An Omnidirectional Stroll-Based Virtual Reality Interface and Its Application on Overhead Crane Training

Jiung-Yao Huang, *Member, IEEE*

***Abstract*—Locomotion is an virtual reality interface that enables the user to walk inside the virtual environment in any directions over a long distance without actually leaving the physical device. In order to enable the user to freely navigate the virtual world and get fully immersed into the virtual environment accordingly, a locomotion device must fulfill the following two distinct requirements. First, it should allow the user to navigate an infinite distance within a limited area. Secondly, the user should not need to wear any tracking devices to detect his motion. This paper presents a locomotion mechanism called Omni-directional Ball-bearing Disc Platform (OBDP), which allows the user to walk naturally on it and thus to navigate the virtual environment. The gait sensing algorithm that simulates the user's posture based upon his footstep data collected from the OBDP is then elaborated. Followed with an omnidirectional stroll-based virtual reality system to integrate the OBDP with the gait sensing algorithm. Significantly, instead of using the three-dimensional (3-D) tracker, the OBDP adopts arrays of ball-bearing sensors on a disc to detect the pace. No other sensor, except the head tracker to detect the user's head rotation, is required on the user's body. Finally, a prototype of the overhead crane training simulator that fully explores the advantage of the OBDP is presented at the end of this paper along with the verification of the effectiveness of the presented gait sensing algorithm.**

***Index Terms*—Gait analysis, locomotion, overhead crane, virtual reality.**

I. Introduction

THE virtual reality technique has been developed over the years and widely used in various realms. Most significantly, the virtual reality technique is not an invention of a new technology. Instead, it is an integration of the existing technologies, such as three-dimensional (3-D) computer graphics, image processing, sound, network and real-time control. With the help of these integrated technologies, the virtual reality technique creates a user-friendly interface for the existing applications, such as surgical training, military drill, merchandise advertising, and recreation.

Interactive visual simulator [1]–[4] for training adds mechanical control and dynamic computation to the virtual reality technique to create a more realistic simulation environment for the trainee. In addition, the interactive virtual simulator is often equipped with a cabin to emulate the training device for the trainee to be fully immersed into the simulated environment. The interactive visual simulator has the benefit of providing a safe and realistic environment for the trainee to exercise his skill repeatedly. Furthermore, the interactive visual simulator can help to integrate the training procedure with the data acquisition for the drillmaster to ameliorate the training courses. Hence, different virtual simulators have been developed over the years, such as the flight simulator [1] and the mobile crane simulator [4]. However, diverse machineries may be required to associate with different interactive visual simulators for various training purposes. For example, a motion platform and a fly cockpit are often used together for the flight simulator.

The locomotion device is a different type of input device for the virtual reality system which allows the user to navigate the virtual world on foot. That is, locomotion is a mechanism that enables the user to walk inside the virtual environment in any directions over a long distance without actually leaving the physical device. The treadmill was the first device that was used to design the locomotion interface for the virtual environment [5]. However, the legacy treadmill only can support one directional movement, which constrains the freedom of navigation inside the virtual environment. Darken [6] and Iwata [7] both proposed omnidirectional treadmills to relieve the directional constraint to the user. The Uniport, on the other hand, is a different type of locomotion device that uses a modified exercise-bike to simulate the user's pace [8].

This article presents a completely new type of locomotion device, called the Omni-direction Ball-bearing Disc Platform (OBDP), which provides a natural way for the user to walk inside the virtual environment. Instead of using the 3-D tracker, arrays of ball bearing sensors on a disc are used to detect the pace of the user. No other sensor, except the head tracker to detect the user's head motion, is required on the user's body. In addition, the ball bearing on the sensor slips the user's foot back to the center position of the disc. Compared with other locomotion devices [6]–[15], the OBDP is the locomotion interface that is more conformed with the kinesiology of the human being. A prototype of the overhead crane training simulator that fully explores the advantage of the OBDP is also designed and introduced in this article.

In Sections II–IV, an overview of related researches on the locomotion device is given first. The architecture of the ball-bearing disc platform is then fully discussed. The system module of the overhead crane simulator follows at the end.

Manuscript received January 28, 2000; revised September 21, 2001. This work was supported by the Institute of Occupational Safety and Health, Council of Labor Affairs, Executive Yuan, Taiwan, R.O.C., under Grant IOSH88-S132.

The author is with the Department of Information Engineering, Tamkang University, Taipei 251, Taiwan, R.O.C. (e-mail: jiungyao@ms45.hinet.net).

Digital Object Identifier 10.1109/TMM.2003.808822

II. Previous Works

Locomotion is the mechanism that enables the user to walk inside the virtual environment in any directions over a long distance without actually leaving the physical device. As indicated by [9], the locomotion device provides a natural way for the user to navigate the virtual environment on foot. Different locomotion mechanisms have been developed over the years, including treadmill [5]–[7], [10], [11], pedaling device [8], [11], motion capture [12], [13], walking-in-place [14], and virtual perambulator [15].

The treadmill, which was originally used for physical fitness, is the simplest way to build a locomotion device for the virtual environment maneuvering. UNC [5] developed the first locomotion device in 1986 to explore the possibility of designing an intuitive user input device. In order to provide the freedom of navigation, they designed a steering bar, which is similar to a bicycle, to enable the user to change his walking direction.

