Modelling and designing a low-cost high-fidelity mobile crane simulator

Jiung-Yao Huang\textsuperscript{a,}\textsuperscript{*}, Chung-Yun Gau\textsuperscript{b}

\textsuperscript{a}Department of Computer Science and Information Engineering, Tamkang University, 151 Yiung-Chuan Road, Taipei Hsien, Tamsui 251, Taiwan

\textsuperscript{b}Institute of Occupational Safety and Health, Council of Labor Affairs, Executive Yuan, Taipei, Taiwan

Received 15 February 2002; received in revised form 3 August 2002; accepted 8 October 2002

Paper accepted for publication by Editor, B.R. Gaines

Abstract

The interactive visual simulation is the integration of real-time computer graphics with multimodalities, such as acoustic display and force feedback, to create a realistic simulation scenario to the user. This paper presents a method to design an interactive visual simulation on a cluster of desktop computers for the mobile crane training. This mobile crane training simulation is a project sponsored by Employment and Vocational Training Administration, Council of Labor Affair, Executive Yuan, Taiwan, to build a low-cost yet effective vehicle for training and licensing. To achieve this goal, a set of locally networked PCs is employed to form a parallel computing environment for the mobile crane simulation.

The most important issue of developing a high-fidelity interactive visual simulator is its integration system for communication and monitoring among functional modules. This paper presents a peer-to-peer architecture on a cluster of PCs to develop the required integration system for the mobile crane simulator. In addition, the developed integration system uses the push-and-pull model to seamlessly communicate the messages among distributed functional modules. This push-and-pull model effectively achieves the parallelism among the distributed functional modules of a mobile crane simulator. With the push-and-pull model on the peer-to-peer architecture, we can easily achieve the modularity and reusability of the functional modules of the simulating system. The presented push-and-pull model satisfies four essential attributes for the parallel computing, which are concurrency, scalability, locality and modularity. Our experience also successfully verifies the effectiveness of the presented...
simulator with the system response rate of 16 times per second which is larger than human acceptable perception rate as suggested by the human factors studies.

© 2002 Elsevier Science Ltd. All rights reserved.

**Keywords:** Interactive visual simulation; Mobile crane simulation

1. Introduction

The virtual reality system is an interactive multi-sensual environment that integrates multimodal display to construct a realistic virtual world for the user to be immersed into this synthetic environment. Depending upon the application and the fidelity of the intended system, different modals are adopted which may include real-time 3D computer graphics, remote video image, binaural sound, and haptic feedback. The fidelities refer to the faithfulness of the sensory outputs generated by the virtual reality system to the user, which includes visual display, acoustic render and haptic feedback. Based upon the application domains, various types of hardware are adopted to fulfil the requirement of fidelity, such as head-mounted display, wide-angle stereoscopic display, stereo digital audio, position tracking system, hand and gesture tracking system.

The interactive visual simulator is the most successful development of the virtual reality researches. The interactive visual simulator integrates real-time computer graphics with other modalities, such as acoustic display and force feedback, to create a realistic simulation scenario to the user. Due to its cost-effective essence, the study of the interactive visual simulator is an active research topic in the virtual reality technique. Different types of interactive visual simulators with various degrees of fidelities were designed for different purposes, ranging from training and ergonomic design to medical treatment. As surveyed by Brooks (1999), several applications of the interactive visual simulation are currently in their production stages, which include vehicle simulation, entertainment, vehicle design, architecture design and spatial arrangement, training, and psychiatric treatment.

Depending upon the applications, the interactive visual simulator provides different degrees of fidelities for the simulated scenario. Among the existing interactive visual simulators, the flight simulation is the most mature and representative application. The flight simulator by CAE Electronics Inc. (Melnyk, 1999) was built with a complex and effective six degree-of-freedom (d.o.f.) motion system to generate the realistic feeling of takeoff, landing and in-flight turbulence. In addition, a mockup with faithful reproduction of cockpit interior is mounted on top of the motion system to immerse the user into the training scenario. On the other hand, the SE-based flight simulator at the Virtual Reality Applications Centre (Menendez and Bernard, 2000) uses the C2, which is a room of four projected surfaces, to create a fix-based airplane and helicopter simulator.

