existing counters and trace attributes were used to record the utilization of resources and the movement and location of entities. Supplementary counters and attributes were added to the simulation when additional information was necessary for the animation.

For comparison with the VRFactory implementation, a workstation-based animation was created directly from AweSim. This animation consisted of a line drawing of the plan view of the work cell. Squares representing parts are drawn at their locations according to the simulation results. The five part types are represented by different colored squares. When more than one part is located at a certain resource, either multiple boxes are drawn or a number appears on a single box. Parts jump instantaneously from one resource to another and are not actually animated through the factory. Because of space limitations on the workstation monitor it is difficult to show the location of each PMT and WIP pallet. Therefore, two regions, one representing PMT utilization and one denoting the number of WIP pallets in use, are filled and depleted accordingly. A clock giving the time in minutes is also provided.

**Table 1 Simulation resources**

	Resource	Label
1	Technician	PMT
2	Pallet	PLT
3	Makino 1	MAK1
4	Makino 2	MAK2
5	Makino 3	MAK3
6	Makino 4	MAK4
7	Fixturing Station 1	WSS1
8	Fixturing Station 2	WSS2
9	Makino Pallets for (1,2,3)	MCPLT123
10	Makino Pallets for 4	MCPLT4

**Table 2 Simulation sections**

Section	Action
1	ASRS to the WIP Conveyor
2	Single Part Machining Process
3	One-in-one-Out Machining Process
4	Gauging
5	Return to ASRS

**Implementation of the Discrete Event Simulation in the Virtual Factory.** The first step in the interfacing of the simulation (AweSim) and the VRFactory is defining a file format that both programs understand. An ASCII text file with customized information was exported from AweSim. This text file contained information on 10 different resources, including the ASRS, WIP conveyor, Makinos, fixturing stations, parts technician, and pallets for 4 part types (Table 1).

The AweSim simulation was divided into 5 sections which are listed in Table 2.

The text file from AweSim contained the time and location of each movement event in the simulation (Fig. 6). Write nodes, which were used to create this output file, were placed throughout the simulation whenever an event occurred. The VRFactory program was written to read in the text file and interpret the input data. Movement of parts throughout the factory was programmed to occur between discrete event points. For instance, part type 1 would show that at an elapsed time of 10.02 min the part would move from gauging to fixturing. The discrete event simulation reflected an instantaneous movement. In real life this movement takes an interval of time. It was up to the programmer of the VRFactory to determine how to implement the movement of the part from gauging to fixturing and to decide how much of the time interval to spend in this process. This generally was a trial and error process. Some time would be taken from the previous station and some time would be taken from the following station. The result was to take a discrete event and turn it into a continuous product flow throughout the factory.

**Animation of the Factory Models.** Animation of the machines in the virtual environment required knowledge of how the actual machines and tools worked and the speed at which they operated. Most of this knowledge came from observing the real

Part Type	Begin ASRS	Begin WIP	WSS	Exit WIP	Exit WSS	Mill #	Exit Mill	Gauge	Return to WIP	End time
3	0	4.42	8	4.42	17.75	4	140.34	0	140.34	232.09
3	0	4.42	8	4.42	17.75	4	140.34	0	140.34	232.09
3	0	9.7	7	10.05	23.38	5	145.97	0	145.97	232.09
3	0	9.7	7	10.05	23.38	5	145.97	0	145.97	232.09
3	10	14.07	8	17.75	65.61	6	208.76	0	208.76	232.09
3	10	14.07	8	17.75	65.61	6	208.76	0	208.76	232.09
3	20	22.37	8	65.61	140.34	4	216.09	0	216.09	232.09
3	20	22.37	8	65.61	140.34	4	216.09	0	216.09	232.09
3	20	26.09	7	71.34	145.97	5	221.72	0	221.72	232.09
3	10	17.14	7	23.38	71.34	3	232.09	0	232.09	232.09
3	10	17.14	7	23.38	71.34	3	232.09	0	232.09	562.31
3	20	26.09	7	71.34	145.97	5	221.72	221.72	249.72	562.31

**Fig. 6 AweSim output file**

factory work cell. Observing the actual work cell helped to develop more realistic animation in the virtual environment.