One of the major problems of designing the locomotion interface is that, it can not accurately simulate the walking behavior which hinders the user to be fully immersed into the virtual space. For example, when the human is walking on an uphill, gravity and inertia will act on his body and ankle to counteract his forward motion. This situation is only one of the examples that will divergence the real world walking from the virtual space locomotion.

To accurately simulate human pedestrian behavior, the University of Utah designed another type of treadmill, Treadport [10], to provide the inertia force feedback to the user. The Treadport is the system that is concentrated on simulating inertial force acts on locomotion, and a tether on the Treadport is designed to achieve the goal. A controller was proposed to act on the tether to cancel the backward inertial force that was caused by the acceleration of the treadmill. Nevertheless, Treadport is still a unidirectional locomotion device.

To revoke the restriction on the freedom of navigation that is imposed on the previous treadmill systems, the Naval Postgraduate school proposed an omnidirectional treadmill (ODT) [6] to allow the locomotion in any directions. The ODT consists of two perpendicular treadmills, one inside the other, to respond to the locomotion of the user. In addition, a mechanical tracking arm is attached to the user's waist to detect the locomotion direction. A servo motor drives the displacement and rotation of the top and bottom treadmills based upon the data from the mechanical tracking arm. By this way, the ODT maps an infinite space of the virtual environment to a small physical space and the user can freely navigate the virtual world.

The Torus treadmill [7], which is built by the Institute of Engineering Mechanics, University of Tsukuba, Japan, is a further enhancement of the omnidirectional treadmill. The Torus treadmill employs twelve treadmills to move the walker in the X-direction. These twelve treadmills are mounted on two endless rails that are actuated by four chains, which move the walker along the Y-direction. Two motion trackers are set at the knees to detect the moving direction of the user. An infinite walking area is then simulated by a torus that was implemented by these treadmills and chains.

There are other variations of virtual reality devices to provide difference sense of locomotion. UniPort [8] is a bipedal device which is very similar to a unicycle. The user pedals to simulate walking or running and uses his waist and thigh to change the direction of locomotion. GaitMaster and ATLAS [11] are two motion-based locomotion devices that integrate the motion-platform with the pedal and treadmill, respectively, to provide active response to the user's walking. GaitMaster uses the sensor on the motion-base to trace the position of the foot and uses the turntable to detect the orientation of the walker. On the other hand, ATLAS mounts a CCD camera with an infrared light filter and an infrared lamp in front of the treadmill. The detection of walking is achieved by tracking the video for the reflection from the IR marker fitted on each ankle of the user.

Virtual Perambulator [15] uses the rubber sandal with a low friction film to enable the user to roam inside the virtual world. The magnetic sensor and touch sensor are two essential mechanisms used by the Virtual Perambulator to trace the walking of the user. Significantly, Walk-in-place [14] is a particular type of mechanism that simulates the actual walking by the head motion without the user to physically locomoting. To achieve this goal, a neural network was designed to recognize the head motion and simulate the walking accordingly.

III. Omnidirectional Stroll-Based Virtual Reality System

Although the locomotion devices discussed previously had successfully provided mechanisms for the user to navigate the virtual world on foot, however, all of them either required the user to wear tracking devices [6], [7], [10] or provided an unnatural way to constrain the user motion [5], [8]. In order to enable the user to freely navigate the virtual world and get fully immersed into the virtual environment accordingly, a locomotion must fulfill the following two distinct requirements. First, it should allow the user to navigate an infinite distance within a limited area. Secondly, the user should not need to wear any tracking device to detect his motion. The OBDP, as shown in Fig. 1, is the first locomotion device that satisfies these two prerequisites. The OBDP is designed by the Division of Occupational Safety, which belongs to Institute of Occupational Safety and Health, Council of Labor Affairs, Executive Yuan [16], Taiwan. In this section, the principle of designing the OBDP is given first. The gait sensing algorithm that fully explores the merit of the OBDP then follows. Finally, this paper will be ended with the discussion of integrating the OBDP with other virtual reality components, such as 3-D render and audio, to design a complete omnidirectional stroll-based virtual reality system.

A. Omnidirectional Ball-Bearing Disc Platform

The omnidirectional ball-bearing disc is originated from the project of developing an overhand crane training device. It is a round-shaped disc with scattered ball bearing sensors. Compared with other locomotion devices, the OBDP is unique in terms that no motor as well as tracking device are used to provide omnidirectional maneuvering. Instead, as shown in Fig. 1,

