A PC Cluster High-Fidelity Mobile Crane Simulator

Jiung-yao Huang[1] and Chung-yun Gau[2]

[1]*Department of Computer Science and Information Engineering*
Tamkang University
Tamsui, Taipei, Taiwan 251, R.O.C.
E-mail: jhuang@mail.tku.edu.tw
[2]*Institute of Occupational Safety and Health,*
Council of Labor Affairs, Executive Yuan,
Taipei, Taiwan, R.O.C.
E-Mail: gau@mail.iosh.gov.tw

Abstract

The mobile crane simulator is a project sponsored by Employment and Vocational Training Administration, Council of Labor Affair, Executive Yuan, Taiwan, to build a safe device for training and licensing. This paper presents the principle and mechanism to build a high-fidelity interactive visual simulator on a cluster of PCs. The implemented mobile crane simulator uses the peer-to-peer architecture with the push and pull mechanism to achieve the parallelism among distributed tasks. A distributive simulation socket, called Communication Backbone(CB), is adopted to integrate the functional tasks of the mobile crane simulator in a PC clustering environment. With CB, tasks of the simulated mobile crane are executed as standalone applications and seamlessly communicate with each other through CB. Finally, the system response rate of the implemented mobile crane simulator achieves 16 times per second which is larger than human acceptable perception rate as suggested by the human factors studies.

Key Words: Virtual Reality, Interactive Visual Simulation, Cluster, Mobile Crane, Motion Platform

1. Introduction

A virtual reality system is an interactive multi-sensual environment that tightly integrates real-time 3D computer graphics, wide-angle stereoscopic display, viewer tracking, hand and gesture tracking, binaural sound, haptic feedback, and voice input/output technologies to create a realistic synthetic environment for the users. The interactive visual simulator is the most successful development of the virtual reality system. The study of the interactive visual simulator is a popular research topic in the recent years. Different types of interactive visual simulators were then designed for different purposes and ranged over various degrees of fidelities. The fidelities of an interactive visual simulation refer to the truthfulness of the sensory outputs generated to the user, which includes visual display, motion tracking, sound render and haptic feedback.

Depending upon the applications, the interactive visual simulator may require to provide different degrees of fidelities for the simulated scenario. Furthermore, depending upon the variance of the sensuousness provided by the system, the user perceives different degrees of immersion and, hence, the system requires a distinct computing environment. For example, a high-fidelity virtual reality system, such as a flight simulator or driving simulator, is capable of

revealing all of the physics phenomena of the simulated entity and displaying the virtual image at the system response rate around 60 Hertz for one full pass for all software modules [19]. When a user pushes the pedal, the fight simulator must recalculate the new position of the airplane according to its current position, velocity, acceleration, altitude, wind speed, and gravity. With the increasing demands of complexity and realism of the simulated scenario, the high performance multiprocessor mainframe system, such as IBM AIX RS/6000 [19] or two 12 processors SGI rack Onyxes with two IR2 graphics pipes [14], is often used as the computing environment for the interactive visual simulator.

With the advancement of the silicon technology, the desktop computer has gained more computing power with less cost in the recent years. By carefully exploring the parallelism among the tasks of a virtual reality system, we can easily cluster several computers by the local area network and employ the pipeline technique to design a virtual parallel computing environment [18]. Hence, recent researches on the interactive visual simulation incorporate personal computers to construct their computing environments.

The paper is to present an architecture and mechanism to design a mobile crane simulator on a cluster of PCs. Although Yoneda [25] presented a similar research on crane operation, their work concentrated on designing an operational assistance system to develop a control rule to assist straight line operation. Hence, the entire computing system contains an I/O PC to receive the user input and a SGI workstation to compute the cable dynamics and render image. Different from their work, the system presented in this paper attempts to build a high fidelity mobile crane simulator on networked PCs. That is, the components of the presented mobile crane simulator include a realistic crane mockup, a 6-dof motion platform, and a surrounded view by three monitors. For the rest of this paper, the overview of the related works is given first. The discussion of the mobile crane is presented next. The analysis and design of modules of the mobile crane simulator then follow. The clustering computing technique to integrate functional modules into a mobile crane simulator is then presented. Finally, the implementation of the mobile crane simulator comes to last and followed by the conclusion and future work.

2. Related Researches

The flight simulation is probably the most representative application of the interactive visual simulator. The flight simulator by CAE Electronics Inc [19]. was built with a complex and effective six-degree-of-freedom motion system to generate the realistic feeling of takeoff, landing and in-flight turbulence. In addition, a realistic mockup is mounted on top of the motion system to immerse the user into the training scenario. In order to control the computation of the flight dynamics and generate a realistic motion in real-time, the multiprocessor IBM AIX RS/6000 is used as the mainframe system. On the other hand, the SE-based flight simulator at the Virtual Reality Applications Center(VRAC) [14] uses the C2, which is a room of four projected surfaces, to create a fix-based airplane and helicopter simulator. To generate images on C2, the SE flight simulator uses two 12-processor SGI rack Onyxes with two IR2 graphics pipes as its mainframe system.

The drive simulator is another representative application of the interactive visual simulator. The Ford company building a driving simulator [7] with an ESIG-2000 computer from Evans & Sutherland as its image generator, a four-processor DN-10000 computer from Apollo to compute the motor dynamics, and a two-processor real-time I/O computer from Harris NightHwak to control the data flow inside the simulator. The IOWA drive simulator [10] also uses an ESIG-2000 computer to generate its simulated image and three real-time multiprocessor systems to simulate the vehicle dynamics, control complex scenario, and manipulate various I/O peripherals. Meanwhile, the PNU vehicle driving simulator [16] uses a dynamics workstation as its server to receive user input from an I/O PC, simulate vehicle motion, and then pass the result to SGI Indigo2 Impact R4000 to render the image and to the controller PC to control the motion platform. On the other hand, presented an approach to design a driving simulator with four PCs only [12].

