

# A multichannel visualization module for virtual manufacturing

Yong-Sik Kim<sup>a,1</sup>, Jeongsam Yang<sup>b,\*</sup>, Soonhung Han<sup>a,1</sup>

<sup>a</sup> Department of Mechanical Engineering, Korea Advanced Institute of Science & Technology, 373-1, Gusong-Dong, Yusong-Gu, Daejeon 305-701, Republic of Korea

<sup>b</sup> Industrial & Information Systems Engineering, Ajou University, San 5, Wonchun-dong, Yeongtong-gu, Suwon 443-749, Republic of Korea

Received 30 June 2005; accepted 15 February 2006

Available online 2 May 2006

## Abstract

The immersive virtual reality (VR) for manufacturing planning helps to cut down the product development period and to improve the quality of the production. However, the immersive VR equipments are generally expensive, both in terms of development and buying. Users spend time to manually repair complex 3D shapes because of imperfect translations between 3D engineering CAD models and the proprietary format of the VR system. In this paper, the proposed VR module uses a commercial virtual manufacturing system (VMS) as the viewer of the immersive VR system on a cluster of PCs and adopts the modified simulation algorithm. The module can make the data translation process unnecessary and ensure good coherence under simulation. At experiment, the proposed immersive VR module was interfaced with the Delmia VMS. Its clustering modules can reduce the cost of VR experiment while offering high performance.

© 2006 Elsevier B.V. All rights reserved.

**Keywords:** CAD; Immersive virtual reality; Multichannel visualization module; PC clusters; Virtual manufacturing system

## 1. Introduction

The recent combination of virtual reality (VR) technology with virtual manufacturing systems (VMSs) enhances competitiveness in areas such as the automotive and shipbuilding industries. Designers can control virtual machines inside a virtual factory and evaluate digital products in advance. These evaluations or verifications can help to eliminate unexpected errors, thereby reducing the overall process time and cost.

Before the VMS has been introduced, decisions regarding job shops are made based on 2D shapes displayed on small monitors [1]. The process of abstracting 2D models from real factories is prone to errors due to the incomplete abstraction of the mathematical model and the limitations of the 2D model itself [2,3]. Considerable experience and cost are required for accurate results.

As a countermeasure, companies expand into 3D VMSs that more accurately represent the actual factory layout [4,6]. This

method, however, still cannot effectively deliver a sense of reality nor enable users to correctly grasp the situation because the large and complex VMS results are displayed on a small monitor with a low resolution. The experiment shows that editing process is easier with mouse and keyboard and error-check work is easier with immersive VR [5]. Because the main purpose of the VMS is the verification of the proposed design, the immersive VR will be better than general monitors for the VMS. Therefore, some VMS companies are shifting to immersive VR technology [7–9].

There are many various immersive VR technologies such as head-mounted display (HMD), Reality Center<sup>TM</sup>, CAVE<sup>TM</sup> and so on. The designers and researchers at work prefer CAVE<sup>TM</sup> or Reality Center<sup>TM</sup> to HMD, for large-scale screen is more convenient to communicate and cooperate with other departments.

The method to control the large-scale screen has two ways. One is to use a graphic-specialized hardware like Onyx, Octane and the other is to use the cluster of PCs. Because the cluster of PCs is cheap and flexible enough to meet various environments, the cluster of PCs is a recent prevailing method.

However, applying immersive VR with the cluster of PCs to the VMS has a problem. The problem is that because the simulation process at VMS is higher priority than the clustering

\* Corresponding author. Tel.: +82 31 219 1879; fax: +82 31 219 1610.

E-mail addresses: [mk37do@icad.kaist.ac.kr](mailto:mk37do@icad.kaist.ac.kr) (Y.-S. Kim), [jyang@ajou.ac.kr](mailto:jyang@ajou.ac.kr) (J. Yang), [shhan@kaist.ac.kr](mailto:shhan@kaist.ac.kr) (S. Han).

<sup>1</sup> Tel.: +82 42 869 3040; fax: +82 42 862 9224.

process, the clustering process cannot use CPU enough to ensure the coherence of all the screens at the system.

And, the VMS has the following problem, too. When immersive VR technology is applied to a commercial VMS, such as Delmia of Dassault Systemes or e-Factory of UGS, the conversion of the different data formats between the immersive VR systems and the VMS causes data loss and distortion [12].

We investigate the above problems, and we propose a multichannel visualization system to reduce the problems. Two improvements are proposed for the multichannel visualization system. In the first approach, we use a commercial VMS as the viewer by adding immersive VR modules into the VMS. This approach does not require data conversion process and this enables us to solve the data loss and errors that occur while converting the data from the VMS to the VR systems.

In the second approach, we propose the modified simulation algorithm for the coherence of the VMSs on a cluster of PCs. Because the simulation engine at the VMS has the exclusive right to use the CPU, the cluster module does not have the right to access the VMS for coherence during the simulation. The modified simulation algorithm divides the simulation into many short-time simulations, and it has the VMS deal with the short-

time simulations and lets cluster module have the right for coherence at intervals between short-time simulations. In this case, many short-time simulations look like a seamless simulation, and the discrepancy between the screens are too small for users to recognize it.

## 2. Related works

### 2.1. Review of previous studies

Many researches on the VMS have focused on proving the usefulness of VR. Table 1 compares the related works and our method. Bazargan-Lari [10] sketched a small layout of a virtual factory on a 2D monitor and found the optimized layout for equipment by using mathematical algorithms.