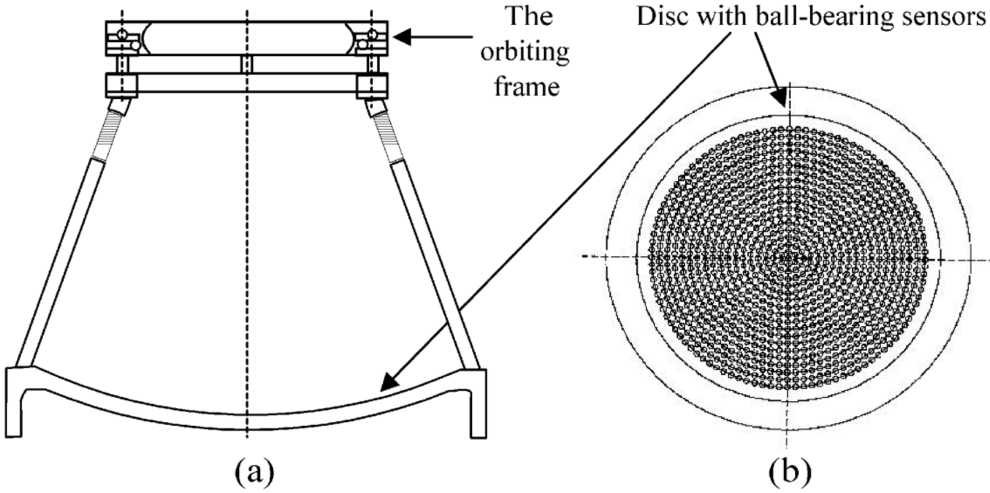


Fig. 1. Omnidirectional ball-bearing disc locomotion device: (a) side view; (b) surface of the OBDP.

OBDP uses arrays of ball-bearing sensors to detect the user pace and support the omnidirectional navigation. The OBDP has the following distinct features as originated from its design principle:

1) User Does Not Need to Wear Any Tracker to Detect his Movement: As discussed previously, in order to enable the user to navigate freely and thus fully immerse himself inside the virtual world, a locomotion device should not demand the user to wear any *tracking* devices to detect his motion. The best approach to solve this issue is to design a platform with scattered sensors to detect the pace of the user and simulate his movement inside the virtual world. However, the adopted sensors should not obstruct the freedom of navigation while the user's footstep is detected. That is, the sensor should not have too much sluggishness to hamper the tread of the user while detecting his pace.

Stemmed from the above study, the ball bearing sensor, as shown in Fig. 2, is designed for the OBDP. Each ball bearing sensor is mainly assembled with steel balls, an isolation axle, a spring and a stainless shim. In order to smoothen the rolling of the steel ball, six small steel balls of 0.25-in diameter are inserted between a big one of 0.75-in diameter and the isolation mandrel. The 0.75-in diameter steel ball on the top of the six 0.25-in diameter small steel balls provides a slick surface for the user. Yet, the frictions of the six 0.25-in diameter small steel balls and the stainless shim prevent slippery while the user is walking on it. There are totally 975 sets of ball bearing sensors embedded on the surface of the OBDP which can detect the pace between 50 cm and 5 cm.

2) It Allows the User to Perform Two-Dimensional Walking in a Limited Area: One of the distinct feature of the locomotion device is that it allows the user to navigate the virtual environment over a long distance within a limited physical area. To achieve this goal, a treadmill is a common mechanism to pull the user back to the center after he has moved forward. However, this approach cause several problems. First, a tracking device needs to be attached to the user in order to detect his motion to launch the treadmill. Secondly, it is very difficult to accurately control the speed of the treadmill which often stumbles the user in opposite direction when he comes to a sudden stop from the previous motion [6].

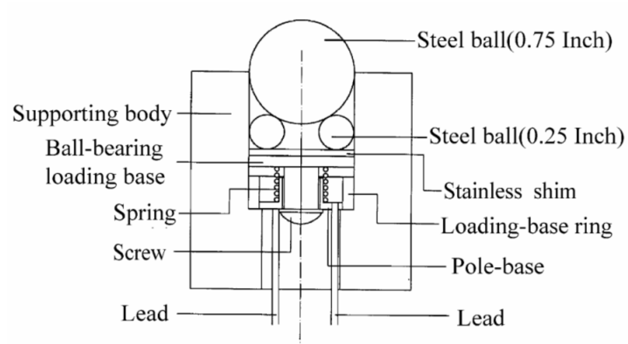


Fig. 2. Cross-sectional view of the position sensor.

To resolve the above two problems, the OBDP uses the ball bearing sensors as presented above to accomplish the goal of retaining the user in the center area. The surface of the OBDP is laid with arrays of ball bearing sensors in rings, as illustrated in Fig. 1, to support the two-dimensional walking of the user. In addition, the surface of the OBDP was designed based upon the natural gait of the human being. While walking, the human being will either swing his one leg in the air with the other on the ground or have both feet touch the ground. The arc shape design on the disc surface is based upon the swing angle of the human walk, so that the foot can slip back to the center of the disc when the user is walking on the disc. By this way, the OBDP allows the user to navigate the virtual world without actually leaving the OBDP. Since the shape of the disc surface is based upon the swing angle of human legs, the user can walk on the OBDP with a natural posture of walking without specific training.

3) OBDP has a Safety Support to Free the User's Hand: Compared with other locomotion devices [6], [7], the OBDP uses the gravity of the user along with the curvature of the disc surface to slide him back to the center of the disc. Although the OBDP will not stumble the user when he comes to a sudden stop from the previous motion, the curved surface may still un-