The Virtual Environment Technology for Training(VETT) [26] program designs a simulator of less fidelities to train the officer of the deck(OOD) on a submarine. Without a sophisticated mockup, the OOD simulation aims to train the officer navigating a submarine through a channel. The OOD uses SGI dual-processor Onyx workstation with Reality Engine^2 graphics subsystem to support the complex training scenario in real-time.

Another researches on applying interactive visual simulation include the power system operator trainer [21], the bicycle simulator [2,11], the motorcycle simulator [6], and the rough crane simulator [25].

ESOPE-VR [21] is a VR operator training simulator prototype for power-utility personnel. Its functionality includes 3-D visual interface, voice recognition and feedback, navigation and manipulation facilities, and expert system, multimedia and multi-user. To support these functionalities, ESOPE-VR uses two SGI Indigo workstations to render the 3D scene, and audio and video I/O, respectively. In addition, a tutorial and decision making system for power system operators, called ESOPE (Expert-System for Operations Environment), is executed on a PC as the backbone of the entire simulation system. Furthermore, another PC is dedicated for speech recognition function to support vocal command.

The Peloton bicycler simulator [2] is a sport simulator that uses the Virtual Reality Modeling Language (VRML) browser as its user interface. It also provides a multi-user virtual environment over the Web architecture so that the user can tour or race with other remote players. The peloton uses a single computer to manage the simulation and render the image. In addition, two proprietary microprocessors are connected to the computer through serial ports to control two respective sensual devices. One is to manipulate the pedaling resistance to simulate the terrain changed, and the other is to control the fan speed to enhance the sense of motion.

The KAIST interactive bicycle simulator [11] is a high fidelity version of bicycle simulator designed by Korea Advanced Institute of Science and Technology. The components of this bicycle simulator include a Stewart platform to generate 6-dof motions, handle and pedal resistance systems to provide force feedback, the visual simulating system to create virtual scene. Its computing environment is composed of three PCs in server/client architecture. The server PC is responsible for receiving the handle and pedal inputs from the bicycle, and computes the bicycle dynamics to trigger the resistance system on the simulator and to control the motion platform master by a client PC. The server PC then forwards the computed result to the render PC to generate a realistic virtual campus.

The MORIS (Motorcycle Rider Simulator) [6] is the study of designing a prototyping and testing tool for the motorcycle manufacturer. Its computing backbone was constructed by a Alpha workstation as the server, a SGI workstation to render the image, another Alpha workstation as Mechanical Subsystem (MES) to control the motion platform, and a PC to generate audio sound.

3. Modeling the Mobile Crane

The mobile crane, or rough terrain crane [25], is a common apparatus in the construction site or factory. Basically, a mobile crane is a hoisting machine mounted a truck or a caterpillar. Its lift mechanism is composed of a hydraulic propelled hanging bracket and winches. The hanging bracket is a multi-section assembly boom with a plumb cable and a hanger on the end. The operation of the hanging bracket is a complex and difficult task. It requires sufficient skills to accurately manipulate the swing of payload to safely and efficiently lift a heavy object. Some studies were conducted to reduce the operating load of the crane system. These studies can be categorized into two types [25]: automatic (or semi-automatic) control system and operational assistance system. The operational assistance system is the study of presenting various types of information to assist the user to perform safe and efficient operation. The work presented in this paper aims to design another type of operational assistance system to train the operator for safer operation.

Furthermore, since the mobile crane is an oversized hoist machine with a multi-section assembly boom, its center-of-gravity position can be easily shifted while driving. This situation often causes the crane to turn over. Hence, the driving simulation of the mobile crane is also an important training item to the operator. To simulate the hazard situation of overturn, a motion platform is required when the driving simulation of the mobile crane simulator is designed.

Due to the hazardous nature of the mobile crane, Employment and Occupational Training Administration, Council of Labor Affair, Executive Yuan, Taiwan, launched a three-year project in 1997 to build a training simulator for the mobile crane. The purpose of this project is to reduce the possibility of the occupational disaster by providing a safe and controllable training environment to train the operator to efficiently manipulate the crane. Based upon the previous studies of the mobile crane, the mobile crane simulator for this project is composed of two parts: the driving simulation and operational simulation.

The operational simulation is an operational assistance system [25], which is to train the operation of manipulating the crane safely. Since the goal of this project is to design a mobile crane simulator for training as well as licencing, the realism of the simulation is a very crucial issue. When operating a crane, the operator relies on three types of information, which are vision, sound, and the dashboard's meters and lights, to control the derrick boom and the plumb cable. Hence, in order to create a realistic simulating environment for the trainee, a mockup equipped with a dashboard from the actual crane must be provided. Since the operational panel of a mobile crane includes two joysticks to control the derrick boom and the plumb cable, and a dashboard of twenty-six indicators, ten meters and eighteen switches, the designed mockup must fully duplicate all of these input and output instruments for the trainee to get fully immersed into the simulated scenario.

Furthermore, the mobile crane simulator must also produce sufficient sensor cues to make the user believe that he is actually inside the synthetic environment. As pointed out in [14], these sensor

cues may include a large field-of-view real-time stereoscopic visual display, surrounded sound digital audio, and haptic interaction. Hence, in order to further provide sufficient sensory cues to the trainee, a surrounding screen of three displays is mounted inside the mockup to emulate the out-of-window views from the crane cabin. The surrounding screen provides wider field-of-view to convince the user that he is actually manipulating a mobile crane.