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 Hz for one full pass for all software modules (Melnyk, 1999). When a user pushes the pedal, the flight 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 multi-processor mainframe system, such as IBM AIX RS/6000 (Melnyk, 1999) or two as processors SGI rack Onyxes with two IR2 graphics pipes (Menendez and Bernard, 2000), 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 tasks of an application, we can easily cluster several computers by the local area network and employ the pipeline technique to design a high-performance distributive computing environment (Rajkumar et al., 1996). Hence, recent researches on the interactive visual simulation often incorporate desktop computers to construct their computing environments (Park et al., 2001; Lee et al., 1998). For the rest of this paper, an overview of the previous researches on the interactive visual simulator is given first. The modelling of the mobile crane simulator is discussed next. Based upon the analysis on the previous section, the design of modules of the mobile crane simulator is then elaborated. The clustering computing technique to integrate functional modules into a mobile crane simulator comes to last. Finally, the implementation of the mobile crane simulator comes to last and followed by the conclusion and future work.

2. Previous works

Asiding from the flight simulator, the other well-known application is the driving simulation. For example, the Ford company built a driving simulator (Greenberg and Park, 1994) with an ESIG-2000 computer from Evans and Sutherland as its image generator, a four-processor DN-10000 computer from Apollo to compute the motor dynamics, and a two-processor real-time input/output (I/O) computer from Harris NightHwak to control the data flow inside the simulator. The IOWA drive simulator (Kuhl et al., 1995) 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.

Compared to Ford Corp. and IOWA driving simulators, the PNUVDS (Park et al., 2001) is a low-cost driving simulator that was developed by Pusan National University. Four hosts were employed and configured in server/client architecture to construct its computing environment. A dynamics workstation was the server to receive user input from an PC, compute the vehicle dynamics, and then pass the result to SGI Indigo2 Impact R4000 to render the image and to a dedicated controller to manipulate the motion platform. On the other hand, Lee et al. (1998) presented their work of designing a driving simulator with four PCs only.
Other researches of the interactive visual simulation include various military training programs (Lindheim and Swartout, 2001; Zeltzer et al., 1995), power-plant operator training (Tam et al., 1998), and others. The Virtual Environment Technology for Training (Zeltzer et al., 1995) 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 Engine² graphics sub-system to support the complex training scenario in real time.

ESOPE-VR (Tam et al., 1998) is a VR operator training simulator prototype for power-utility personnel. Since the manual operation of the switching station equipment is a risky task and demands rigorous personal instruction, the ESOPE-VR is to provide a training environment for station operators. Based upon the requirements of the power system station operator, its functionality includes 3D 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 Expert-System for Operations Environment (ESOPE), is executed on a PC as the backbone of the entire simulation system. Furthermore, another PC is dedicated for speech recognition function to support the vocal command.

The overhead crane training system (Huang, submitted) is another example of the training application. The overhead crane is a porterage device that is commonly used in the manufacturing industry. In order to control the lift hook, the overhead crane requires the user to follow and operate the crane on foot. The overhead crane training system employed a locomotion device, called the omni-direction ball-bearing disc platform, to allow the trainee to be immersed inside the training scenario by following the overhead crane on foot while controlling the lift hook. This overhead crane simulator is also designed and implemented on a cluster of desktop computers.

The Peloton bicycler simulator (Carraro et al., 1998) is a sport simulator that uses the Virtual Reality Modelling Language 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 (Kwon et al., 2001) 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-d.o.f. motions, the handle and pedal resistance systems to provide force feedback, the visual simulating system to create the virtual scene. Its computing environment is composed of three PCs in the 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 mastered by a client PC. The server PC then forwards the computed result to the render PC to generate a realistic virtual campus.