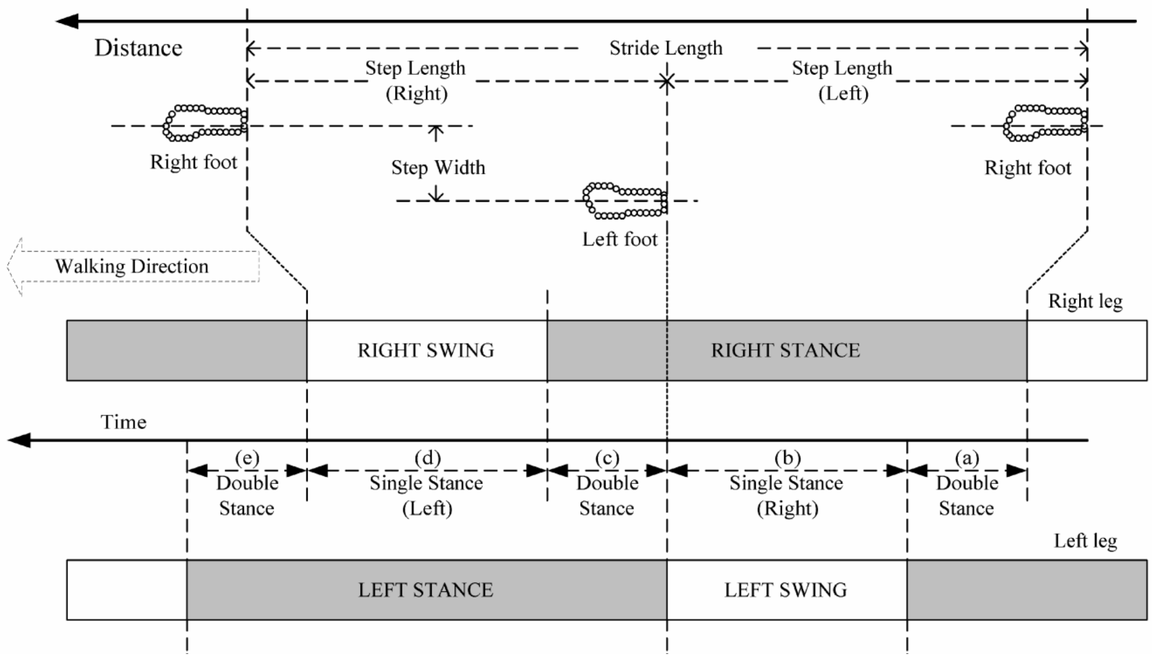


Fig. 3. Definition of temporal distance of the walking posture.

balance the user when a novice is walking on it. To solve this problem, the OBDP has an orbiting frame on the user's waist.

This orbiting frame can be easily rotated with the user when he turns around. In addition, this orbiting frame is a half-open support to make it easy for the user to pass in and out. During the operation, this orbiting frame can constrain the user in the center of the OBDP and equilibrate him when he is walking. Furthermore, the supporting frame can simulate the friction on the foot by counteracting the force feedback produced by the foot slipping on the surface of the OBDP.

4) OBDP Can Interactively Respond to the User's Stroll: One of the main issues of building a locomotion system with the omnidirection disc is that the number of the signals collected from the ball bearing sensors is large. In order to provide interactivity for the user, the OBDP uses a set of A/D and D/A interface cards to translate the signals of the ball bearing sensors into the computer to analyze the gait. The surface of the disc is laid with 975 point-sensors in 19 concentric circles. In order to increase the scanning rate of these point sensors, they are logically organized into a matrix of 28 rows and 36 columns. By this way, the system can effectively scan the point-sensors and analyze the gait within 10 ms to realistically reflect the walking posture of the user.

B. Gait Model

After the position information of the user is collected from arrays of ball bearing sensors, a gait sensing algorithm is required to recognize the footstep of the user and reason his gait accordingly. Hence, the walking postures of the human have to be studied first. By the study of the human walking posture, a gait model is developed. The gait model is then used to derive the walking status of the user after his footstep is recognized.

1) The Study of Gait: According to the dictionary, the meaning of "Gait" is generally referred to the manner of walking or running of the human. Walking is a natural motion of the human. However, it requires complex coordination among the skeleton, muscle, and nervous systems to accomplish this basic task. Hence, the gait analysis is widely studied by the biomechanics, the exercise dynamics, the neurophysiology, and the anatomy. Among these studies, the gait research from the biomechanics is the most objective and has a long history.

There are three important parameters from the gait analysis researches, which are temporal distance, ground reaction force, and knee joint and ankle joint angular motion. These three parameters have different definitions and values depending upon the area of researches. Our research is based upon the definition proposed by E. Y. Chao [17] in 1983.

In order to correctly calculate the walking distance from the walking posture, Chao added a temporal reference point. This reference point, as shown in Fig. 3, was cited from the medical science. The arguments from this definition for the temporal distance can effectively help us to model the human walking status.

As shown in Fig. 3, during the time intervals (a), (c), and (e), both feet are touching the ground. On the other hand, only the foot of a leg is standing on the ground and the other leg is swinging in the air at time intervals (b) and (d). In the following discussions, for the time intervals (b) and (d), we call the leg that is touching the ground as the stance leg and the other one as the swing leg. Hence, the human walking posture is then composed by a series of exchanging the swing leg and the stance leg between the two legs.