Since driving the mobile crane is a testing item for licencing, a driving simulation must also be integrated into the mobile crane simulator. A six-axis motion platform is then introduced to realistically simulate the hazard situation of driving. Hence, as depicted in Figure 1, seven modules were identified to design the mobile crane simulator. These seven modules are Cabin module, Motion platform controller, Instructor monitor, Scenario module, Dynamics model, Visual display and Audio module. These seven modules are distributed among networked computers and coupled with each other to form a cluster of real-time computing environment.
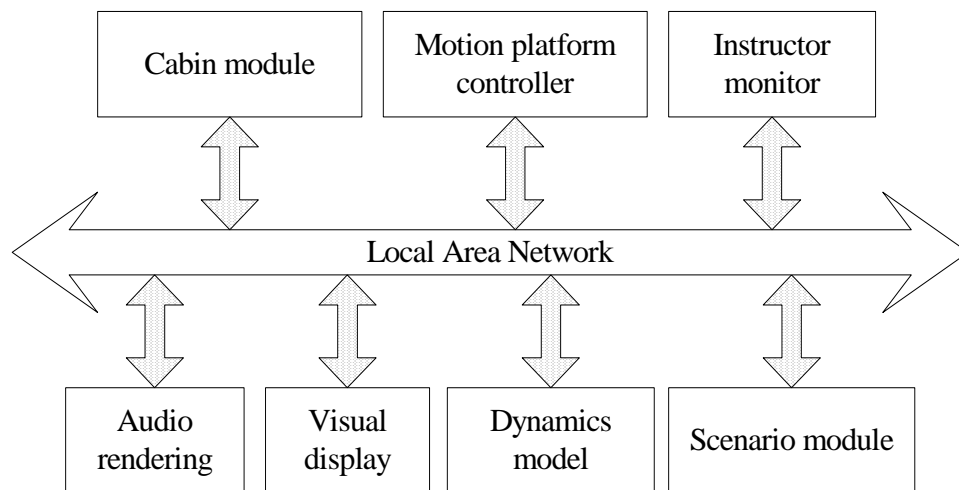

Figure 1. The infrastructure of the mobile crane simulator

The cabin module is an I/O module that receives signals from the input on the dashboard of the mockup and triggers the indicators and meters on the dashboard. The motion platform controller is an output module that receives messages to change the posture of the motion platform. The instructor monitor is another I/O module that enables the instructor to send commands to trigger the meters and indicators of the dashboard and

receive messages from the dashboard while the trainee is controlling the derrick boom and the plumb cable. The audio rendering module is an output module that generates various background sounds to immerse the trainee. The visual display is an output module that controls the display images of the surrounding screens. The dynamics model module is responsible for computing the physical phenomena and posture of the simulated

crane. Finally, the scenario module controls the flow of the script during the training scenario. The detail descriptions of these seven modules are given in the following section.

4. Design of the Mobile Crane Simulator

4.1 The Cabin Module

The cabin module is a part of the I/O device for the simulator. There are two categories of input/output devices on the mobile crane simulator. One is called sensory devices which are composed of a surround display of three monitors, digital audio, and a motion platform controller. The other is a mockup of the mobile crane for the trainee to be familiarized with the controlling instruments. Such a mockup also includes a realistic duplication of the dashboard of the actual mobile crane. As shown in Figure 2, the instruments on the dashboard contain various indicators, meters, steering wheel, gas pedal and brake. The cabin module is a program that monitors the signal of each instrument on the operational panel and translates the received signal into the message to be passed to other modules. In addition, the dashboard module receives messages from the Instructor monitor, which will be fully discussed in the next section, to drive the meters and indicators on the dashboard.

Figure 2. The dashboard inside the mockup

Specifically, the input devices embedded into the mockup include a steering wheel, brake and gas pedals, and two joysticks to manipulate the derrick boom and the plumb cable of the mobile

crane. Furthermore, there are two categories of input devices on the dashboard, which are indicators and meters, and various switches as the input device. Twenty-six indicators are provided on the dashboard which include direction indicators, oil contamination, and suspension luck etc. Ten meters are provided on the dashboard, which include speed odometer, directional meter, and oil gauge etc. The cabin module uses the industrial Analog/Digital board to send and receive signals to and from the dashboard inside the mockup.

4.2 The Instructor Monitor

Since the mobile crane simulator is designed to be the training vehicle, an instructor monitor module is an important interface for the instructor to monitor the operation of the trainee. Two monitoring windows are designed for the instructor to supervise the trainee. The first one is called the Status window, as illustrated in Figure 3, which is a two-dimensional display to the status of the simulated mobile crane.
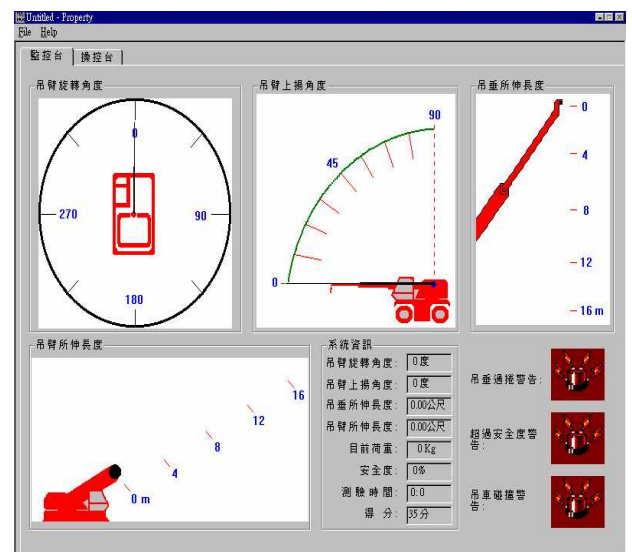

Figure 3. The status window for the derrick boom

The top-left window of Figure 3 displays the current swinging angle of the derrick boom. Since the derrick boom will shift the gravity of the mobile crane when it is swinging, it is an important safety factor to monitor the swing angle of the derrick boom. The top-center window depicts the raising degrees of the derrick boom. Similar to the swinging angle, the raising angle will also cause hazard situation if the angle is over some safety value. The top-right window shows the current length of the plumb cable, and

the left-bottom window displays the elongate length of the derrick boom. In addition, the pictorial information on each window is also be digitalized and displayed on the small dialogue boxes next to the bottom-middle area. For the sake of training, alarm signals are added on the right-bottom area to signal the misconduct of the operator if it occurs. For example, if the derrick boom overshoots the safety zone, the second alarm will be lighted along with a warning sound to alert the possible danger. Hence, the status window on Figure 3 can significantly assist the instructor to administrate the trainee if he is safely manipulating the simulated mobile crane. Moreover, the scores calculated by the scenario module will also be displayed on the status window.