Weyrich and Drews [4] developed a workbench for a virtual manufacturing factory. The workbench is a virtual working stand that enables 3D projections of virtual objects. The user can observe a 3D factory from the workbench by wearing LCD shutter glasses, and can interact with the workbench system through a six-dimensional input device. The workbench system, which comprises hierarchical object structures, can

Table 1  
Comparison of the related works

	Bazargan-Lari [10]	Weyrich and Drews [4]	Korves and Loftus [5]	Ng et al. [6]	Choi et al. [11]	Our method
Hardware	2D monitor	Workbench, LCD shutter glasses (stereoscopy)	SGI Indigo2	HP workstation	2D monitor	Multichannel visualization system
OS	N/A	IRIX	IRIX	HP-UX	Windows	Windows
Input device	Keyboard	Space joystick, Dataglove, 3D mouse	HMD, 2D or 3D mouse	HMD, 3D mouse	2D or 3D mouse, keyboard	Keyboard
Visualization method	2D display module	Proprietary library	N/A	N/A	OpenGL	Delmia viewer
VMS interface	N/A	Partially handles layout issues in a virtual environment	N/A	N/A	Provides programming environment for VMFactory	Interfaces with Delmia
Data format	N/A	Uses CAD format such as DXF and IGES after repairing. Use VR format such as VRML, OpenInventor and OpenFlight	N/A	Additional repairing is required during conversion. The results of the cable layout can be converted to DXF	Proprietary data format. Convertible to VRML format but additional repairing is required	Convertible with Delmia format
Features	Mathematical approach for layout optimization	Focuses on building a hardware interface and virtual environment	Focuses on the usefulness of virtual reality for layout planning	Focuses on the cable harness design and planning within virtual environments	Provides modeling of virtual equipment and factory layout, command language or model-based simulation and schedule simulation	Uses a multichannel screen based on a cluster of PCs for a better immersive experience. Implements control modules that can be interfaced with a VMS or CAD system
	Focuses on finding effective algorithm for equipment layout	Applicable to the level-of-detail				
		Uses multigen to modify 3D shape				

handle large amounts of shape data, level-of-detail (LOD), texture mapping. Weyrich et al. used the Onyx system from SGI as the hardware for the immersive environment, and are visualized through its own libraries. The study of Weyrich et al. focused on expanding the application fields of VR. Additional procedures, such as data conversion, are required if the data created by other VMSs is used within the workbench system.

To arrange manufacturing cell equipment, Korves and Loftus [5] developed an interactive layout planning system with HMD. The system uses an immersive interface in which the necessary equipment is selected and moved from the database to the workspace, and in which the working environment is arranged according to the user's choice. Whenever the user violates pre-defined constraints, audiovisual feedback is provided. To show the usefulness of VR, Korves et al. compared immersive VR – for which they used a HMD and a 3D mouse – with non-immersive VR – for which they used monitors and a normal mouse. According to the comparison, when the user has experience with the layout planning and is familiar with immersive VR, the immersive VR environment proved to be better at error checking than the non-immersive-type environment.

Ng et al. [6] used a virtual environment to investigate the design and planning of a cable harness. They designed automated cable routes by using a method of artificial intelligence to infer the 3D space. The user observes 3D images through a HMD, uses a 3D mouse for navigation and input, and provides haptic feedback with audiovisual expressions to detect collisions.

Choi et al. [11] developed a modeling and simulation solution *VMFactory* for virtual factory applications. Based on OpenGL, *VMFactory* provides functions such as modeling of virtual equipment and the factory layout, command language or model-based simulation and schedule simulation, which is connected to the manufacturing execution system. However, in *VMFactory*, no consideration is given to immersive VR.

## 2.2. Commercial VMSs

Delmia of Dassault Systemes can provide a pre-evaluation of the problems that can occur during the manufacturing process through simulations based on a virtual factory. When we include human-like avatars in the process and enable the avatars to imitate the user's actions, Delmia provides an environment in which actual working can be inferred.

The e-Factory of UGS can arrange and analyze the equipment of a virtual factory. The e-Factory comprises of FactoryCAD for designing the layout and FactoryFlow for analyzing the layout. FactoryCAD can provide the optimal layout by selecting information from the library of the shapes of the cranes, conveyors and robots. It also provides purchase costs, power, cycle time and connection time. FactoryFlow helps to understand the process of layouts in a virtual factory and it provides improvement ideas by graphically showing costs and usage to enable the user to analyze the flow of products during the manufacturing process.

But Delmia and e-Factory provide visualization of process simulations on desktop monitors. Expensive dedicated hardware such as the Onyx system of SGI is required to build on immersive VR. To visualize the model created in other VMSs in an immersive VR system, we must convert the data into a neutral format such as VRML. The process of converting the data, however, can introduce distortions and data loss depending on the complexity of the shapes [12]. For kinematics information, the user needs to input additional information to avoid an incomplete conversion. For instance, the movement of the camera should follow the movement of the axis, but the camera is found to behave separately after conversion. Various studies have been conducted on neutral formats that solve these conversion problems.

Page and Smith [13] introduced a methodology that can convert military data using Synthetic Environment Data Representation and Interchange Specification (SEDRIS). However, standardization of SEDRIS is underway, and there are not enough utility tools.