The motorcycle rider simulator (MORIS) (Ferrazzin et al., 1999) is a motion-based two-wheel vehicle simulator. The MORIS is designed as a tool for the motorcycle manufacturer to test new prototypes before actually producing them. Its computing backbone was constructed by a Alpha workstation as the server, an SGI workstation to render the image, another Alpha workstation as mechanical sub-system to control the motion platform, and a PC to generate the audio sound.

This paper is to present a method and mechanism to design a mobile crane simulator on a cluster of desktop computers. Although Yoneda et al. (1999) also conducted a similar research on the crane operation, their work was concentrated on designing an operational assistance system to develop a control rule to assist straight-line transfer of the payload. Hence, its computing system is mainly built on a SGI workstation to compute the crane dynamics and render the image. In addition, a PC was used as its I/O host to receive the user input and control the force feedback of the joystick. Different from their work, the simulator presented in this paper attempts to build a high-fidelity mobile crane simulator on the networked PCs for training and licensing. Hence, the components of the presented mobile crane simulator include a realistic cabin mockup, a 6-d.o.f. motion platform, and a large field-of-view (FOV) display by three monitors. All of these three components are integral parts of a high-fidelity interactive visual simulator.

3. Analysis and modelling of the mobile crane simulator

The mobile crane, also called the rough terrain crane (Yoneda et al., 1999), is a common apparatus in the construction site or factory. Basically, a mobile crane is a hoisting machine mounted on 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 winched cable and a hook on one end. The operation of the hanging bracket is a complex and difficult task because of the following reasons (Yoneda et al., 1997):

- An operator must usually control a payload with two or three levers at once.
- A payload is suspended by a long cable, so it oscillates easily.
- It is hard to know the status of the payload from the driver’s seat.
- Crane operation requires extreme concentration.

Thus, it requires the operator to have sufficient skills to safely and efficiently manipulate the crane when lift a payload. Some studies were conducted to reduce the operating load of the crane system. These studies can be categorized into two types (Yoneda et al., 1997): the automatic (or semi-automatic) control system and operational assistance system. The former is to develop an automatic
(or semi-automatic) crane system to support operators. The other is the study of presenting various types of information to assist the user to perform safe and efficient 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 being driven. This situation often causes the crane to turn over. Hence, the driving simulation of the mobile crane is also an important training item for 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 3-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 safely and 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 similar to an operational assistance system (Yoneda et al., 1999), 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 licensing, the realism of the simulation is a very crucial issue. When operating a crane, the operator relies on three types of information, which are out-of-window view, various sounds, and meters and lights on the dashboard, to control the boom and the winching cable. Hence, in order to create a realistic simulating environment for the trainee, a mockup must be embedded with devices that can faithfully display these information to the trainee.

First, the mockup is embedded with a dashboard from the actual crane. Since the operational panel of a mobile crane includes two levers to simultaneously control the boom and the winching cable, and a dashboard of 26 indicators, 10 m and 18 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. Hence, an operational panel module is required as the I/O module that receives signals from the switches, buttons, and levers input on the dashboard and triggers the indicators and meters on the dashboard.

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 Menendez and Bernard (2000), these sensor cues may include a large FOV real-time stereoscopic visual display, surrounded sound digital audio, and haptic interaction. Experiments from NAVE (Seay et al., 2001) system show that a three-screen set-up surrounding display that provides a 180° FOV which allows the driver to have a higher feeling of presence and immersion in the surroundings. Hence, in order to further provide sufficient sensory cues to the trainee, a surrounding screen by three monitors is mounted inside the mockup to emulate the out-of-window views from the crane cabin. The surrounding window display provides wider FOV to convince the user that he is actually manipulating a mobile
crane. Hence, the visual display module is another output module that controls the display images of the surrounding screens.