The second monitoring window is the pictorial duplication of the instruments on the dashboard, called the Dashboard window, as shown in Figure 4. The dashboard window is the complete duplication of the dashboard inside the mockup. It aims to allow the instructor to oversee the training procedure of the trainee. In addition, since the cabin module can send signals to trigger the dashboard, the dashboard window can be used by the instructor to train the user on trouble shooting. For example, the instructor can click to snuff off the oil light to signal the empty of the fuel tank during the operation and to observe the reaction of the trainee.

Figure 4. The dashboard window

4.3 Motion Platform Controller

The Motion platform controller is the

module to manipulate the six-manipulator motion platform. The motion platform is another important sensory device for the trainee to fully immerse himself into the training scenario. The six-manipulator motion platform is able to deliver six degree-of-freedom postures to the trainee, including rotations and shifts with respect to X-, Y- and Z-axis. With this six-manipulator motion platform, the system can simulate the situation of acceleration, vibration, and turn over etc.

The six-manipulator motion platform was first presented by Stewart [13] in 1965 and called Stewart Platform Based Manipulator thereafter. As shown in Figure 5, The Stewart Platform Based Manipulator is a parallel manipulator consisting of a fixed base plate and an upper moving plate, which are connected together by six adjustable links with ball joints at each end. These six manipulators can be expanded and contracted individually to control the gesture of the platform.
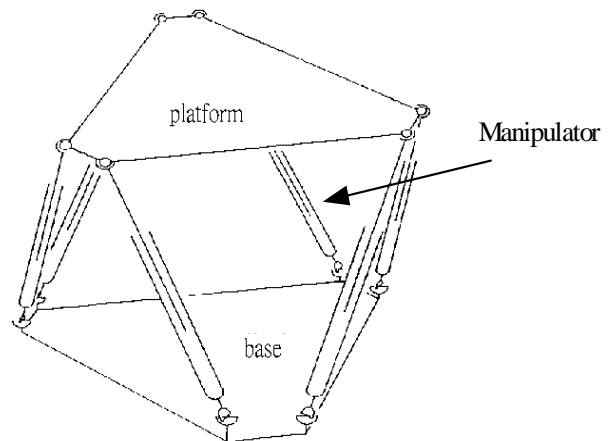

Figure 5. The Stewart Platform Based Manipulator

In order to realistically simulate the pose of the mobile crane inside the virtual environment, the Motion platform controller must smoothly transform the posture of the platform between two consecutive statuses. To achieve this goal, the controller program must preserve the state value of each posture and interpolate motion between two continuous postures. In addition, the frequency of this interpolation should be synchronized with the visual display in order not to disorder the sensorium of the user. Otherwise, for example, the user may visually see the mobile crane going downhill while the motion platform is still in an uphill posture. Hence, the frame rate of the visual display is required to be used as the parameter for the interpolation.

Finally, the vibration of the motion platform is also another important factor to realistically simulate the mobile crane. Since the lower body of the mobile crane is a truck or a caterpillar, it will create constant noise and vibration while its engine is ignited. Hence, the motion platform controller constantly generates a random up-and-down vibration to realistically simulate this condition.

4.4 Scenario Control Module

The scenario control module is designed to manage objects within the virtual world and, hence, to evaluate the performance of the trainee. Since the purpose of this mobile crane simulator project is to design a vehicle for training as well as licencing, a scenario is required to evaluate the trainee. As shown in Figure 6, the designed scenario includes driving the mobile crane from the starting point to a designated location and then manipulating the derrick boom. In order to increase the difficulty of training, the scenario requires the trainee to drive the mobile crane through the sand land and hills.
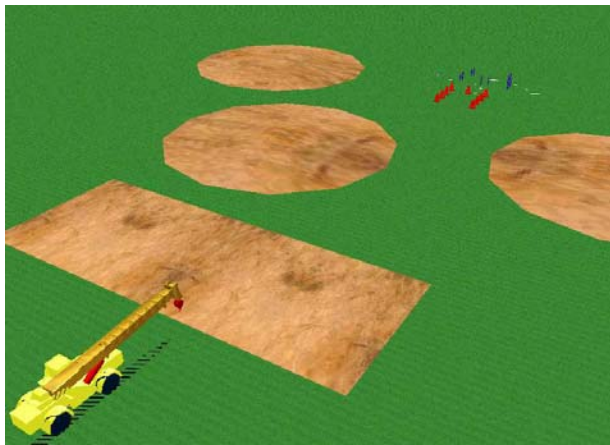

Figure 6. Bird eye view of the training scene

After arriving at the examinatorial spot, the trainee is asked to control the derrick boom to lift a cargo and move that cargo according to a specific trajectory. As shown in Figure 7, bars are placed on the path of this trajectory to obstruct the movement of that cargo. The trainee is asked to lift the cargo located in the white circular zone at the left side of Figure 7, to move the cargo to its right side along the trajectory and then back to its original position. Scores will be deducted if the bar is collided and the scores will be dynamically displayed on the status window that is shown in Figure 3.
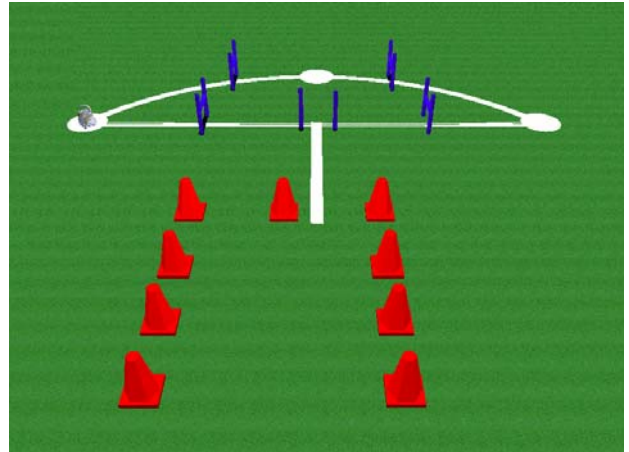

Figure 7. The trajectory of licensing exam

4.5 The Dynamics Module

The dynamics module increases the realism of simulation by calculating various physical phenomena. The dynamics computed by the mobile crane simulator includes the inertia oscillation of the cable and the lift hook, the collision detection and the terrain following.