## 2.3. Cluster-based VR libraries

There are VR libraries that allow PC clusters to produce synchronized views of the VR environment frame-by-frame. The NAVERLIB library, which is based on the OpenGL Performer, has the advantages of easy-to-use editing functions and compatibility with multimedia [14]. CAVELib, which runs on a pair of networked SGI computers supports for VR clustering [15]. The library shares fixed data through the cluster, such as the navigational matrix and input device values. The DIVERSE library supports networking infrastructure while it does not explicitly support clustering [16,17]. DIVERSE provides a framework for networked shared memory that can be used to connect input devices to a VR application, or to construct shared virtual environments. VR Juggler, a VR library similar in function to the CAVELib, provides two extensions to support clustering: Net Juggler and Cluster Juggler [18]. In each case, a full copy of the VR Juggler application runs on each render node of the cluster. Applications share input events and are synchronized at the end of each drawn frame. And some VR systems like R3-Interactive [20] focuses only on controlling display devices to make seamless display. If the overlapped or curved screens are used, then this kind of system is useful.

Table 2 compares four cluster-based VR libraries. For visualization, these libraries use graphic libraries such as OpenGL or OpenGL Performer. Some of the VR libraries can be connected to immersive equipment or have network functions that support immersive features. VR libraries cannot directly read engineering data such as CAD, and therefore the data must be converted into a neutral format such as the VRML.

## 3. Design of multichannel visualization module

### 3.1. System requirements

To build a multichannel visualization system based on a cluster of PCs, we need to synchronize time for the screen

Table 2  
Comparison of four cluster-based VR libraries

	NAVERLIB	Net Juggler	CAVELib	DIVERSE
OS	Redhat Linux, Windows	SGI IRIX, Linux, Windows	SGI IRIX, Linux, Windows	SGI IRIX, Linux
VR devices	CAVE-like system, I/O device	Trackers, datagloves, I/O device	A variety of VR equipment	HMD, tracker, joystick, motion base
Graphic library	Supports multichannel functions  OpenGL Performer or OpenGL-based visualization	OpenGL-based visualization	OpenGL Performer or OpenGL-based visualization	OpenGL Performer-based visualization
Data format	Compatible with OpenGL Performer	N/A	Compatible with OpenGL Performer	Compatible with OpenGL Performer
Features	Easily manipulated with XML script Supports rendering functions of 3D video signal  Enables a cluster of PCs to be built and synchronized	Supports the cluster for VR Juggler	Suitable for a multichannel system	Supports CAVE and immersive desk Implements high-end applications such as a VR juggler

switching at animation and ensure the consistency of the property information so that several small screens can logically function as one large screen.

### 3.1.1. Synchronization of time for screen switching

When a model is displayed with multichannel screens, each screen displays different parts of the model. There are differences in the graphic load that each PC needs to handle. The difference in the graphic load causes time delays for completing the rendering of certain frame on each PC. As a result, a PC with heavy load updates frame more slowly. The user only sees screens that seem disjointed instead of logically synchronized images. Fig. 1 shows a model of landing gear on three screens. Differences in the rendering time occur because

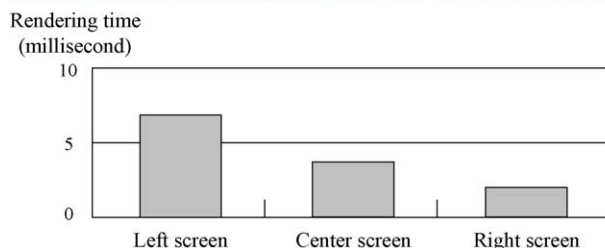
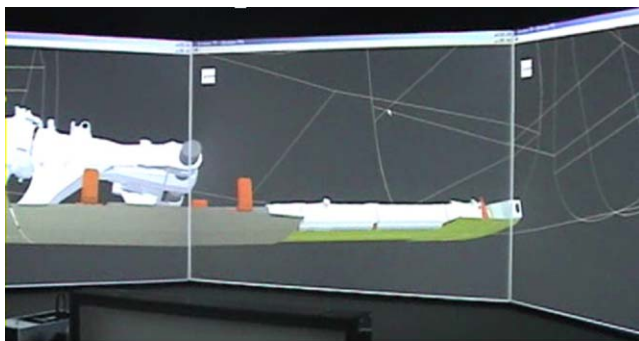


Fig. 1. Differences in rendering time due to the graphic load.

the graphic load on the left screen is greater than that of the middle and right screens, thereby causing time delays. The speed of rendering is 7 ms for the left screen, 4 ms for the middle screen and 2.8 ms for the right screen, and these differences cause the model to appear irregularly. Network traffic problems may also disrupt the synchronization of the screen switching. Sending large amounts of visualization data can slow the network down because the network's bandwidths are limited. The refreshed data should be transmitted to the slave PCs after being confirmed by the master PC each time a frame is displayed.

When a cluster of PCs is used to display a single image, all the screens connected to the PCs must be refreshed simultaneously as if they form a single screen. The images should not be projected on the screen immediately after being drawn on frame buffers. Rather, the images should be synchronized with other frames. To display the images from each PC on several independent screens, we need to synchronize the connected screens.

To allow simultaneous display of frames after corresponding images have been displayed on each PC, the master PC sends a signal notifying each slave PC that the display of frames has been completed. When all the images are drawn on frame buffers of all the PCs, the simultaneous swap buffering, allows the entire image to be synchronized. Although the time that each image data is drawn to each frame buffer may differ, the screen display can be performed simultaneously.

