Int J Interact Des Manuf (2007) 1:107–113
DOI 10.1007/s12008-007-0013-5

ORIGINAL PAPER

A virtual container terminal simulator for the design of terminal operation

Henry Lau · Leith Chan · Rocky Wong

Received: 13 September 2006 / Revised: 10 October 2006 / Accepted: 16 October 2006 / Published online: 12 April 2007

Abstract Port container terminal operation requires a skill workforce who participates in performing a number major operation including container loading/unloading, task planning, scheduling of terminal resources, operating of crane systems, etc. in a highly coordinated manner. To facilitate the optimal design of terminal operation for both management and operators, and to provide a flexible environment for operator training and skill evaluation, a real-time distributed virtual environment that simulates container terminal operation is developed. The system includes an imseCAVE, which is a low cost fully immersive virtual reality system developed at the University of Hong Kong, together with several network computers that are linked to the imseCAVE for remote control and monitoring of terminal operations, and providing user interfaces that simulate the planning systems for terminal operations. The simulator recreates a vivid terminal operating environment that is found useful for both teaching and research in logistics and other related fields.

Keywords Interactive virtual reality · CAVE · Container handling · System design and analysis

1 Introduction

With the increased in the deployment of automated material handling systems in the ever growing logistics industry in Hong Kong, the design and analysis of these complex systems such as port container terminals and distribution centres require effective means for their evaluation. A low cost, versatile immersive virtual reality system known as the imseCAVE has been developed at the University of Hong Kong for the interactive simulation of these complex material handling systems.

The imseCAVE provides 3-D stereographic imaging, which is driven by a distributed object-oriented virtual reality engine running on a cluster of PCs connected with a high bandwidth LAN. User interfaces including wireless joysticks, tracking devices, etc. are integrated to provide ergonomic means to interact with the virtual reality models in real-time. With this system, simulators of a number of industrial systems including the quay crane simulator and the simulators for automated distribution centre for air cargo were developed with collaboration with the logistics industry. By mapping the material handling processes in association with the vivid computer graphic entities, system designers, managers and operators can rapidly evaluate their concepts prior to full implementation.

This paper describes the design and construction of a simulator for the simulation of the salient operations in a typical port container terminal using the imseCAVE. We describe the system design, architecture, modelling and the visualization of these systems with the imseCAVE.

H. Lau (✉) · L. Chan · R. Wong
Department of Industrial and Manufacturing Systems Engineering,
The University of Hong Kong, Pokfulam Road, Hong Kong, China
e-mail: hyklau@hku.hk

L. Chan
e-mail: lkychan@hku.hk

R. Wong
e-mail: rockywonghk@gmail.com

2 The design of the imseCAVE

The imseCAVE system at the Department of Industrial and Manufacturing Systems Engineering was developed based on the concept of the Cave Automatic Virtual Environment or CAVE [1]. Typically, the CAVE produces stereoscopic 3-dimensional images that are rendered from the viewer's

Fig. 1 An illustration of the imseCAVE

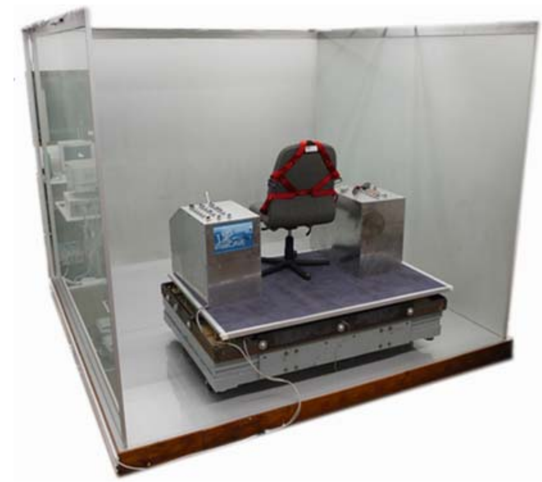

Fig. 2 The physical construction of the imseCAVE at the University of Hong Kong

perspective. A conventional CAVE uses active stereoscopic viewing devices such as shutter glasses to recreate the visual effective of stereo images and with the fully surrounding images, a fully immersive virtual environment is created.