The program's importation of 3D models was based chiefly on how each individual machine in the virtual world would be animated. The geometric models in the VRFactory were created using a hierarchical data structure. Each entity in the program was a node related to other entities through either parent nodes or child nodes. Using this data structure, motion of the parent node is carried through to affect all the child nodes. For example, a door on a machining center was loaded into the program as a child node of the machining center node. This allowed for the door to be animated independently of the machining center, but if the location of the machining center were moved, the door would also move.

This hierarchical structure was also used to facilitate the simulation. As the simulation runs, the parts are attached (become child nodes) and detached from other nodes. When the part is sitting on a particular pallet it is attached to the pallet. If the pallet were to move, the part would also move. Detaching and then attaching the part to another node would then make the part move with this node. In this manner, the parts are "carried" throughout the manufacturing process.

Collision detection is implemented in the VRFactory only for certain static objects in the environment. Collision detection is used on the worker platform so that as the user reaches the platform, the user is required to step up onto the platform. Without the collision detection, the user would walk on the floor under the platform. Similarly, collision detection was built into the ladder in the application so users could move forward and climb the virtual ladder to get a bird's eye view of the factory.

## Results

Once the basic part flow logic was transferred from the AweSim output to VRFactory, multiple simulations could be loaded and examined. Built into the VRFactory was the ability to move ahead in time within the simulation period to examine the part flow at any point in the period without having to wait through the actual time period. The time of day was displayed in the environment as an aid. As the user moved to different times during the simulation period, the time of day was updated. The menu system allowed users to quickly examine multiple scenarios while immersed in the virtual environment. Immersion provided the users with a sense of the floor space occupied with waiting parts. It was obvious when comparing two scenarios that more parts spent more time waiting when there were fewer pallets available. Users were able to explore the simulation in the virtual environment and provide input on changes to make for the next simulation run.

After running several AweSim simulations, the users concluded that adding one more Makino machining center and increasing the number of pallets for each individual part number would increase the average throughput of the system from 57.4 parts per day to 80 parts per day.

## Conclusions

Integrating results from a discrete event simulation into a virtual factory model provides a three-dimensional environment in which to examine these results. The traditional workstation-based animation did not fully convey the movement of parts through a three-dimensional space. VRFactory allows a user to be completely immersed in a functioning factory work cell. Through this visualization tool, the user might see where a problem could arise instead of tracing a problem through charts and graphs of the simulation. Because of this, the VRFactory was a successful representation of a visualization tool for discrete event simulations.

The VRFactory also serves to illustrate the use of virtual reality to interface with existing commercial software simulations. Although the VRFactory was designed specifically to obtain input from AweSim, the methods presented here could be applied in

general to other discrete event programming software packages. These commercial software packages are widely used, robust, and have a significant number of features that have been developed over the life span of the software. Our approach to building virtual design tools for engineers is to take advantage of these software applications that are already highly-developed and specialized and use their results as the input to the virtual environment. In this manner, the specific software application is used for what it is designed for, and virtual reality is used to provide the three-dimensional immersive environment for engineers and others to explore this data.

Users were able to use the VRFactory to explore the discrete event simulation results and communicate implications associated with design changes to others not as familiar with discrete event programming. Immersion in the virtual factory facilitated the exploration of design changes and their effect on the simulation. There were many variables in this situation which could be changed for each iteration. The virtual factory experience guided the users into making informed decisions on where to look for the next improvement in simulation performance. In this way, the VRFactory was used as a computational steering aid.

## Future Work

There are many aspects of the visualization of a simulation that the VRFactory has not fully explored and are currently being developed. A direct coupling of the VRFactory simulation to the AweSim simulation software would allow users to make changes to the simulation while in the virtual environment and examine the effects. This would greatly enhance the usability of the system and provide an overall design tool.

A performance study should also be performed to determine the benefits of VR in the visualization of manufacturing simulations. The VRFactory could be compared against traditional workstation-based animation. Results of this type of study could justify the use of VR as a visualization tool for simulations.

## Acknowledgments

This work is supported through funding of the National Science Foundation project DMI-9525998. Equipment for this project is supplied by the Virtual Reality Applications Center at Iowa State University. The authors would like to thank Professor Thenkurussi Kesavadas from the University at Buffalo for his work creating the initial models of the machining centers while he was employed at Iowa State University.