When the mobile crane and its lift hook are moved in the virtual environment, the dynamics computation uses the multi-level collision detection algorithm [15] to effectively perceive the collision if there is any. The dynamics module sets up a bounding box and a bounding sphere for each object, and an object may have a hierarchy of bounding boxes if it is composed of a hierarchy of sub-objects. The bounding sphere is the first criterion for the collision detection. If a collision is detected among the bounding spheres, then the bounding box collision detection follows for further examination. If a collision is detected, the dynamics module first animates this collision event and then sends messages to the Audio rendering module and the Visual display module to playback a collision sound and render this visual event, respectively.

In addition, in order to realistically simulate the mobile crane, the dynamics module also computes the inertia oscillation of the plumb cable when the derrick boom is moved. When the derrick boom is moving, the dynamics module computes the inertia of the left hook acts on the cable based upon the moving direction, speed and weight of the cargo. When the derrick boom is stopped after moving, the same computation of the inertia will be repeated and the cable is oscillated until a full stop. This computed information is passed to the Visual display module to realistically render the physical phenomenon inside the virtual scene.

However, constrained by the availability of the computation power, the cable is assumed as an inflexible wire to simplify the computation.

The terrain following is another important factor for the realism of simulation. Since its center of gravity is higher than that of other types of vehicle, driving the mobile crane is also a dangerous process. The terrain following mechanism along with the motion platform provides a method to train the operator to drive the mobile crane safely.

4.6 The Visual Display and Audio Rendering Modules

The visual display and audio rendering modules are another two important sensory devices for the user to be fully immersed inside the virtual environment. The visual display module is implemented through the Microsoft Direct 3D library. When a scene file is read, the Direct 3D library will automatically organize all objects within the scene into a tree structure [4]. Unfortunately, this scene tree is an exclusive data structure and it is difficult for an application program to control an individual object in the tree structure. In order to integrate the visual display module with the dynamics module, an object table is designed to interface the dynamic module and the Direct 3D's scene tree. In addition, as shown in Figure 8, three graphical PCs are used to render the left, center and right views of the mobile crane to provide a 120-degree surround view. This surround view system is fully synchronized with each other so that a consistent view will be displayed [9]. With this surround view system, the trainee can fully immerse himself into the training scenario.

Figure 8. Three synchronized monitors to provide a surrounded view

The audio rendering module is responsible for producing the static sound, such as the background noise, as well as the dynamic sound effect, such as collision sound or motor working noise. For a virtual reality system, the sound effects along with the realistic image are two important ingredients for the user to be fully immersed in the synthetic environment. The Microsoft DirectSound library is used to implement the sound module. With this Audio rendering module, a realistic training scenario can be provided to the trainee.

5. Integration with a Cluster of PCs

After modules of the mobile crane simulator are developed, an integration system to communicate among these modules is developed. The system integration technology is an important factor that will significantly affect the fidelity and performance of the designed simulator [16]. Due to the evolution of the computing technology, many of the recent researches on the interactive visual simulation either entirely use PC(s) [11,21] or adopt PCs to construct a heterogeneous system [6,16] as the computing environment. The traditional approach is to use the server/cluster architecture with the server to compute the dynamics of the simulated entity and coordinate the data among the clients [11,16]. One potential drawback of this approach is that the server becomes a bottleneck when a high-fidelity simulation is performed and high volume of data is communicated among the clients. To overcome this problem, this paper presents a peer-to-peer architecture on a cluster of PCs to design the computing environment for the mobile crane simulator. To achieve this goal, tasks of a virtual reality system application are treated as physical processes of various capabilities and are distributed among a cluster of computers. The communication among distributed tasks is then transmitted through the message passing methodology or the shared memory mechanism. In this way, the sequence of the tasks of an interactive visual simulation can be distributed among networked computers and pipelined execution to accelerate the entire simulation process. In the following sub-sections, the principle of designing a peer-to-peer architecture on a PCs cluster is given first. The model and mechanism of the presented architecture then follow. The integration of a mobile crane simulator with the presented mechanism is discussed at the last.

5.1 Principle of Peer-to-Peer Architecture

The basic concept of designing a peer-to-peer architecture is to allow each PC to automatically communicate with other PCs in the cluster. Since there is no server to coordinate the message flow among distributed tasks on PCs, the asynchronous message passing method [3] is the most popular approach to design the required communication mechanism. As surveyed by Fadlallah [5], the message passing is the surest way to achieve the high performance for the majority of distributed applications.

The principle of designing a peer-to-peer interactive visual simulating environment on a cluster of PCs (COP) is that each process on the COP runs as a standalone program to construct a parallel computing environment. As listed in [5], four essential attributes for the parallel computing are concurrency, scalability, locality, and modularity. Two de facto standards that satisfy these attributes to construct a parallel computing environment over a cluster of computing hosts are PVM [17] and MPI [24]. However, since neither PVM nor MPI are designed for the interactive computing environment, a transparent communication layer, called Communication Backbone(CB) [8], is adopted to provide a seamless communication environment over a cluster of PCs.