### 3.1.2. Consistency of property information

To achieve a single logical display, we need to synchronize the property information too; for example, color property. We also need to adjust the field of view (FOV) from various angles. Fig. 2 shows an example of inconsistent property information. The left scene is what user wants to see. Image transformation by a user's input influences only the left picture but does not the

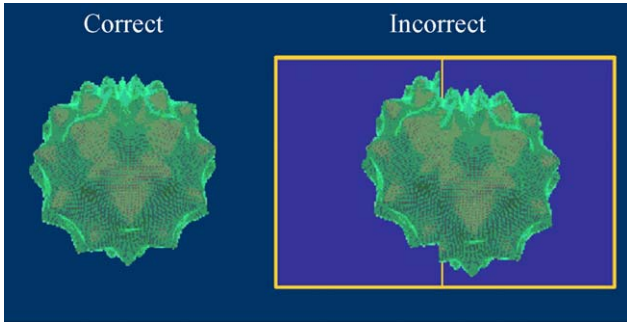


Fig. 2. Example of inconsistent property information.

right picture. This inconsistent property is because the information held in both slave PCs are not updated consistently.

For a virtual factory with a multichannel system, various situations should be considered, such as dynamic movement of equipment, color changes that reflect the condition of equipment and changes of FOV. The changes in property should not be limited but should be synchronized to allow consistent display under a multichannel system. The master PC transmits consistent properties to slave PCs through the network.

### 3.2. Module design

To construct an immersive VR environment, we interface the commercial VMS Delmia with a multichannel visualization system based on a cluster of PCs.

#### 3.2.1. Design of the control module

As shown in Fig. 3, the multichannel visualization system consists of a master PC, which controls the system, and a cluster of several slave PCs.

Because of the high performance of today’s PCs, the network load is a factor that influences performance. When the model data are located only at the master PC, the master PC has to transfer both instructions and data. On the other hand, if the data are located at every slave PCs, transmitting only the

instructions and pointer to the data modified by the instructions enables the master PC to control all slave PCs faster. For example, the WireGL, a centralized multichannel library, requires network load so much that its animation looks like slow motion. Therefore, the distributed system can reduce the network load and accelerate the overall system speed. To improve the speed of the system, we adopt the distributed system. This type of system requires that both the model data and VMS should be installed at each slave PC.

The master PC comprises a user interface module and a global control module. The user interface module instructs the global control module so that the multichannel visualization system can respond to the user. The global control module supervises the synchronization of time and property by controlling the cluster of PCs.

Each slave PC comprises a local control module and a translator module, and has local information about its relative geometrical position that allows the coherent image. The local control module supervises the translator module to execute commands received from the global control module and sets the local view by combining the local position with received data. The translator module translates the commands so that the commands can be executed in the VMS.

The global control module which synchronizes the screen switching and maintains the consistency of the property information supervises the slave PCs in real time. After checking the user input data and the script data for the dynamic movement, a signal that controls the screen is sent to the slave PCs if any of the data are modified. The script data comes from the VMS and controls the model’s dynamic movement.

When the screen has been completely updated, the local control module sends the completion signal to the global control module. The global control module waits until the completion signals are received from all slave PCs. Although differences in the screen refreshing may happen due to the graphic load and the network traffic, this control module can overcome the differences. When the screen has been completely refreshed, the global control module signals the next frame.

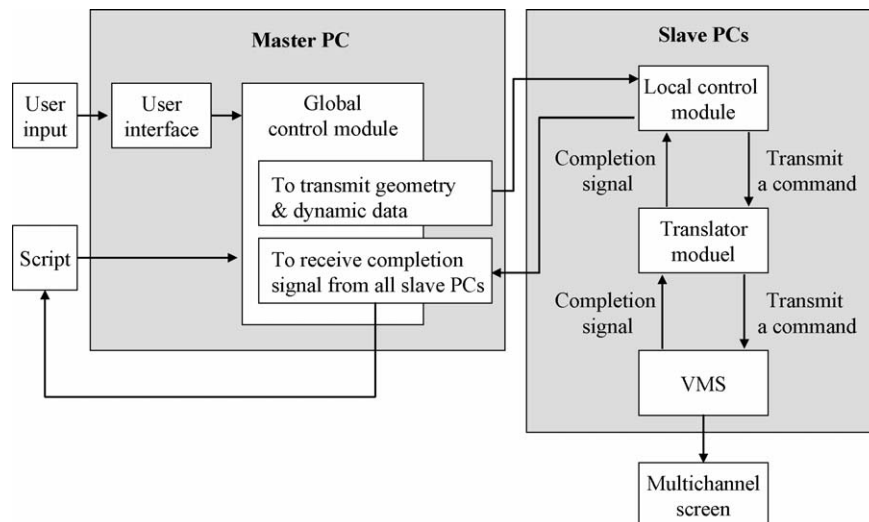


Fig. 3. Configuration of the multichannel visualization system.