2) State Diagram of Walking: As presented in Section III-B1, we can calculate the human's walking direction, speed and distance by continuously detecting the position of both feet. That is, from Fig. 3, it is clear that the walking distance can be derived from the step length. The step length is measured as the distance between the two feet when they are

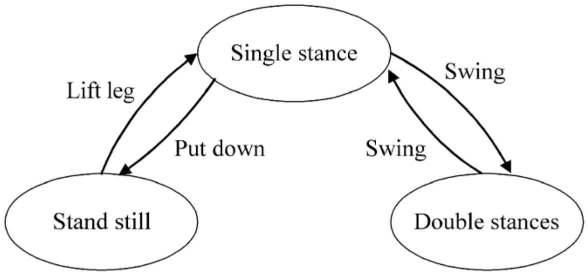


Fig. 4. State diagram for the gait.

both touching the ground and are not side by side, as illustrated in Fig. 3. The walking speed is the value of dividing the step length by the time interval (b) or (d). Hence, when the stance leg is recognized, we only need to continuously check the status of the swinging leg to detect the step length. With this observation, a gait sensing mechanism can be designed by continuously calculating the step length.

Further, since the walking posture is equivalent to the position change of the user's gravity, there is no difference if the swing leg is the left leg or the right one. Hence, the gait sensing algorithm does not need to distinguish the left foot from the right one. For example, note the following two cases.

Case 1) The user is facing the North and he swings his right leg forward.

Case 2) The user is facing the South and he swings his left leg backward.

Both cases have the same meaning to the gait analysis. That is, since both cases have the similar footprint in the same direction, both cases will be interpreted as moving to the north. Hence, the gait sensing mechanism can be simplified by calculating the gravity of the user to derive his moving distance and direction without considering which is the stance leg and which is the swing leg. With this observation, the walking posture can be modeled by the state diagram as shown in Fig. 4.

Fig. 4 depicts that the walking posture can be modeled by three states, the Stand still state, the Double stance state, and the Single stance state. The Stand still state represents the case when both feet are on the ground, side by side. The Double stance state also represents the situation when both feet are on the ground. Different from the Stand still state, however, the two feet are distance apart in the direction of moving. The Single stance state is the situation when one leg is swinging in the air. Hence, the walking starts from the Stand still state. When the user swings out one leg, he immediately enters the Single stance state. The user can go back to the Stand still state when he decides not to move forward. When his swinging leg touches the ground, he enters the Double stance state. He will remain in that state until he shifts his gravity toward the direction of moving and launches the other leg to swing to the direction of moving, which brings him back to the Single stance state again. The walking posture of the human is then a continuous cycle between the Double stance state and the Single stance state.

3) Gait Model for OBDP: After deriving the state model for the gait, we can then use the above model to design the gait sensing algorithm. However, due to the distinct feature of the OBDP, the above diagram has to be slightly modified. Since the OBDP always constrains the user in the center of the platform, the user can not shift his gravity and swing the other leg. Instead, the original swing leg slides back to the center of the platform as illustrated in Fig. 5.

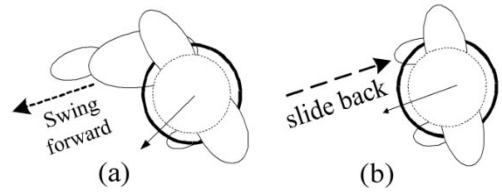


Fig. 5. Walking posture on the OBDP: (a) the right leg is lifted off and swung forward to prepare walking and (b) the right leg is slipped back to produce locomotion.

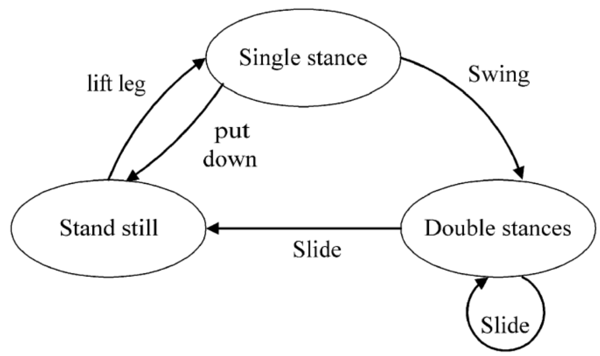


Fig. 6. Gait model for OBDP.

To match up with the constraint of the OBDP, we divide the sensors on the surface of the OBDP into two areas, the center area and the sliding area. The center area is used to decide whether the user is in the Stand still state. That is, whether his both feet are within the center area. When the user is not in the Stand still state, the mechanism will then check the sliding area to detect if the user is in either the Single stance state or the Double stance state. When the Double stance state is detected, the mechanism then continuously probes the sliding area to decide the distance between the two feet until the swung foot slides back to the center area.

Hence, we can modify the state diagram in Fig. 4 into the gait model as shown in Fig. 6 to accommodate the feature of the OBDP. When the user has his both feet in the center area, he is in the Stand still state. If the user raises one of his legs, he enters the Single stance state. After the user swings his leg to the sliding area, he then enters the Double stance state and remains in that state until he slides his foot back to the center area.

C. Gait Sensing Algorithm

Based upon the gait model in Fig. 6 and the feature of the OBDP, the gait sensing mechanism includes the following functional modules.