The acoustic sound display is the third sensual display that is essential to immerse the trainee. The sound of a mobile crane operation includes background engine noises, boom extension, rotation and hoisting sounds, cable winding sound, and collision sound. The audio rendering module is then the third essential output module of the mobile crane simulator.

Since driving the mobile crane is a testing item for licensing, a driving simulation must also be integrated into the mobile crane simulator. The real-world mobile crane has separated the driving cabin from the crane control cabin. In order to simplify the training procedure as well as to implement the mobile crane simulator, the driving cabin is integrated into the mockup. A steering wheel and the gas pedal and brake then become part of the operational panel devices and are controlled by the operational panel module. Furthermore, a 6-d.o.f. motion platform is introduced to realistically simulate the hazard situation of driving. The motion platform controller module then becomes an output module of the simulator that receives messages to change the posture of the motion platform.

Hence, as depicted in Fig. 1, seven modules were identified to design the mobile crane simulator. These seven modules are operational panel module, motion platform controller, instructor monitor, scenario module, dynamics model, visual display and audio module. These seven modules are distributed among the networked computers and coupled with each other to form an interactive computing environment.

Since the mobile crane simulator aims to be the training and licensing vehicle, an instructor monitor module is a by-product module that is used by the instructor only. In terms of the mobile crane simulator, this instructor monitor module is another I/O module. It enables the instructor to set up testing scenario from the monitor module and receive messages from the dashboard, while the trainee is controlling the boom and the winching cable. The dynamics model module is responsible for computing the physical phenomena and postures 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.

---

**Fig. 1.** The modules of the mobile crane simulator.
4. Modules of the mobile crane simulator

4.1. The operational panel module

The operational panel module is a part of the I/O device for the simulator. There are two categories of I/O 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 familiar with the instruments. Such a mockup must include a dashboard which is faithfully replicated from an actual mobile crane. As shown in Fig. 2, the instruments on the dashboard contain various indicators, meters and switches.

Since the mobile crane simulator is composed of the crane simulation and the driving simulation, except the dashboard, the mockup is also inlaid with steering wheel, gas pedal and brake. Specifically, the input devices in the mockup include a steering wheel, brake and gas pedals, and two joysticks to manipulate the boom and the winched cable of the mobile crane. Furthermore, there are two categories of input devices on the dashboard, which are indicators and meters, and various switches. There are 26 indicators on the dashboard which include direction indicators, oil contamination, and suspension luck, etc. Ten metres are provided on the dashboard, which include the speed odometer, directional meter, and oil gauge, etc. In order to communicate with these inputs, the operational panel module uses an industrial analog/digital board to send and receive signals to and from the dashboard inside the mockup.

The operational panel module is a program that monitors the signal of each instrument on the dashboard and translates the received signal into the message to be passed to other modules. In addition, the operational panel module receives

Fig. 2. The replica of the dashboard.
messages from the instructor monitor, which will be fully discussed in the next section, to drive the meters and indicators on the dashboard. In order to conveniently debug the dashboard in case of the hardware failure, a user interface is designed, as shown in Fig. 3, for the operational panel module.

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 Fig. 4, which is a two-dimensional display on the status of the simulated mobile crane.

The top-left window of Fig. 4 displays the current rotation angle of the boom. Since the boom will shift the gravity of the mobile crane when it is rotating, it is an important safety factor to monitor the rotation angle of the boom. The top-centre window depicts the hoisting degrees of the boom. Similar to the rotation angle, the elevation angle will also cause hazard situation if the angle is overshot a certain safety value. The top-right window shows the current length of the winched cable, and the left-bottom window displays the elongate length of the boom. In addition, the pictorial information on each window is also to be digitalized and displayed on the small dialogue boxes next to the middle-bottom 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 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 in Fig. 4 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 dashboard, called the Dashboard window, as shown in Fig. 5. The dashboard window is the 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 operational panel 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.