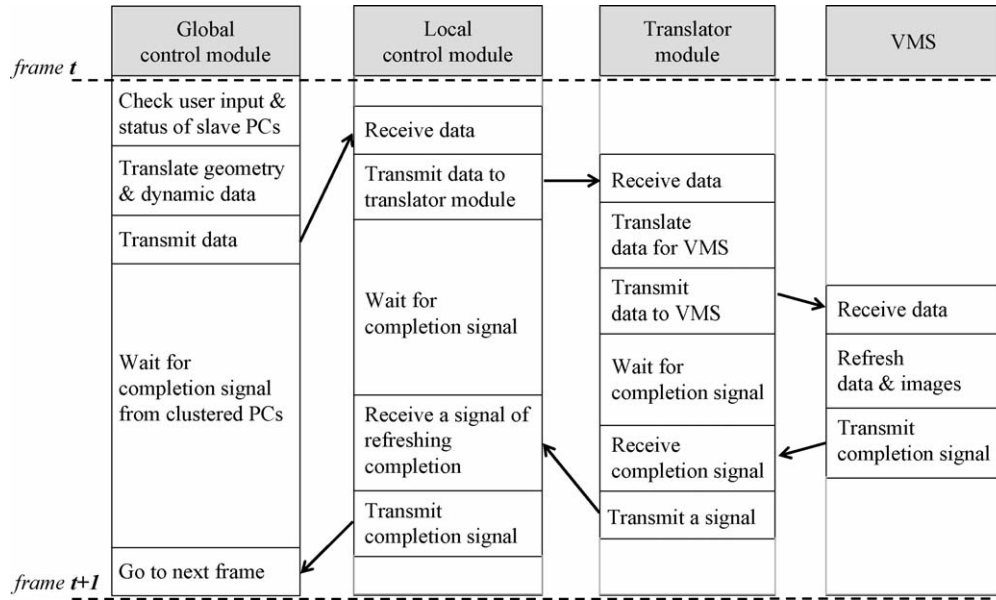


Fig. 4. Data flow between the control modules and the VMS.

Fig. 4 shows the relationship and data flow between the global control module of the master PC, the local control module and the translator module of each slave PC. The arrows between the modules show that the altered image data inside the corresponding frame, as well as the related data such as the user command, are transmitted between the modules. The unchanged parts of the image are not transmitted because the VMS of a slave PC has the pre-existing data. At the start of every frame, the global control module confirms which factors – such as the user input, previously recorded dynamic data and time – may cause an alteration, and it sends the data that require alterations to each slave PC. After receiving the data from the local control module, the translator module converts the data into appropriate commands which are then processed at the VMS.

3.2.2. Design of the translator module

The translator module converts the data received from the local control module into a command that can be processed by the VMS. The content of the converted data includes the FOV data that change with the user location; the data for starting, stopping and pausing the animated movement, the data for the screen refreshing and the data on the transmission of signals for the completion of the refreshing process. For the multichannel visualization system, we used Delmia of Dassault Systemes, a commercial VMS. To access the simulation engine of Delmia, we implemented the translator module using the command line interpreter (CLI) [19]. By externally approaching simulation

Table 3 Environment for the implementation

Operating system	Windows 2000 Professional
Global and local control modules	C++, Win32 API and MFC
Translator module	C++, CLI and Nettools
Networking	UDP/IP and Nettools
VMS	Delmia of Dassault Systemes

engines, the CLI, which is provided as the API of Delmia, can control commands related to the visualization.

4. Implementation and experiment

4.1. Implementation environment

Table 3 shows the environment in which the program is implemented. Windows 2000 is used as the operating system. The global control module and the local control modules are implemented on the basis of MS-Win32 API and Microsoft Foundation Class (MFC) is used for the user interface. The translator module that is connected to the VMS is implemented with the aid of the CLI to provide access to Delmia’s simulation engine. The Nettools program is used to control the network between the master PC and the slave PCs, and, instead of using TCP/IP, we have used the faster UDP/IP to guarantee network stability.

As shown in Fig. 5, the multichannel visualization system, which is based on a cluster of PCs, has five channels that

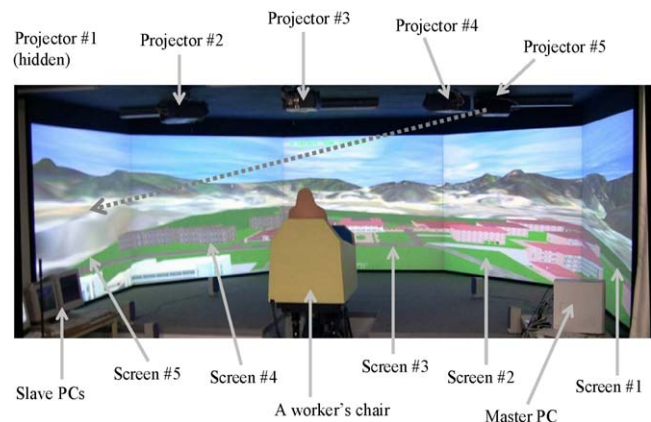


Fig. 5. Hardware arrangement of the multichannel visualization system.

provide the user with a sense of immersion. Each channel has a 120 in. screen, a slave PC and a beam projector. To improve the immersive experience, the screens surround the user. The beam projectors are hung on the ceiling to give front projections which give brighter images with less cost. Because the screen arrangement of the Reality Center™ is convenient for cooperation work in the industry, these surrounding screens provide similar experience with the Reality Center™, and provide a wider FOV than wall-style display screens. After considering the performance of the equipment, the conditions of the ambient environment and the size constraints, we arranged the system to give the user a 45° horizontal FOV and a 45° vertical FOV for each screen. In this case, the overall horizontal FOV is equivalent to the sum of the horizontal FOV for each screen.

A translator module and a local control module are installed on each slave PC. In addition, the master PC has a user interface module for the system operation and a global control module, which is used to exchange instructions with the slave PCs. For the cluster of PCs, we used general desktop PCs with a 2.4 GHz Pentium processor CPU, and they are connected to a 100 Mbps Ethernet network.

Refreshing of 20–30 frames is required for the user to naturally recognize moving pictures. The time required to refresh a single frame is less than 30 ms. According to the test results of the hardware performance, the time required to visualize a single frame on a PC with a 75 Hz refresh rate is less than 13 ms, and the network transmission time between PCs is less than 0.1 ms. We found that the multichannel visualization system can maintain its speed above the required level.