With the help of CB, each process of COP only needs to convey its event to CB without knowing the existence of other processes. Hence, under the COP architecture, each computer can be executed at its own pace and parallelisms among distributed tasks are then automatically explored. The COP architecture can be depicted in Figure 9. Each computer in this COP environment executes CB as its communication layer, and Logical Processes (LPs) run on different computers communicating with each other via CB.
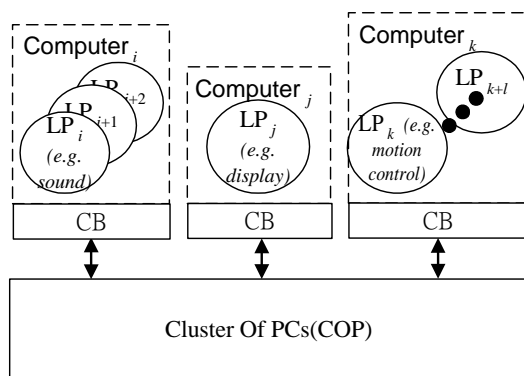
Each LP of COP run on top of CB does not have to be concerned about the existence of other LPs. One or many LPs can run on a computer, depending upon the computational load of each LP. An LP only needs to register to its resident CB upon its execution by issuing supported service calls. The CB will then schedule the message flow among distributed tasks, no matter that the corresponded LP is in the same machine or across network. With this capability, a heterogeneous computing environment can be constructed, and different tasks can be easily plugged to form different types of simulation environment.

5.2 The Push and Pull Model

To satisfy the essential attributes of the parallel computing [5], the push and pull model [20] is used by COP. That is, each LP is a modular task of the simulator and it is locally executed as a standalone program. From the point of view of Logical Process (LP), CB plays the role of agency of the COP environment. Each LP only needs to register to its resident CB on what type of data that it is going to produce. The CB will treat this LP as one of the publishers of this distributed environment. Similarly, an LP may also need to inform CB of what kind of information that it requires and CB will treat it as one of the subscribers. An LP can both be the subscriber and publisher at the same time. Hence, during the initialization phase, CB will be responsible for matching publishers with their corresponded subscribers to interconnect the registered LPs to create this distributed environment. A virtual channel is then formed between each pair of the publisher and subscriber when the initialization phase is completed. Notice that, in order to enable a new LP to dynamically join an existing simulation, the initialization phase will be repeatedly executed at a predefined intervals in the duration of simulation.

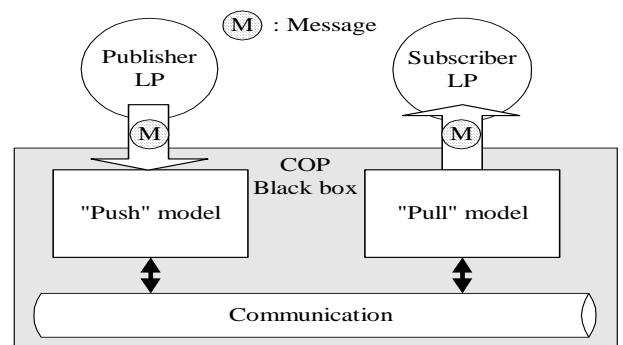Figure 9. The infrastructure of the COP environment
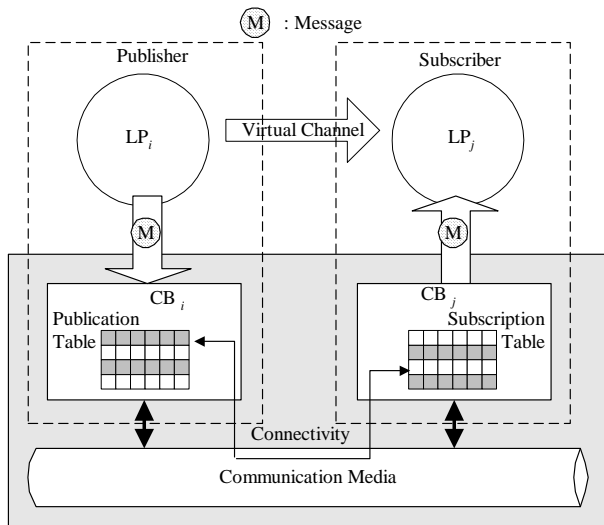

Figure 10. The push and pull model for the COP

Figure 11. The virtual channel between LPs

During the run-time phase, as illustrated in Figure 10, the publisher will treat COP as "push" model and "pump" its data to COP. On the other hand, the subscriber treats COP as the "pull" model that it can dig information out of COP. With this model, a transparent communication among LPs can be easily designed and constructed.

The virtual channel mechanism is an important notion for COP. Conceptually, virtual channels are the pipelines that seamlessly interconnect LPs to form a simulating environment. Physically, as shown in Figure 11, a virtual channel is an entry mapping between CBs. That is, after an LP registers to CB as a publisher or subscriber, CB will respectively record the LP's information in its Publication table or Subscriptiontable. During the initialization phase, when a publisher is matched with a subscriber, an entry of its Publication table will be "linked" to the corresponding entry of Subscription table of that subscriber.

5.3 The Integration of the Simulator

The major benefit of the presented peer-to-peer architecture is to develop a high fidelity mobile crane simulator on a cluster of PCs. Furthermore, with the help of the push and pull model, the functional modules of the simulator can be dynamically added to form a cluster of computing environment. Furthermore, this cluster environment belongs to a loosely-coupled architecture [22]. Figure 12 illustrates the infrastructure of the designed mobile crane simulator constructed by the push and pull model.
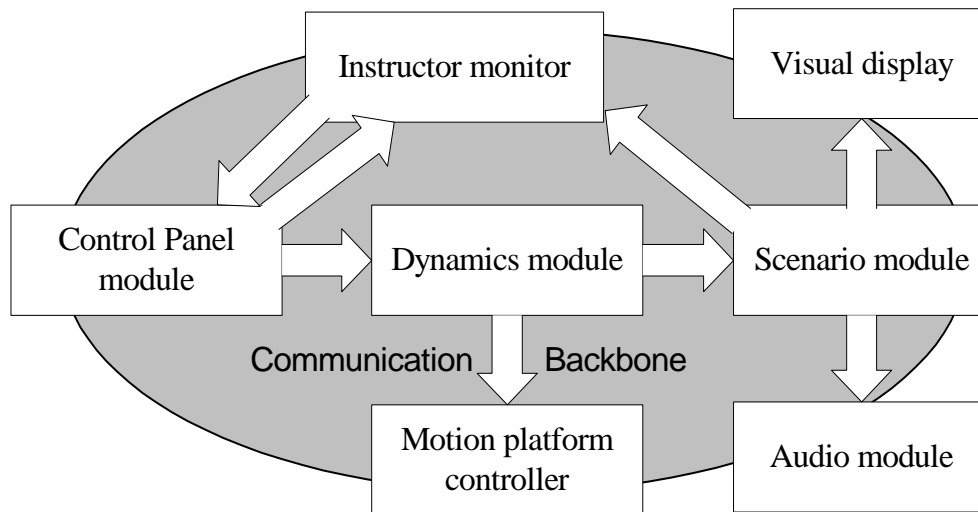