1) Gait States Reasoning: The first step for the gait sensing mechanism is to collaborate the size of the feet. Since the data received from the OBDP are arrays of 1-bit values from the point-sensors, which are meaningless unless the size of both feet is known. Hence, the user is requested to stand still in the center area at the initialization stage. By counting the bits clusters in

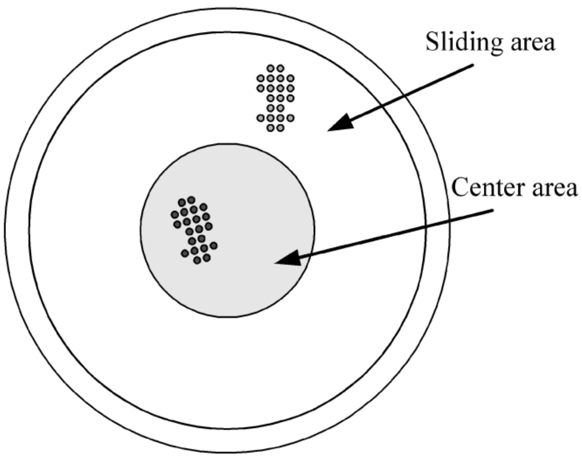


Fig. 7. Gait state recognition.

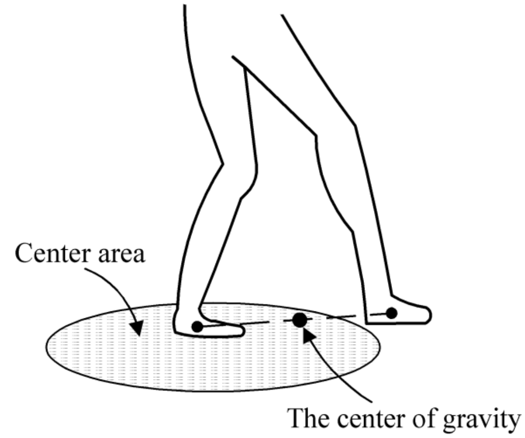


Fig. 8. Simulating the gravity point of walking.

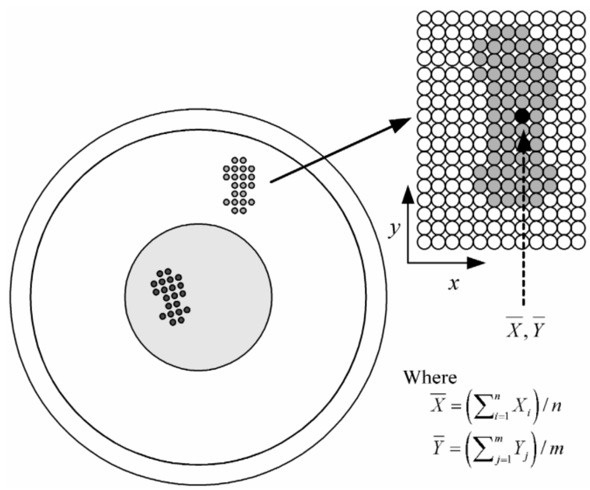


Fig. 9. Computation of the geometrical center.

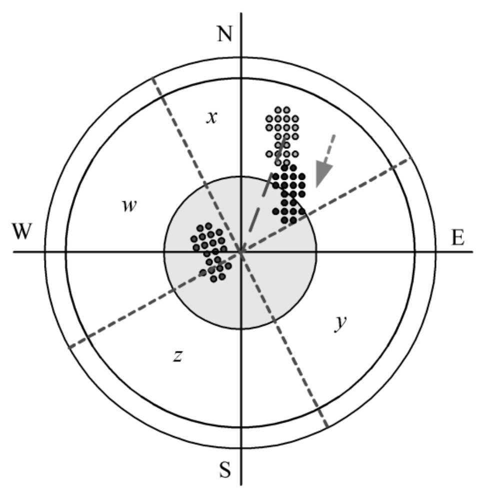


Fig. 10. Speeding up the state recognition by sectoring.

the center area, the size of the two feet is detected and recorded. We call this bit-clusters for the two feet as the feet-clusters. Similarly, the bit-cluster of a single foot is called foot-cluster. This sizing information is then used to detect the walking state of the user as modeled in Fig. 6. Notice that we assume the tolerance of the sizing difference is 15%. That is, the size of the foot-cluster may be 15% smaller or larger than one-half of the feet-cluster.

By continuously detecting the center area of the OBDP, when the size of the bit-cluster changes from the feet-cluster to the foot-cluster, we then reason that the user is entering the Single stance state. The sliding area then begins to search for the foot-cluster. When the foot-cluster is detected in the sliding area, the user then enters the Double stance state as illustrated in Fig. 7. Fig. 7 shows that, except the foot-cluster in the center area, there exists another foot-cluster in the sliding area and the user is in the Double stance state.

After the user is in the Double stance state, the sliding area will be continuously detected until the size of the bit-cluster on the sliding area is 85% smaller than the foot-cluster. At this moment, we assume that the user is back to the Stand still state and a cycle of walking posture is completed.

2) Geometrical Center of the Footstep: Since the walking posture and the shifting of the gravity have a causal relationship, we can use the position change of the gravity to represent the status of walking. In addition, the gravity of the human always balances between the two legs. Hence, we can simulate the position of the gravity from the information of the position of the two feet. First of all, we calculate, respectively, the geometrical center points of the left and the right feet, and then simulate the gravity point as the middle point between these two geometrical center points, as illustrated in Fig. 8. Further, we use the foot-cluster to calculate the geometrical center of each foot. For each foot-cluster, we sum up the X-coordinates of every bit and divide the result by the number of bits to derive the X-coordinate value of the geometrical center. Similarly, the Y-coordinate value of the geometrical center is also calculated as illustrated in Fig. 9. By this way, we can easily compute the gravity change without complicated pattern recognition and accommodate the cluster size change with the walk posture at the same time.