## 4.2. Implementation of the multichannel visualization modules

### 4.2.1. User interface module

The user interface module is implemented by using MFC of the Windows environment. With the aid of a keyboard, the user

sends to the user interface module the data that are necessary for adjusting an observer's location, for starting and stopping the simulation and for controlling the simulation. The user interface module then sends the data to the global control module.

### 4.2.2. Global control module

To enable the global control module to be used with Delmia or with other VMSs after small modifications, we used the Winsock of Win32 API for the network part of the global control module. Many VMSs including Delmia are blocked to outer network during the simulation. As the result, the synchronization of the screen switching is impossible due to the blocking mode of VMS. This implies that the global control module needs to control the VMS through the slave PCs using other method while the VMS is executing the simulations. To overcome this constraint, a global control module splits the whole simulation process into many short-time simulation processes. The global control module executes the short-time simulation process continuously and handle slave PCs for VR cluster between those short-time simulation processes. This interweaving process is too short for user to perceive something strange during simulation. To accomplish this process, the global control module must be able to handle two cases: one case where dynamic movement occurs and the second case where they do not move.

Fig. 6 shows the process by which dynamic and static data are handled inside the global control module. When the VMS is under simulation, the global control module looks like simultaneously handles the user's requirements for view control and the simulation data. The global control module divides whole simulation process into many short simulation processes. The global control module lets all slave nodes do short simulation process and also updates slave nodes with view control information between the short simulation processes. This fast switching or interweaving process enables the user to feel that system is interactive while in simulation. If the status is

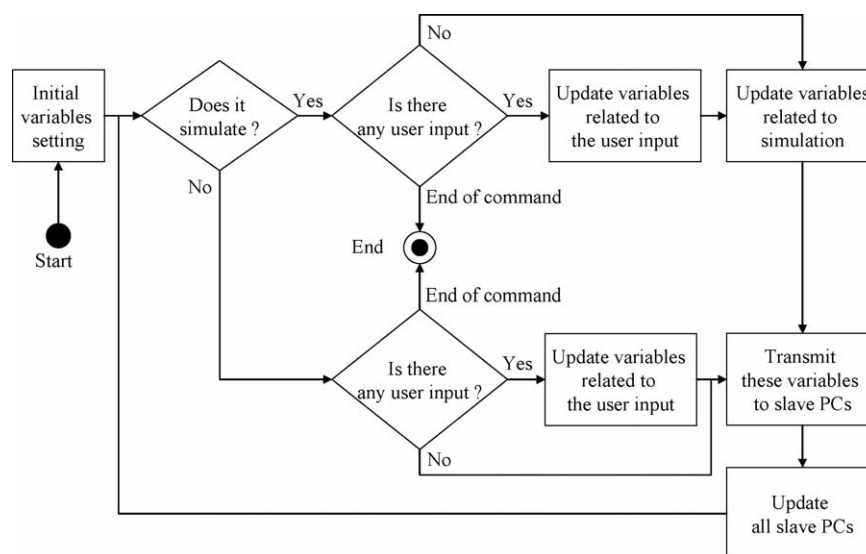


Fig. 6. Flowchart of the global control module.

static, the global control module checks for only the user's requirements for view control. The user's requirements are then reflected in the slave PCs.

#### 4.2.3. The translator module and the local control module

The translator module receives the user's requirements from the local control module, and translates them suitable for the VMS which is installed inside the slave PC. Using these converted data, the local control module is connected to the VMS. The converted data comprises the user location data and auxiliary data for additional functions. The user location data contain the direction of six degrees of freedom. This data are updated and sent to the slave PCs at every frame. The auxiliary data contain the information for file loading and resetting, and for starting, stopping and pausing the simulation. This data are sent to the slave PCs only when required.

The translator module is designed as a console mode for efficient management of system resources. The translator module depends on VMS. If VMS is replaced, we need to modify or replace only the translator module. The translator module converts the received data into suitable form for VMS. The data received from the local control module include pre-defined commands and variables for commands. The translator module converts this commands set into commands for VMS. This command is sent to Delmia via Nettools which is the network interface for Delmia, and executed in Delmia.

Some APIs in VMSs are not always sufficient for the VR environment. For some VMSs including Delmia have limited viewing function, for example, a direct access may be difficult. Combining several commands can solve this problem. For example, Delmia does not have the API command for the direct view control. This can be solved indirectly by using the camera mounting API. The indirect method inserts a special primitive into Delmia and adds the camera node to this primitive. By controlling this primitive we can get the direct view control.

The local control module is responsible for setting the initial environmental variables for the slave PCs and connecting the global control module with the translator module. Setting the initial environmental variables involves setting the global display functions that are related to the direction of the FOV; this type of setting ensures the physical arrangement that makes discrete screens appear as a single logical screen. Other initial environmental variables include the local display functions that are related to horizontal FOV of each screen. Because these settings are different at each slave PC, each local control module is not same.

#### 4.3. Experiment

Fig. 7 shows the landing gear module of airplanes on the multichannel visualization system. The data of the landing gear module includes the landing gear box, the landing gear, the landing gear support and suspension system, as well as the corresponding kinematics data. It has a simulation that shows opening the cover of the landing gear box and unfolding the landing gear. This type of simulation requires exact time sequence so that ordinary graphic browsers cannot satisfy this

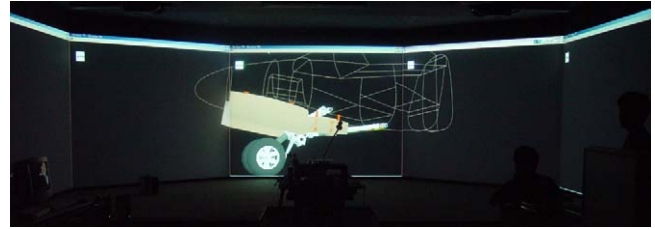


Fig. 7. Experiment 1: simulation of the landing gear module.