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 6-d.o.f. postures to the trainee, including rotations and shifts with respect to the 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 (1965) in 1965 and called Stewart platform thereafter. As shown in Fig. 6, The Stewart platform is a parallel manipulator. It consists of a fixed base plate and an upper
payload platform connected by six extensible legs with ball joints at each end. These six manipulators can be expanded and contracted individually to control the gesture of the upper platform.

The motion platform controller is composed of a washout filter and an actuation system. The purpose of the washout filter is to transform the trajectories computed by the dynamics module into the actuators commands to generate realistic motion cues to the trainee without driving the simulator out of its workspace. The output of the washout filter is in the form of actuators lengths which are then used by the actuation system as command to drive manipulators.
In order to smoothly simulate the posture of the mobile crane inside the virtual environment, the washout filter also includes an interpolation equation to smoothly transform the posture of the platform between the two consecutive statuses. To achieve this goal, the washout filter must preserve the state value of the previous posture and interpolate the motions between the two successive 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 responsible for managing objects within the virtual world and, hence, evaluates the performance of the trainee. Since the purpose of this mobile crane simulator project is to design a vehicle for training as well as licensing, a scenario is required to evaluate the trainee. As shown in Fig. 7, the designed scenario includes driving the mobile crane from the starting point to a designated

![Fig. 7. Bird eye view of the virtual training ground.](image-url)
location and then operating the hoist. In order to increase the difficulty of training, the scenario requires the trainee to drive the simulated mobile crane through the sand land and hills.

After arriving at the examinatorial spot, the trainee is asked to operate the boom to hoist an object and move that object according to a specific trajectory. As shown in Fig. 8, bars are placed on the trajectory to obstruct the movement of that object. The trainee is required to lift the object located at the white circular mark at the left side in Fig. 8, to move this object to its right side along the trajectory and then back to its original position. Scores will be deducted if a bar is collided and the scores will be dynamically displayed on the status window that is shown in Fig. 4.

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 below.

When the mobile crane and its lift hook are moved in the virtual environment, the dynamics computation uses the multi-level collision detection algorithm (Moore and Wilhelms, 1988) to effectively detect 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 subobjects. 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 cable when the boom is moved. When the boom is moving, the dynamics module computes the inertia of the lift hook acts on the cable based upon the moving direction, speed and weight of the cargo. When the boom is stopped from 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 capability is another important factor for the realism of simulation. Since its centre of mass is higher than that of other types of vehicle, driving the mobile crane is also a dangerous process. The terrain following capability combined 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 Direct3D library. When a scene file is read, the Direct3D library will automatically organize all objects within the scene into a tree structure (Direct3D Overview). 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 communicate the dynamic module and the Direct3D’s scene tree.

As reported from experiments of the NAVE (Seay et al., 2001), a simulator with higher FOV will allow the driver to have greater sense of involvement and immersion to the simulated scenario. Three monitors are then mounted inside the mockup to create a surrounding display as shown in Fig. 9. The two side monitors are positioned at 120° angles to the centre monitor to give a three-sided display area, which create an approximately 120° degree FOV to the trainee. Three PCs equipped with high-performance 3D graphical acceleration board are used to render images for these three-sided monitors to emulate the left, centre and right views of the mobile crane, respectively.

In order to balance the rendering load on these graphical PCs, the rendering process is partitioned into server and client sites (Huang and Bai, 2001). The server PC is responsible for managing the virtual scene and performing the view-depended culling for each graphical PC. The computed result is then forwarded to each graphical PC through Ethernet for rendering. However, since each graphical PC is rendering images of different directions that contain different numbers of geometrical objects inside their respective view frustum, the frame rate of these
PCs must be fully synchronized to achieve a consistent surrounding view among these displays. To fully synchronize the display frame among graphical hosts, a synchronization box is designed as the ticker to trigger the display of graphical PCs. Fig. 10 depicts the architecture to support three-sided display of the surrounding view. This surrounding view system is fully synchronized with each other so that a consistent view will be displayed. With this surrounding view system, the trainee can fully immerse himself into the training scenario.