As CAVE provides a highly versatile platform for 3-D visualization, it has been deployed in various research and development endeavours including the visualization of software objects and their relationships in complex software development [2], virtual exploration [3] and reconstruction [4] of archaeological sites, capturing of human skills for performance evaluation [5], exploration of new statistical graphics applications [6], collaborative product design and development [7], simulation of complex molecular dynamics and interactions between atomic particles [8], performing assembly planning [9], and even in the creation of artifacts [10].

Despite the desirable features and performance of the CAVE, its very high cost and somehow fixed configuration are key barriers for its deployment and ownership. In this respect, the major advantages of the imseCAVE are its low cost design, high performance and versatility in content authoring and visualization. The imseCAVE is designed based on the EVL CAVE system [1] that consists of three 10 foot by 8 foot projection walls and a 10 foot by 10 foot silver screen as the floor projection screen (Figs. 1, 2). With the use of high resolution LCD projectors instead of cathode-ray projectors, and specially designed polarizing lenses, a high performance passive stereoscopic projection system is obtained. To enable computer generated models and images to be simulated, a cluster of network PCs are used as the virtual reality engine and these PCs are linked with a dedicated high bandwidth Ethernet network.

With this PC-cluster VR engine, computer models can be efficiently simulated and distributed to different PCs for projection to corresponding screens forming a distributed virtual environment.

With this novel architecture, the imseCAVE becomes a user configurable, flexible and low-cost CAVE system that is amenable for diverse applications such as the design of automated warehouses and distribution centres, simulation study of the operation container terminal cranes systems, 3-D product development, and architectural design for urban development, etc.

While the virtual reality application software manages the animation of the distributed graphical model, the system also interfaces to external input devices that enable real-time interaction with the graphical model. In our imseCAVE, a number of joysticks, digital I/O switches and indicators, tracking devices as well as motion devices are incorporated. In particular, a motion platform is designed that can interface directly to the virtual reality software such that users of the imseCAVE can experience the motion that is associated with the virtual scenarios. The hardware architecture of the imseCAVE is shown in Fig. 3.

3 Graphical content authoring

The imseCAVE deploys an open architectured virtual reality system for producing the graphical images; the content of the virtual reality model can therefore be efficiently constructed using common 3-D modelling tools. For the imseCAVE, creating the content of the virtual world is divided into two main phases: 3-D modelling and 3-D simulation. During the 3-D modelling phase, sophisticated 3-D modelling software packages such as Autodesk© Maya© and 3D Studio MAX© can be used to create realistic 3-D objects and the surrounding environment.

Usually, in order to create a compelling 3-D scene, highly detailed 3-D models should be used. However, being

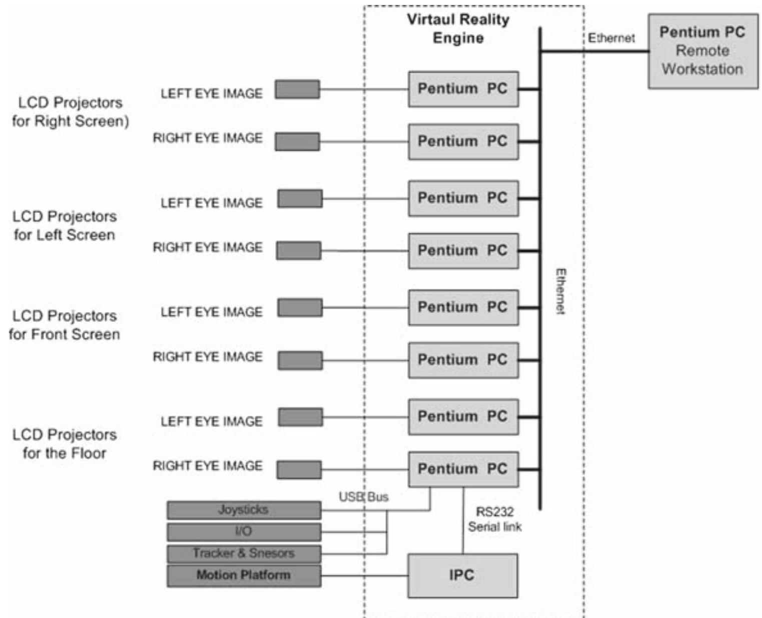

Fig. 3 The architecture of the distributed virtual reality engine and the hardware configuration of the interactive visualization system of the imseCAVE