Figure 12. The infrastructure of the mobile crane simulator

For the designed mobile crane simulator, the publication and subscription relationship among modules are as follows. The control panel module is responsible for publishing the control signal from the dashboard and driving instruments. The dynamics module subscribes the user's inputs and publishes the computed result for further processing. Since a scenario module is an import function to control the progress of the training scenario, it subscribes the dynamic state of the simulated crane to compute its status within the simulated scenario. The scenario module then publishes the computed scene in terms of audio message and visual information, which are then subscribed by the audio rendering module and visual display module, respectively. The motion

platform controller also subscribes the dynamic state of the simulated crane to change the posture of the mockup. Finally, since the mobile crane simulator requires an instructor to monitor the training process, the instructor monitor module subscribes the user's inputs to reproduce the dashboard status inside the cabin. In addition, it also needs to subscribe the simulated status to perform evaluation of the training scenario. Significantly, the instructor monitor can also publish the command initiated from the instructor to change the scenario. This function enables the instructor to train the operator on handling the unexpected hazard situation. At current implementation, the control panel module subscribes these commands which enable the instructor not only to monitor the dashboard of the simulated mobile crane but also to control its switches and lights.

6. Implementation and Result

The presented mobile crane simulator is implemented on the Microsoft Windows platform. These seven modules, which are I/O control module, instructor monitor, scenario control module, dynamics module, visual display and audio rendering modules, are distributed among rack of PCs as shown in Figure 13. These PCs are connected by 100Mb Ethernet network to form a distributed interactive computing environment.

Figure 13. Rack-mounted computers to form a COP

The top three PCs are the graphical computers that control the displays of the three monitors inside the mockup to create a surrounded view. Each graphical PC is equipped with a TNT2 M64 3D graphic acceleration card. The fourth PC from the top is the synchronization server that synchronizes the frame rate of the above three graphical PCs to generate the surrounded view [9]. The fifth PC is the I/O computer that executes Cabin module to communicate with the control panel in the mockup. The last two PCs execute the instructor monitor and motion platform controller modules, respectively. Figure 14 shows the appearance of the mobile crane simulator.

Figure 14. The mobile crane training system

Figure 15 is the snapshot of the train scenario, and the virtual scene contains 3235 polygons that are rendered at 1024x1024 pixels with 24 bits. The simulation lag [23] of the implemented mobile crane simulator is 0.0625 second, which is equivalent to 16 times per second of the system response rate. This simulation lag includes the communication delay among cluster of PCs, I/O delay from the control panel, rendering delay, simulation delay, synchronization delay among three graphical PCs, motion platform control delay,

and frame-rate-induced delay. Since the human factor studies indicate that people begin noticing latencies when they exceed 100 milliseconds [1], our mobile crane simulator is around 62 milliseconds and is proved to be a successful high fidelity interactive visual simulating system on a cluster of PCs.
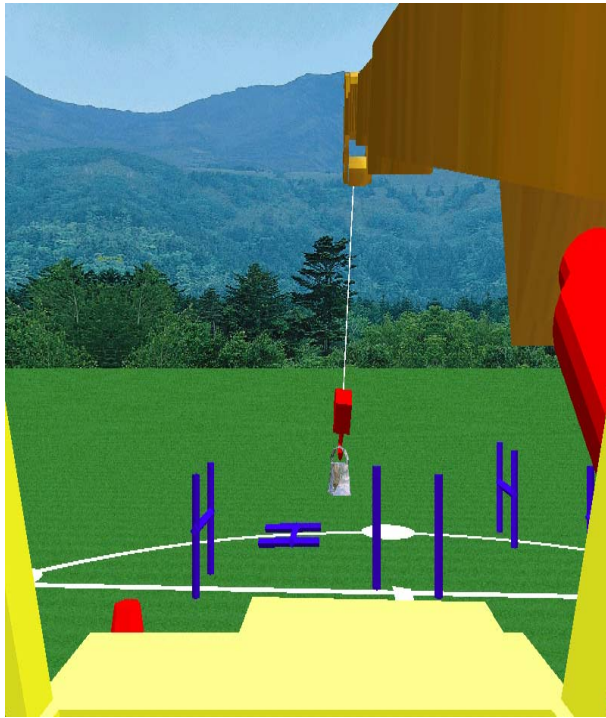

Figure 15. The scenario of mobile crane training

7. Conclusion and Future Works

This paper presents a peer-to-peer approach to design a high-fidelity simulator on a cluster of desktop computers. The legacy simulator systems often use the multi-processor mainframe as the interactive computing environment. With the advance of the computational research in the recent years, most of the recent researches on the interactive visual simulation begin to incorporate PCs into the simulating system. However, these studies were focused on server/client approaches and most of them used a multi-processor workstation as the server. This paper presents a peer-to-peer approach combined with the push and pull mechanism to construct an interactive visual simulation environment on a cluster of PCs. With a proper design of the communication layer, the presented peer-to-peer architecture can gain the effect of parallel computation on a set of networked computers. A major benefit of the presented mechanism is the modularization. Since the peer-to-peer architecture promotes a modular

design of an interactive simulating system, an additional function can be easily incorporated into an implemented system as required. Furthermore, this mechanism also promotes the reusability of the functional modules of an implemented interactive visual simulator.