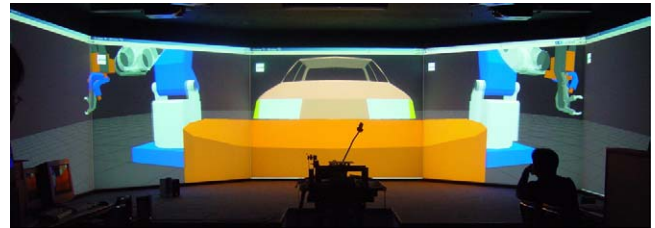


Fig. 8. Experiment 2: simulation of the spot-welding robot for BIW.

criterion. Though Delmia can be used in the multichannel system, its simulation is identical with that of Delmia at single PC.

Fig. 8 shows a part of an automotive assembly line. This simulation shows the moving station of an automotive body in white (BIW) through the assembly line with two spot-welding robots, and welding robots' jigs. Process layout designers can check the welding situation of robots through the virtual factory that provides variable-size graphic images of the assembly line, and they can analyze problems that may occur during the assembly process.

Fig. 9 is the tire attachment process in an automotive assembly line and shows the usefulness of the true size immersive VR. The worker in the form of an avatar selects and mounts a tire on a true size vehicle that enters the assembly line through the conveyor belt. By analyzing the worker's motion path and calibration by using true size, the system allows ergonomic evaluations of the work environment.

#### 4.4. Comparison with existing systems

There are two ways to visualize VMS data at immersive VR system: (1) using a graphic server that can run the VMS and

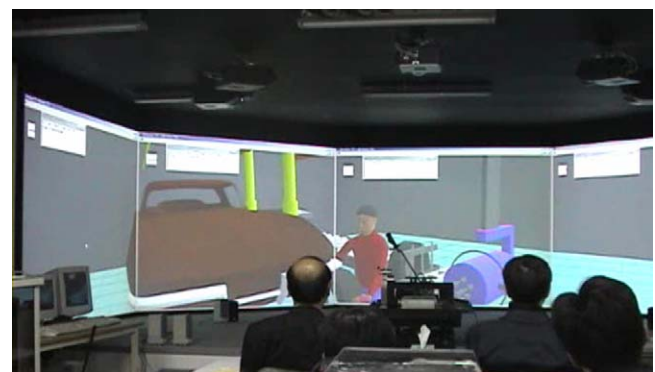


Fig. 9. Experiment 3: simulation of the tire attachment.



Table 4  
Comparison with existing systems

	Our method	SGI Onyx hardware dedicated to Delmia	VR viewers with data conversion
Hardware	Low price desktop PC	Exclusive hardware	Low price desktop PC
Construction cost	Inexpensive	Expensive	Inexpensive
Data accuracy	No need for data conversion	No need for data conversion	Causes data loss due to data conversion
Execution speed	Reduced speed due to the network	High speed with the system bus	Reduced speed due to the network
System expansion	Expandable	Expensive to expand from the initial design specifications	Expandable

immersive VR software at the same time; it is not a common case and it requires high cost for developing VMS and VR software and (2) converting the geometric and kinematics data created from a VMS into a neutral or another format that are compatible with legacy VR software. The converting processes cannot ensure the data integrity and may not be cheap or fast. The complex structure can cause flaws.

Instead, we have used the commercial VMS as the viewer of the immersive VR system. This can reduce the cost and eliminate the data integrity problem. Table 4 compares our method with the SGI Onyx hardware, high performance graphic server, dedicated to Delmia and general immersive VR software with data conversion. The Onyx hardware architecture is designed to support the multichannel display and the immersive VR environment. Even though its excellent speed, its inflexible hardware configuration does not easily meet diverse user needs.

## 5. Conclusion

We have created the immersive VR module which can be related with VMS. The module has proposed the process to use the VMS as the graphic viewer of the immersive VR system on the cluster of PCs, because using the VMS as the graphic viewer for the VR system can avoid data translation process which may make flaws in the translated data. This skipping data translation also causes to reduce the preparation time and ensure the reliability.

Because the vision is higher priority than other sensor, the realization of the proposed module was done mainly on the visualization part. Although interactive devices are keyboard and mouse only, other immersive device like 3D mouse and position tracker can be applied to the proposed module. Because Reality Center<sup>TM</sup> is better for cooperation in the industry; the realized system is similar to Reality Center<sup>TM</sup>. But, changing setting can make it used in CAVE<sup>TM</sup> type screen.

The proposed module displays shape and simulation process of manufacturing equipments exactly. This can give immersion and true size reality to special users like designers or experienced worker.

We have tested this immersive VR module with commercial VMS: the Delmia. Users can see exact same models in the single PC and can get more information according to the number of screen. Using the Delmia VMS and the proposed immersive VR module, the result of the experiment shows that the immersive VR module can give more immersion to the

users without any changes in data when it is under static state or simulation.