restricted by limited 3-D computer graphics processing capability and the requirement of high frame rate during a real-time simulation, low polygon counted models with photorealistic texture mapping are more desirable.

In the phase of 3-D simulation, the virtual reality software Virtools^{TM} [11] is used to control the graphical entities of the virtual model for distributed virtual reality engine to simulate the dynamics of the system and allow user to interact with the model. In particular, behaviors are embedded into different objects so that they can behave and interact with each others. For example, collision detection prevents objects from penetrating each others and implementation of physics properties such as gravity helps to create a convincing virtual world. Moreover, the interaction between users and the virtual world has to be programmed in this stage. In order to optimize the performance, the technique of Level of Detail (LOD), that is, by displaying closer objects with more details and further objects with fewer details are also employed.

4 Architecture of the virtual container terminal simulator

One of the key to develop a multi-computer multi-screen virtual reality system is to synchronize the graphical entities that exist in the virtual world that are to be handled by different computers. As a distributed virtual environment is to be employed in the virtual container terminal simulator, a virtual reality engine that is supported by Virtools^{TM} is developed to synchronize the various computers in the system. The synchronization of computers applies the concept of distributed computing. In the following sections, an overview of the system architecture, the approach to handle virtual entities with realistic physical properties, two distribution approaches for the design of distributed virtual environment are discussed. In addition, the design and implementation of the user interface is presented.

4.1 System overview

The virtual container terminal simulator consists of two major modules. The first module simulates the working environment of quay crane control cabin. User in the control cabin controls the movement of quay crane and its spreader, that is, the device for holding the containers. Within the virtual control cabin, the operator can see the surrounding environment from within. The second module is a user interface that mimics the control tower of the container terminal and to recreate the operations that take place inside. In the control tower, operators produce the container unloading plans that define the locations of different containers in a container vessel.

As one of the key functions of the virtual container terminal simulator is to provide a realistic training platform for quay crane operators, the system is designed such that operation of the quay crane as well as the operation information available in the control tower is replicated by the system. In real operation, the control tower will instruct the quay crane operator the sequence of containers to be handled based on the loading plan for individual vessel. Such information as the loading plan is generated by the user interface of the control tower and the loading/unloading sequence will then be communicated to the virtual quay crane control cabin. Through the network of between the imseCAVE quay crane simulator and the simulator of the control tower, container handling information can be shared and communicated. In the actual operation of the entire simulator, mobile

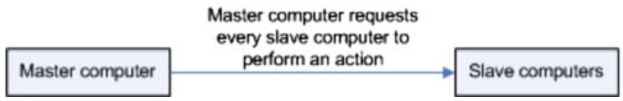

Fig. 4 Illustration of cause distribution

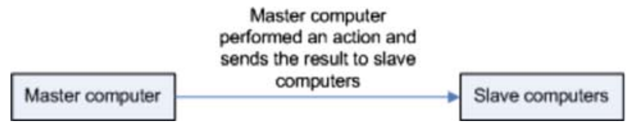

Fig. 5 Illustration of effect distribution

communication means such as walkie talkies are used by the users in the simulated control tower and the quay crane simulator to communicate such information. In doing so, a realistic container terminal working environment can be vividly recreated and users can appreciate the actual operation within a container terminal.