Although the current system response rate is 16 times per second, further accelerating of the frame rate is possible and is currently under investigation. At this moment, the bottleneck of the implemented mobile crane simulator is the visual rendering engine. Since the current visual rendering engine was implemented from scratch using Microsoft Direct3D library, further speed-up is achievable if a commercial 3D graphical rendering package is used. In addition, more dynamics of the mobile crane can be added to the system to further increase the realism of the simulation. However, the dynamics computation often implies complex mathematical equations that may slow down the performance of the simulating process. Finally, the expert system can be integrated into the mobile crane simulator. Within a construction site, a human guide is often necessary to ensure a safe operation of the mobile crane. If such capability is to be incorporated into the simulating scenario, an expert system is required to reason the hand gesture of the guide.

Acknowledgment

References

[1] Bailey, R. W., *Human Performance Engineering: A Guide for System Designers*, Englewood Cliffs, N. J., U.S.A. : Prentice-Hall (1982).

[2] Carraro, G. U., Cortes, M., Edmark, J. T. and Ensor, J. R., "The Peloton Bicycling Simulator," *Proceedings of the Third Symposium on Virtual Reality Modeling Language*, Monterey, California, U.S.A., pp. 63-70 (1998).

[3] Chandy, K. and Misra, J., "Asynchronous Distributed Simulation via a Sequence of

Parallel Computations," *CACM*, Vol. 24, pp. 198-206 (1981).

[4] "Direct 3D Overview," Available at http://www.microsoft.com/directx/overview/d3d/.

[5] Fadlallah, G., Lavoie, M. and Dessaint, L. A. , "Parallel Computing Environments and Methods," *International Conference on Parallel Computing in Electrical Engineering (PARELEC)*, Trois-Rivieres, Que., Canada, pp. 2-7 (2000).

[6] Ferrazzin, D., Salsedo, F. and Bergamasco, M., "The MORIS Simulator," *8th IEEE International Workshop on Robot and Human Interaction (RO-MAN '99)*, Pisa, Italy, pp. 135-141 (1999).

[7] Greenberg, J. A. and Park, T. J., "Driving Simulation at Ford," *Automotive Engineering*, pp. 37-40 (1994).

[8] Huang, J. Y., et al., "Extension of RTI to Construct a Simulator over Distributed PCs," *1997 Spring Simulation Interoperability Workshop*, Orlando, FL, U.S.A., pp. 201-206 (1997).

[9] Huang, J. Y. and Bai, H. H., "The Synchronization Algorithm for Constructing CAVE System on the PC Cluster," *The Seventh International Conference on Distributed Multimedia Systems (DMS'2001)*, Tamsui, Taiwan, pp. 135-141 (2001).

[10] Kuhl, J., et al., "The IOWA Driving Simulator: An Immersive Research Environment," *IEEE Computer*, Vol. 28, pp. 35-41 (1995).

[11] Kwon, D. S., et. al., "KAIST Interactive Bicycle Simulator," *IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, South Korea, pp. 2313-2318 (2001).

[12] Lee, W. S., Kim, J. H. and Cho, J. H., "A Driving Simulator as a Virtual Reality Tool," *1998 IEEE International Conference on Robotics and Automation*, Leuven, Belgium, pp. 71-76 (1998).

[13] Lin, C. T., Chung, I. F., Liann, S. A., and Su, F. Y., "Multipurpose Virtual-Reality-Based Motion Simulator," *Proceedings of the 3rd World Congress on Intelligent Control and Automation*, Vol. 4, pp. 2699-2704 (2000).

[14] Menendez, R. G., Bernard, J. E., "Flight Simulation in Synthetic Environments," *IEEE 19th Proceedings of Digital Avionics Systems Conferences*, Philadelphia, PA, U.S.A., pp. 2A5/1-2A5/6 (2000).

[15] Moore, M. and J. Wilhelms, J., "Collision Detection and Response for Computer Animation," *Proceedings of the 15th Annual Conference on Computer Graphics*, Atlanta, Georgia, U.S.A., pp. 289-298 (1988).

[16] Park, M. K. et al., "Development of the PNU Vehicle Driving Simulator and its Performance Evaluation," *IEEE International Conference on Robotics and Automation (ICRA)*, Seoul, South Korea, pp. 2325-2330 (2001).

[17] PVM: Parallel Virtual Machine – http://www.netlib.org/pvm3/book/node1.html

[18] Rajkumar, A. R. R., Prakash, H. and Kishore, B., "PARDISC: a cost effective model for parallel and distributed computing," *IEEE 3rd International Conference on High Performance Computing*, Trivandrum, India, pp. 451-456 (1996).

[19] Roman, M., "Flight Simulators: A Look at Linux in the Aerospace Training Industry," *Linux Journal*, Vol. 1999 (1999). Available online at http://www.linuxjournal.com/.

[20] Srinivasan, S. and Reynold, P., "Communications, data distribution and others goodies in the HLA performance model," *1997 Spring Simulation Interoperability Workshop*, Orlando, FL, U.S.A., pp. 329-338 (1997).

[21] Tam, E. K., et al., "A Low-Cost PC-Oriented Virtual Environment for Operator Training," *IEEE Transactions on Power Systems*, Vol. 13, pp. 829-835 (1998).

[22] Tanenbaum, A. S., *Operating systems: Design and Implementation*, Prentice Pub., ISBN 0-13-637406-9 (1987).

[23] Taylor, V. E., Chen, J., Disz, T. L., Papka, M. E. and Stevens, R., "Interactive Virtual Reality in Simulations: Exploring Lag Time," *IEEE Computational Science and Engineering*, Vol. 3, pp. 46-54 (1996).

[24] The Message Passing Interface (MPI) standard - http://www-unix.mcs.anl.gov/mpi/.

[25] Yoneda, M., Arai, F., Fukuda, T., Miyata, K., and Naito, T., "Operational Assistance for Straight-line Operation of Rough Terrain Crane," *IEEE International Conference on Systems, Man and Cybernetics (SMC'99)*, Sendai, Japan, pp. 27-32 (1999).

[26] Zeltzer, D., Pioch, N. J. and Aviles, W. A., "Training the Officer of the Deck," *IEEE*

Computer Graphics and Applications, Vol. 15,
pp. 6-9 (1995).