3) Speed up the State Recognition by Sectoring: We can further speed up the gait sensing algorithm by constraining the state recognition process only in the sliding direction when the state is transited from the Double stance state to the Stand still state. As shown in Fig. 10, when the Double stance state is detected, we can draw a dashed line from the geometrical center of the foot-cluster in the sliding area to the center of the OBDP. The surface of the OBDP is then logically divided into four sectors

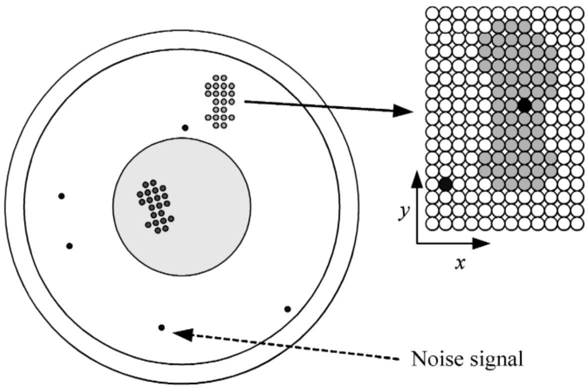


Fig. 11. Noise signal on the OBDP.

with the dashed line as the midline of one of the sectors, which is sector x in Fig. 10. As shown in Fig. 10, under a normal walking posture, the foot in the sliding area will slide back to the center area only within the sector area x. Hence, we can speed up the gait state reasoning process by scanning the sector area x only, until the Stand still state is reached.

4) Noise Filtering: Another benefit of sectoring the surface is that the sectors can help us to filter the noise signal from the point-sensors. Since the point-sensor is a ball-bearing pressure-based switch, occasional malfunction of the sensor is unavoidable. As shown in Fig. 11, this malfunction will induce noise signal to the scanned bit-arrays. This noise signal will influence the accuracy of computing the geometrical center of the foot-cluster. That is, such a noise signal will shift the computed geometrical center from its actual position.

To eliminate such a noise effect, only the bit-arrays within the sector area will be used to compute the geometrical center. In addition, when the geometrical center is computed, those bits whose distances to the geometrical center beyond three-fifths (3/5) of the foot length are regarded as the noise signal. These bits are then removed from the foot-cluster and the geometrical center is recomputed. This method along with the sectoring technique provides a simple yet effective mechanism to filter out the noise signal.

5) Calculate the Moving Direction and Distance: Based upon the coordinates of the geometrical centers of both feet, we can derive the gravity's coordinate and the moving distance as well as the moving direction can be subsequently computed.

As shown in Fig. 12, assuming that L_1 and R_1 are the coordinates of the geometrical centers for the left foot and right foot, respectively, at time T_1. The coordinate C_1 is then the computed gravity at time T_1. Given that R_2 is the new coordinate of the geometrical center for the right foot at next scan and C_2 is the new coordinate of the gravity. By the Geometry computation, we can derive the vector $\overrightarrow{C_2C_1}$ and thus the moving distance as the value $D = |\overrightarrow{C_2C_1}|$. Similarly, the moving direction is the direction of $\overrightarrow{C_2C_1}$. Notice that, since the user is allowed to rotate to any direction within the OBDP, special care has to be taken to the Quadrant problem when we estimate the moving direction.

6) Algorithm: From the above studies of the gate detecting steps, the gate sensing algorithm can be easily designed. The gait sensing algorithm starts from the collaboration stage to

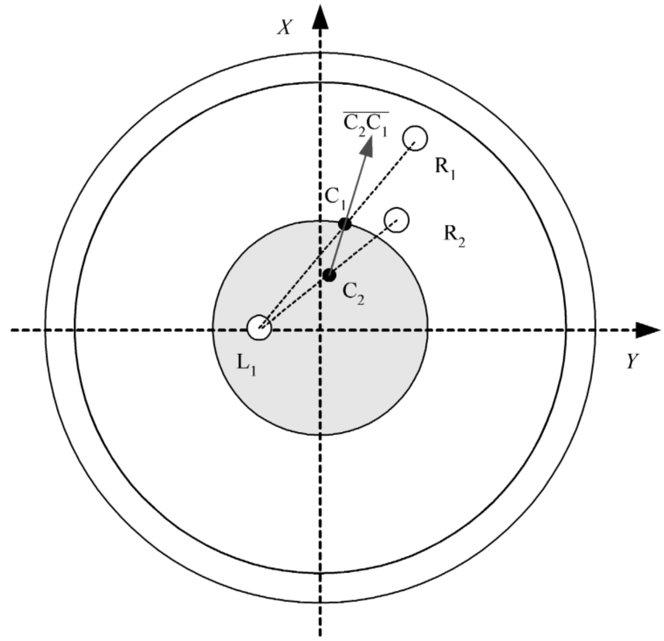


Fig. 12. Computation of the moving distance and direction.