4.2 Approaches to handling physical entities

In the simulation of the operation of a quay crane, when a user controls the movement of 3-D entities such as the movement of the crane, the lowering of the spreader and handling of containers in a scene, interactions such as collisions between different objects will take place. Moreover, each object in the physical world has well defined physical properties including its mass, inertia, elasticity and friction, and therefore they are all governed by the laws of physics. To simulate a virtual container terminal with high realism, the movement and interactions of physical objects with corresponding physical properties have to follow the law of physics. Efficient computation of statistics and dynamics of these entities has to be performed in order to achieve this realism.

As such, approaches for the computation of these equations of motion in the virtual environment are performed to accurately simulate phenomena including the sliding action, stacking of container, collision between container and other solid objects such that their visualization can be vividly recreated. As a result, user perception can thus be greatly enhanced.

4.3 Cause distribution

Cause distribution is one of the approaches to synchronize the events and interactions between virtual entities in different computers. The master computer reads the peripheral such as trackers, keyboard and mouse, etc. states and sends the event in terms of corresponding commands to the slave computers. When all the slave computers received the commands, they will perform the associated actions requested by the master computer and these slave computers then compute their own frames according to these commands (Fig. 4).

Since every computer performs their specific actions according to the same command generated by the master computer, they are expected to behave exactly the same if no external factors are affecting their actions.

The advantage of using the cause distribution approach is the ease of the setup process. It does not require complex algorithm to keep track of the position and orientation of all the objects that are maintained in the virtual world. Moreover, system performance does not suffer even when there are a large amount of objects in the same scene as the overall system does not need to monitor their information. In the proposed simulation system, however, cause distribution is not appropriate for synchronization in our system because every computer may produce slightly different results as the virtual environment maintained by the different computers is governed by the laws of physics. Such effect of mismatch cannot be easily eliminated by using cause distribution. The differences in the virtual environment and the behavior of the entities as maintained by each computer will become more pronounce and obvious as the simulator is running for a long time. Eventually, over a longer period of time, the mismatch may become so obvious that the whole virtual environment becomes no longer acceptable. To overcome this limitation, another approach is therefore adopted.

4.4 Effect distribution

Effect distribution is the synchronization approach that has been adopted in the production of the simulation system. In contrast to the cause distribution approach, the master computer performs all the necessary computation locally as needed, and distributes the results to all the slave computers. The only job of the slave computers is to change their scene according to the results of master computer (Fig. 5).

Since there is only one computation performed for each salient event, which is performed by the master computer, all the slave computers will display exactly the same result as the master computers.

The benefit of using the effect distribution approach is to eliminate mismatch and random effect between the slave computers. The problem mentioned in the cause distribution approach does not occur because there is only one global result generated by the master computer. There will be no difference between slave computers even when the simulator is running for long time. One drawback of this approach is that it requires a large communication bandwidth between computers in order to transmit large amount of data from the master computer as the overall system needs to know about the specification represented by world matrices of each moving entities in the scene (in the current implementation, each world matrix requires 64 bytes of data in each frame). The problem becomes more significant when the number of objects increases exponentially. In addition, the implementation becomes complicated when there are a large numbers of object that are moving in and out of a scene. One solution to improve the performance of effect distribution for systems with entities that are frequently appearing and disappearing in scenes is to apply dynamic monitoring of objects. However, this option is not available in the standard version of Virtools. For Virtools^{TM}, it has the capability to transmit 1500 bytes of information in each time frame, which means that it can process 23 world matrices in a single time frame. Despite this limitation of handling 23 moving objects in the same time frame of Virtools^{TM} implementation, with a high performance master computer running at a high frame rate, the net effect for the proposed application is acceptable.

For the container terminal simulator developed, a trade off between the quality of visual simulation and the complexity of the system is made such that at any one time, a maximum number of containers to be handled is set to 64 for best use of computation resource as well as providing a flicker free display. Further optimization of system resources can be undertaken to improve the overall efficiency of the simulation system.

4.5 User interface development

In addition to the imseCAVE-based quay crane simulator module that represents the control cabin of quay crane, the other module of the container terminal simulation system is the control tower. In control tower, operators are able to view the location of the containers on a particular vessel, to generate and define a loading plan of the vessel for a particular port, to update the loading plan and to monitor the activity of the loading/unloading operation when the vessel is berth.