## References

- [1] M. Iqbal, M.S.J. Hashmi, Design and analysis of a virtual factory layout, *Journal of Materials Processing Technology* 118 (2001) 403–410.
- [2] M.B. Lari, Layout design in cellular manufacturing, *European Journal of Operational Research* 112 (1999) 258–272.
- [3] C. Shukla, M. Vazquez, F.F. Chen, Virtual manufacturing: an overview, *Computers and Industrial Engineering* 31 (1) (1996) 79–82.
- [4] M. Weyrich, P. Drews, An interactive environment for virtual manufacturing: the virtual workbench, *Computers in Industry* 38 (1999) 5–15.
- [5] B. Korves, M. Loftus, Designing an immersive virtual reality interface for layout planning, *Journal of Materials Processing Technology* 107 (2000) 425–430.
- [6] F.M. Ng, J.M. Ritchie, J.E.L. Simmons, R.G. Dewar, Designing cable harness assemblies in virtual environment, *Journal of Materials Processing Technology* 107 (2000) 37–43.
- [7] J.S. Bao, Y. Jin, M.Q. Gu, J.Q. Yan, D.Z. Ma, Immersive virtual product development, *Journal of Materials Processing Technology* 129 (2002) 592–596.
- [8] D. Paillot, F. Merienne, M. Neveu, J.P. Frachet, S. Thivent, Virtual immersive review for car design, *Stereoscopic Displays and Virtual Reality Systems* 5006 (2003) 445–452.
- [9] F. Mancini, G. Viganò, Z. Liao, M. Sacco, C.R. Boer, The virtual factory a semi-immersive interactive 3D environment, in: *Proceedings of the 2004 Summer Computer Simulation Conference*, San Jose, USA, (2004), pp. 502–506.
- [10] M. Bazargan-Lari, Layout designs in cellular manufacturing, *European Journal of Operational Research* 112 (1999) 258–272.
- [11] B. Choi, M. Park, H. Ryu, Virtual factory simulator for line prototyping, *Journal of Advanced Manufacturing Systems* 3 (1) (2004) 5–20.
- [12] S. Ressler, Q. Wang, Using Web3D technologies to visualize and analyze Caesar data, in: *Proceedings of the 15th Triennial Congress International Ergonomics Association* ([http://www.itl.nist.gov/iaui/ovrt/IEA2003\\_pix.pdf](http://www.itl.nist.gov/iaui/ovrt/IEA2003_pix.pdf)), Seoul, Korea, 2003.
- [13] E.H. Page, R. Smith, Introduction to military training simulation: a guide for discrete event simulationists, in: *Proceedings of the 30th Conference on Winter Simulation*, Washington DC, USA, (1998), pp. 53–60.
- [14] NAVERLIB <http://vdream.kist.re.kr/naver/news/>.
- [15] C. Cruz-Neira, D. Sandin, T. DeFanti, Surround-screen projection-based virtual reality: the design and implementation of the CAVE, *Computer Graphics of the ACM SIGGRAPH* 27 (1993) 135–142.
- [16] J. Kelso, S. Satterfield, L. Arsenault, R. Kriz, DIVERSE: a framework for building extensible and reconfigurable device independent virtual environments, in: *Proceedings of the IEEE Virtual Reality Conference*, Orlando, USA, (2002), pp. 183–190.
- [17] DIVERSE <http://diverse.sourceforge.net/papers/2001-whitePaper/Main.html>.
- [18] A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker, C. Cruz-Neira, VR Juggler: a virtual platform for virtual reality application development, in: *Proceedings of the IEEE Virtual Reality Conference*, Orlando, USA, (2002), pp. 89–96.

[19] Delmia Solutions: CLI Reference Manual <http://www.delmia.com/>.

[20] R. Raskar, Projector-based three dimensional graphics, Doctoral degree thesis, University of North Carolina, 2002.



**Yong-Sik Kim** is a guest researcher at the Intelligent System Division (<http://isd.mel.nist.gov/>) of National Institute of Standards and Technology (NIST, <http://www.nist.gov/>). His research interests include virtual reality (VR) for engineering design and nano-scale thermal actuators for positioner. He has a BS and a MS from the Korea Advanced Institute of Science and Technology (KAIST, <http://www.kaist.ac.kr/>).



**Jeongsam Yang** is an assistant professor in Industrial & Information Systems Engineering at Ajou University (<http://www.ajou.ac.kr/english/>). He worked at Clausthal University of Technology, Germany, in 2002 as a visiting scholar and the University of Wisconsin, Madison, USA, in 2005 as a post-doctoral associate. He obtained his PhD in mechan-

ical engineering in 2004 at KAIST. His current research interests are product data quality (PDQ), VR application in design, product data management (PDM), knowledge-based design system and STEP.



**Soonhung Han** is a professor at the Department of Mechanical Engineering (<http://me.kaist.ac.kr/>) of Korea Advanced Institute of Science and Technology (KAIST, <http://www.kaist.ac.kr/>). He is leading the Intelligent CAD laboratory (<http://icad.kaist.ac.kr/>) at KAIST, and the STEP community of Korea (<http://www.kstep.or.kr/>). His research interests include STEP (ISO standard for the exchange of product model data), virtual reality (VR) for engineering design, and knowledge-based design system. His domain of interests include automotive and shipbuilding. He has a BS and a MS from the Seoul National University of Korea, another MS from the University of Newcastle upon Tyne of UK, and a PhD from the University of Michigan of USA. He is involved in the professional societies of CAD/CAM (<http://www.cadcam.or.kr/>) and e-Business (<http://www.calsec.or.kr/>). He is an editorial member of the web-based journal, International Journal of CAD/CAM (<http://www.ijcc.org/>).