## References

- [1] Cruz-Neira, C., 1993, "Virtual Reality Overview," *ACM SIGGRAPH '93 Notes: Applied Virtual Reality*, ACM SIGGRAPH '93 Conference, Aug 1–6, Anaheim, CA.
- [2] Maiteh, B. Y., Leu, M. C., Blackmore, D., Liu, G., and Abdel-Malek, L., 1999, "Swept-Volume Computation for Virtual Reality Application of Machining Simulation," *Proceedings of Industrial Virtual Reality: Manufacturing and Design Tool for the Next Millennium*, Nov 1–2, Nashville, TN, ASME publications, MH-Vol. 5, pp. 3–10.
- [3] Kesavadas, T., and Subramaniam, H., 1998, "Virtual Tools with Attributes for Programming Robotic Finishing Operations," *International Journal of Industrial Robotics*, **25**(4) June, pp. 268–275.
- [4] Kesavadas, T., and Cannon, D., 1996, "Virtual Tools with Attributes for Robotic Based Intermediate Manufacturing Processes," *Proceedings of the IEEE International Conference on Robotics and Automation*, Apr 22–28, Minneapolis, MN, Vol. 3, pp. 1845–1850.
- [5] Ebbesmeyer, P., Gehrman, P., Grafe, M., and Krumm, H., 1999, "Virtual Reality for Power Plant Design," *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, Sep 12–15, Las Vegas, NV, DETC99/CIE-9037.
- [6] Bullinger, H. J., Richter, M., and Seidel, K.-A., 2000, "Virtual Assembly Planning," *Human Factors and Ergonomics In Manufacturing*, **10**(3) Summer, pp. 331–341.
- [7] Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., and Hart, P., 1999, "VADE: A Virtual Assembly Design Environment," *IEEE Comput. Graphics Appl.*, **19**(6) pp. 44–50.
- [8] Fernando, T., Wimalaratne, P., and Tan, K., 1999, "Constraint-based Virtual Environment for Supporting Assembly and Maintainability Tasks," *Proceed-*

- ings of the 1999 ASME Computers in Engineering Conference, Sep 12–15, Las Vegas, NV, DETC99/CIE-9043.
- [9] McDermott, S. D., and Bras, B., 1999, "Development of a Haptically Enabled Dis/Re-assembly Simulation Environment," *Proceedings of the 1999 ASME Design Engineering Technical Conferences*, Sep 12–15, 1999, Las Vegas, NV, DETC99/CIE-9035.
- [10] Evans, D. J., Feddema, J. T., Johnson, W. A., Jayaram, S., Jayaram, U., and Seywald, H., 2000, "Haptic Display of Interaction Forces in MEMS Assembly Processes," *Proceedings of the 2000 ASME Design Engineering Technical Conferences*, Sep 10–13, Baltimore, MD, DETC2000/CIE-14581.
- [11] Kesavadas, T., and Ernzer, M., 1999, "Design of Virtual Factory Using Cell Formation Methodologies," *Industrial Virtual Reality: Manufacturing and Design Tool for the Next Millennium*, *Proceedings of the ASME Symposium on Virtual Environment for Manufacturing*, Nov 14–19, Nashville, Tennessee, MH-Vol. 5, pp. 201–208.
- [12] Pritsker, A. B., O'Reilly, J., and LaVal, D., 1997, *Simulation With Visual SLAM and AweSim*, Systems Publishing Corporation, West Lafayette, Indiana.
- [13] Nance, R. E., 1995, "Simulation Programming Languages: An Abridged History," *Proceedings of the 1995 Winter Simulation Conference*, Dec 3–6, Arlington, VA, pp. 1307–1313.
- [14] Chen, S. E., 1995, "QuickTime® VR-An Image-Based Approach to Virtual Environment Navigation," *Computer Graphics: Proceedings of SIGGRAPH '95*, Aug 9–11, Los Angeles, CA, pp. 29–38.
- [15] Fleischer, K., Laidlow, D., Currin, B., and Barr, A., 1995, "Cellular Texture Generation," *Computer Graphics: Proceedings of SIGGRAPH '95*, Aug 9–11, Los Angeles, CA, pp. 239–248.