Fig. 9. Three synchronized monitors to provide a large FOV.

Fig. 10. The architecture of the three-sided display.
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. The PC-based simulator

The last step yet the most important step to design an interactive visual simulator is to develop an integration to communicate and coordinate the messages flowing among modules. This integration technology is an important factor that significantly affects the fidelity and performance of the designed simulator. Due to the progress of the computing technology, many of the recent researches on the interactive visual simulation either entirely use PC(s) (Lee et al., 1998; Tam et al., 1998; Huang, submitted; Carraro et al., 1998; Kwon et al., 2001) or adopt PCs to construct a heterogeneous system (Park et al., 2001; Ferrazin et al., 1999; Yoneda et al., 1999) as the computing environment. The traditional approach is to use the server/cluster architecture with a server to compute the dynamics of the simulated entity and coordinate the data among the clients. One potential drawback of this approach is that this server quickly 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 adopts the peer-to-peer architecture (Singhal and Zyda, 2000) on a cluster of PCs to construct the computing environment for the mobile crane simulator. In the following subsections, 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 communication

The concept of peer-to-peer approach is to treat the functional modules discussed in the previous section as logical processes (LPs) and to distribute them among locally networked PCs. For the rest of the paper, the functional modules and LPs are referred to the same thing. The messages among these distributed LPs are then transmitted in the peer-to-peer fashion. In this way, the simulation pipeline is then created on distributed LPs and the parallelism among these LPs is implicitly executed. There are two de facto standards to construct a parallel computing environment on a locally networked computing hosts, which are PVM\(^1\) and MPI\(^2\). However, since neither PVM nor MPI are designed for the interactive computing

\(^2\) The message passing interface (MPI) standard - http://www-unix.mcs.anl.gov/mpi/
environment, neither is suitable to be used as the integration system of the interactive visual simulator. Furthermore, neither PVM nor MPI are originally designed for the peer-to-peer architecture.

To design a peer-to-peer architecture for the interactive computing environment, there are two essential issues, which are the initialization mechanism and the communication mechanism among distributed modules, have to be solved. Since there is no server to coordinate the messages flow among the distributed LPs, the method to establish the communication links among the distributed LPs is a critical issue to achieve the peer-to-peer communication. Extension from the description of the peer-to-peer architecture (Singhal and Zyda, 2000), peer-to-peer means that we have to decide the distribution of tasks at the early design stage. This implies to design a custom-made integration system only for this mobile crane simulator. Even though it is possible for a custom-made integration system to achieve optimized performance, the simulating system with this approach has a very poor reusability. To overcome this problem, we can either use broadcast or multicast for packet transmission to “declare” the existence of an LP. That is, we can design a communication backbone (CB) as the agent for each host to broadcast the information of that host to solve the initialization problem of the peer-to-peer architecture (Huang et al., 1997).

The CB is the agent of a host that accepts the declaration from an LP on the same host on what kinds of messages that it wants to send. The CB is then responsible for broadcasting this message-dispatching event to CBs on other hosts. Similarly, an LP also needs to notify its CB on what type of message that it wishes to receive and the CB then “listens” to the network for the matched message-dispatching declaration. If there is a match, a link will be established between these two CBs along with other necessary information. In this way, each LP of this architecture run on top of the CB does not have to be concerned about the existence of other LPs. 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 the distributed LPs, as depicted in Fig. 11, no matter that the corresponded LP is in the same machine or across the network. One or many LPs can run on a computer, depending upon the computational load of each LP. With this capability, a heterogeneous computing

![Fig. 11. The infrastructure of the peer-to-peer communication.](image-url)
environment can be constructed, and different LPs can be easily plugged to form different types of simulation environment.