In the implementation of the simulation system prototype, loading plans with eight cross sections of a vessel is produced, and the operators can view the distribution of the containers through the interface (Fig. 6).

Each container on the vessel has its own identification (ID) number so that terminal operators are able to trace a particular container by searching its ID. The ID numbers are shown on the loading plan corresponding to the eight cross sections of the vessel. Through this information, operators at the control tower operators can inform quay crane operator the exact location of a container to be handled. On the right hand side of the screen of the user interface, there is a container unloading sequence list. Based on this unloading list, control tower operators then inform crane operator the exact location of the container to be handled given the sequence list. When a container is loaded or unloaded from the vessel to the truck, the loading plan of the vessel will be automatically updated

5 System trials (Experimentation)

5.1 Design of trials

In order to evaluate the performance and operation of the simulation system as a training tool for container terminal operators, the actual working procedures is mimicked using the virtual environment produced. In a typical container terminal, control tower operators use mobile telecommunication devices to communicate with the quay crane operators about the information stored in a loading plan. The quay crane operators with the assistance of the stevedores follow the control tower's instructions and handle the containers.

In the experiment, two parties are involved in participating in the container terminal simulator training trial. One group represents the quay crane operators and the stevedores, and the other group represents the operators on the control tower. Operators on the control tower shall generate the loading plan and the container handling sequence while the quay crane operators and the stevedores shall perform the actual container handling using the simulator.

5.2 Actual trial operation

When the simulator was initialized, walkie-talkies were given to the two groups of the operators that are located in different rooms in order to avoid direct communication. They could only use the walkie-talkies to communicate with each other. A loading plan and container handling sequence was generated randomly so that operators receive different scenarios during their training session. Commands were given by control tower operators and the quay crane operators gave feedback to the control tower to ensure smooth operation (Figs. 7, 8).

5.3 User feedback

Users commented that the virtual terminal environment created by the simulation system was realistic and vivid. Through the virtual container terminal simulator, users not only train to acquire skill in the controlling a quay crane, but more importantly, they appreciate the operation of a container terminal, in particular the details of the quay side operation. In addition, the importance of effective communication between the control tower and quay cranes is revealed and most users found that it is essential to develop a set of clear and unambiguous communication protocol for effective communication. In fact, efficient terminal operation not only relies on operator skills, but also effective communication between different parties. As a whole, operators found that the system provides an effective means to develop their skills in quay crane operation, a means for better understand the general operation of a container terminal, and a useful

Fig. 6 An illustration of the user interface representing the information displayed in the control tower

Fig. 7 The virtual quay crane simulation running inside the imseCAVE. Crane operators and stevedores are working together in respond to the command of the control tower to handle the containers for a berthed vessel

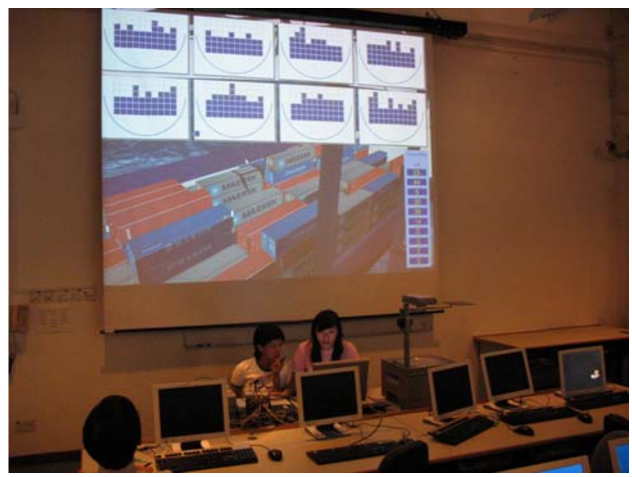

Fig. 8 Inside the virtual control room, the user interface displays the loading plan and container handling sequence that can be manipulated by the operators

platform for experimentation such as evaluating the various operating policies and resource management.

6 Conclusion

In this paper, we have described the design and architecture of the imseCAVE system as a low cost and versatile immersive virtual reality system for interactive visualization of complex systems. The system is developed based on the specification of the CAVE with associated tools for generating the graphical entities and simulating them in a distributed virtual environment. With the incorporation of specific user input devices such as joysticks and navigation devices, users can easily navigate themselves through the virtual environment and interact with the entities in real-time.

In this research, the operation of a typical port container terminal is built that includes an imseCAVE-based quay crane simulator and an user interface of the control tower. The objectives of our study are two folds, while we would like to recreate the operation of the container terminal with the imseCAVE and evaluate its performance and capability; we also used the simulation system as a means to teaching and research in logistics engineering and supply chain management. As such, we successfully demonstrated that the imseCAVE acts as an effective and user friendly system to simulate and study the container handling processes as shown by the experimental trails described in this paper.

Fig. 9 Students using the imseCAVE to gather information of container terminal operation

Through the development of the simulation system, we demonstrated that the virtual reality engine that supports the imseCAVE and other networked computers provides a flexible platform for recreating versatile virtual environment and in particular, the stereoscopic 3-D images and the immersive environment provided by the imseCAVE very much enhance the realism of the simulation.

Building from the current development, the use of the imseCAVE and its related system is extremely flexible and versatile. Currently, another of our initiatives of using the imseCAVE is to develop an immersive interactive e-learning environment (Fig. 9) for students in our Logistics Engineering and Supply Chain Management curriculum whereby students use different interfaces to navigate themselves in the virtual logistics facilities and to perform various activities that occur inside the facilities as in a typical game environment.

References

1. Sandin, D.J., DeFanti, T.A., Kenyon, R., Cruz-Neira, C., Hart, J.C.: The cave automatic virtual environment. Commun. ACM **35**(2), 64–72 (1992)
2. Ma, M., Bonyuet, D., Jaffrey, K.: 3d visualization for software development. EEE international conference on web services, pp. 708–715 (2004)
3. Laidlaw, D.A., Vote D.H., Feliz, E., Joukowsky, M.S.: Discovering petra: archaeological analysis in vr. IEEE computer graphics and applications, pp. 38–50 (2002)
4. M. Reconstructing Chavin de Huantar (Peru) Gotsis. Designing an educational tool for art historians. Seventh international conference on virtual systems and multimedia, pp. 106–112 (2001)
5. Suzuki, S., Hirana, T., Matsui, K., Okuma, Y., Kim, J.-H., Hayakawa, S., Tsuchida, N.: Capturing and modeling of driving skills under a three dimensional virtual reality system based on pwps. 29th Annual Conference of the IEEE Industrial Electronics Society (IECON '03), 1, pp. 818–823 (2001)
6. Cook, D., Kohlmeyer, B.D., Symanzik, J.S., Cruz-Neira, C.: Dynamic statistical graphics in the cave virtual reality environment. Dynamic Statistical Graphics Workshop, Sydney, Australia, pp. 1–11 (1996)
7. Bochenek, G.M., Ragusa, J.M.: Virtual collaborative design environment: a review, issues, some research, and the future. International Conference on Management of Engineering and Technology (PICMET' 01) (1), 726–735 (2001)
8. Papka, M., Stevens, R., Disz, T., Pellegrino, M.: Virtual reality visualization of parallel molecular dynamics simulation. Simulation Multi-conference Symposium, Phoenix, Arizona, pp. 483 – 487 (1995)
9. Banerjee, P., Banerjee, A., Ye, N., Dech, F.: A comparative study of assembly planning in traditional and virtual environments. IEEE Trans. Syst. Man Cybernet. **29**(4), 546–555 (1999)
10. Lim, M.Y., Aylett, R.: My virtual graffiti system. IEEE International Conference on Multimedia and Expo (ICME '04) (2), 847–850 (2004)
11. Dassault Systèmes. Virtoolstm 4. http://www.virtools.com (2006)