Since there is no server to coordinate the message flow among the distributed tasks on PCs, the asynchronous message passing method (Chandy and Misra, 1981) is adopted to design the required communication mechanism. As surveyed by Fadlallah et al. (2000), the message passing is the surest way to achieve the high performance for the majority of distributed applications. Hence, after the initialization phase, each LP only needs to convey its message to CB without knowing the existence of other processes. The CB will act like an agent of its LP to forward this message to the corresponded CB which then passes the received message to its servicing process. Hence, with the CB mechanism, each computer can be executed at its own pace and parallelisms among the distributed tasks are then automatically explored. Consequently, a transparent peer-to-peer communication infrastructure is created by means of this mechanism.

5.2. The push-and-pull model

Based upon the presented peer-to-peer communication architecture, a push-and-pull model (Srinivasan and Reynold, 1997) then can be adopted to design the integration system of the interactive visual simulation. The essential concept of the push-and-pull model is that each LP is a functional module of the simulation and it is locally executed as a standalone program. From the point of view of LP, CB plays the role of agency of the simulating 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 simulating 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 be the subscriber and publisher at the same time. Hence, during the initialization phase, CB will be responsible for matching publishers with their corresponding 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.

During the run-time phase, as illustrated in Fig. 12, the publisher will treat this simulating environment as a “push” model and “pump” in its data. On the other hand, the subscriber treats the simulating environment as a “pull” model that it can dig information out of the simulating environment. With this model, a transparent communication among LPs can be easily designed and constructed.

The virtual channel is an important notion to implement the push-and-pull model. Conceptually, a virtual channel is the pipeline that seamlessly interconnects two LPs to form a peer-to-peer communication. Physically, as shown in Fig. 13, 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 Subscription table. 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. Finally, there are four essential attributes for the parallel computing, which are concurrency, scalability, locality, and modularity (Fadlallah et al., 2000). Significantly, this push-and-pull model satisfies the essential attributes of the parallel computing.

5.3. The mobile crane simulator

A high-fidelity mobile crane simulator on a cluster of PCs can be easily integrated by the presented peer-to-peer architecture. Furthermore, the push-and-pull model enables a new functional module of the simulator can be dynamically added to the
existing interactive computing environment. Fig. 14 illustrates the infrastructure of the resulted mobile crane simulator constructed by the push-and-pull model.

For the 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 mobile crane simulator is composed of seven modules, which are I/O control module, instructor monitor, scenario control module, dynamics module, visual display and audio rendering modules. In addition, the visual module contains a display server module and three rendering modules to support a three-sided display area. All of these ten functional modules are implemented on the Microsoft Windows platform and are distributed among the rack of seven PCs as shown in
Fig. 15. These seven PCs are connected by 100 Mb Ethernet network to form a distributed interactive computing environment.

The top three PCs are graphical computers which execute rendering modules to generate images of the three monitors inside the mockup to support a large FOV. Each graphical PC is equipped with a TNT2/M64 3D graphical acceleration card. The fourth PC from the top is the display server that synchronizes the frame rates of the above three graphical PCs (Huang and Bai, 2001). In order to precisely synchronize the displays rendered by the three graphical PCs, a hardware synchronization box is designed. The main function of this synchronization box is to trigger all the graphical PCs to simultaneously display their respective frame images and collect the acknowledge messages from all the graphical PCs. This triggering action is activated by the display server after it has transmitted the view-dependent information to each graphical PC by network. With the help of the synchronization box, the display server can accurately control the image displays among the graphical PCs without being interfered by the network bandwidth.

The fifth PC is the I/O computer that executes the 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. Fig. 16 shows the appearance of the designed mobile crane simulator.
The virtual world of the training scenario contains 3235 polygons that are rendered at $1024 \times 1024$ pixels with 24 bits. The simulation lag (Taylor et al., 1996) of the implemented mobile crane simulator is 0.0625 s, which is equivalent to the system response rate of 16 times per second. 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 ms (Bailey, 1982), our mobile crane simulator is around 62 ms and is proven to be a successful high-fidelity interactive visual simulating system on a cluster of PCs.

7. User experiences

The mobile crane simulator was moved to Central Vocational Training Centre for the vocational training of the mobile crane operator. Central Vocational Training Centre, Council of Labor Affair, Executive Yuan, is an Employment and Vocational Training Administration bureau in the central part of Taiwan. The mobile crane simulator enables the vocational training of the mobile crane to be held anytime during the office hours which greatly increases the training effectiveness. Several user experiences were as feedback since the simulator was transferred to that location. The user experiences can be summarized as follows:

Fig. 16. The mobile crane training system.
1. **The simulation fidelity**: Most of the users grumble about lacking a human guidance which obstructs their immersion into the simulating scenario. In the field operation of the mobile crane, a human guidance is required to monitor the hanger and instruct the operator. The operator often relies on the guider to safely manipulate the crane. Hence, the users are complaining about their mental pressures without the assistance of the guider. However, in order to realistically simulate such a human guider inside the virtual world, the artificial intelligent technique must be incorporated to generate an accurate guiding gesture. Since this capability is beyond the goal of this mobile crane experimental project, the study of the human factor and the psychological issue will be the next step to design the virtual human guider.

2. **The depth perception**: The users are complaining about the missing depth perception for them to accurately operate the mobile crane. The depth perception is the most important clue for the human beings to grab an object. Although the simulator employs three-screen consecutive displays to provide around 120° of FOV, it does not fully redeem the issue of depth perception. As illustrated in Fig. 17, without the depth perception, the operator has the trouble to move the hanger through the H barrier especially when the hanger is vibrating. To relief this deficiency, the shadow of the boom and hanger are added to the system. With the help of the shadow, this issue is alleviated.

3. **The visual realism**: Finally, the last issue is the imaging reality of the screen. Some users dislike the saturation and colours of the virtual scene which they thought that it is not realistic enough. Since the render engine of the simulator is based upon Microsoft Direct3D rendering engine, the imaging reality is not comparable.
with that of a workstation such as SGI Onyx machine. The experience of the designer is another factor for the realism of the virtual world. Since this issue can be easily improved by using new rendering hardware and employing professional designer, we will not discuss this issue further. However, the users did point out that the realism of the training scenario will allow them to ignore the realism of the scene image.

8. Conclusion and future works

This paper presents the principle and the architecture of a PC-based high-fidelity mobile crane simulator. The legacy interactive visual simulating systems often adopt the multi-processor mainframe as the backbone to develop its supported computing environment. With the evolution of the Silicon technology in the recent years, the modern personal computer is also equipped with high performance computing power. Hence, most of the recent researches on the interactive visual simulation begin to incorporate PCs into the simulating system. However, previous 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 architecture on locally networked PCs to develop an interactive visual simulation environment.

The presented peer-to-peer architecture adopts the push-and-pull model to gain the effectiveness of parallel computation on a set of networked PCs. Significantly, our experiment successfully demonstrates this point. A major benefit of the presented mechanism is that it enables modular development of the interactive simulating system. Since the peer-to-peer architecture promotes a modular design of an interactive simulating system, an additional function or fidelity can be easily incorporated into an implemented system as required. Hence, a high-performance multiple-fidelity interactive simulation environment can be constructed with the presented architecture. In addition, this paper also elaborates the method to model and design a high-fidelity mobile crane simulator on a cluster of PCs.

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 3D rendering engine. Since the current rendering engine was implemented from scratch using the Microsoft Direct3D library, further speed-up is achievable if a commercial 3D graphical rendering package is adopted. In addition, more dynamics of the mobile crane can be added to the system to further increase the realism of the simulation. However, the dynamic computation often implies complex mathematical equations that may slowdown 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.
Acknowledgements

This project was supported by Division of Occupational Safety, Institute of Occupational Safety and Health, Council of Labor Affairs under contract number IOSH89-S131. The presented mobile crane simulator was a three-year project which was contributed by many hard working graduate students of department CS&IE, Tamkang University.

References


Direct3D Overview. Available at http://www.microsoft.com/directx/overview/d3d/