record the sizes of the left foot and right foot, respectively, as well as the foot length. The algorithm is then followed with a continuous loop to detect the gait state and the foot-cluster information to calculate the moving distance and direction. Hence, the complete gait sensing algorithm is as follows:

```
Step 1: Collaborate and record the feet
 size information;
Step 2: Scan the center area to detect if
 the user is in the Stand still state?
Step 3: Repeat Step 2 until the Single
 stance state is detected, then compute
 the foot-cluster in the center area for
 its geometrical center coordinate and
 scan the sliding area for the second
 foot-cluster;
Step 4: If the foot-cluster on the sliding
 area is detected, then the user is en-
 tering the Double stance state;
 Step 4.1: Compute the geometrical center
 of the foot-cluster in the sliding area;
 Step 4.2: Perform the sectoring process
 based upon the computed geometrical
 center;
 Step 4.3: Compute the gravity position
 from the coordinates of the geomet-
 rical centers of the foot-clusters in
 the center area and sliding area, respec-
 tively;
Step 5: Continuously scan the sector area,
 until the Stand still state is detected;
 Step 5.1: Compute the geometrical center
 of the foot-cluster in the sliding area;
```

Step 5.2: Compute the gravity position from the coordinates of the geometrical centers for the foot-clusters in the center area and sliding area, respectively;
Step 5.3: Calculate the moving distance, speed and direction as discussed in Section III-C-V;
Step 6: Go back to Step 2.

D. Locomotion System

The omnidirectional ball-bearing disc is originated from the project of developing an overhead crane training system. The goal of this project is to design a modular and cost-effective interactive visual training device as well as a permit examination instrument for the overhead crane operator. Since such a training system requires tight collaboration among the processes of input processing, 3-D rendering, audio playback, and dynamical computation, the legacy interactive visual simulator often uses a multiprocessor workstation to build its computing system. However, with the evolution of the computer technology, the personal computer has gained more computation power with less cost in the recent years. In addition, the cluster technology enables the researchers to fully explore the parallel computing power over networked personal computers [18]. Hence, an omnidirectional stroll-based virtual reality system which uses the omnidirectional ball-bearing disc as its input device is designed on a cluster of personal computers. This cluster of desktop computers are communicated and synchronized by a Communication Backbone (CB), which is the kernel of the Multiple User Distributed Simulation (MUDS) system [18]. The MUDS system provides a transparent interface among distributed tasks that are run on networked personal computers.

The basic concept of the MUDS system is that each computer of this cluster environment works as a standalone machine so that it only needs to convey message to MUDS kernel without knowing the existence of other computers. Hence, under the MUDS structure, each computer can execute at its own pace and the parallelism among distributed tasks is automatically explored. In order to achieve this goal, a transparent interface, CB, is designed to provide seamless communication among distributed tasks. Each computer in this cluster environment executes CB as its backbone, and tasks of a virtual simulation run on different computers communicate with each other via CB. In other words, the CB acts like an agent for tasks on each computer to communicate and synchronize data among tasks, no matter they are executed on the same computer or not.

Under the MUDS architecture, each task only needs to register to its resident CB on what type of data it is going to produce. The CB will treat this task as one of the publishers of this cluster environment. Similarly, a task may also need to inform CB on what kind of information it requires and CB will treat it as one of the subscribers in this cluster environment. A task may be both 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 seamlessly connect registered tasks to construct this cluster environment. A virtual channel is then constructed between each pair of publisher and subscriber when the initial phase is completed.

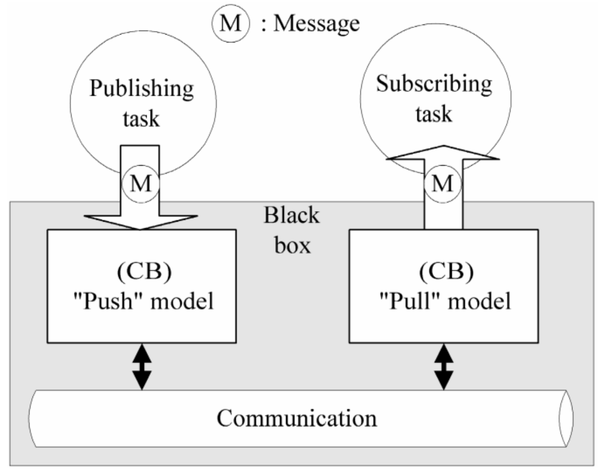


Fig. 13. Push and pull model of the MUDS system.

During the run-time phase, as illustrated in Fig. 13, the publisher will treat CB as a "push" model [19] and pump its data to CB. On the other hand, the subscriber treats CB as the "pull" model that it can dig information out of CB. With this model, a transparent communication among tasks can be easily designed and constructed.

With the help of the CB, the designed locomotion system achieves the goal of modular and cost-effective interactive visual system. As will be presented in Section IV, the overhead crane training system is composed of six modules: the omnidirectional ball-bearing disc, crane control panel, 3-D head tracker, 3-D render, audio player and scene manager. Each module is run as an independent task. When the user maneuvering the virtual world with the omnidirectional disc, the disc controller scans the ball-bearing sensors to collect the data of the user's pace. The collected data is then input to the gait analysis algorithm to simulate his walking posture. The virtual scene is then updated based upon the gait analysis result to respond to the user's motion instantaneously.

IV. Application and Experiment of the OBDP

A. Overhead Crane Training System

The overhead crane is a porterage device that is commonly used in the manufacturing industry. As illustrated in Fig. 14, its main structure is an H-steel frame under the roof and an alternator-driven main body is run on that frame. In addition, there is another motor inside that main body, which controls a lift hook under it to portage cargo. In order to control the lift hook, the overhead crane requires the user to follow and operate the crane on foot. The occupational disaster on the overhead crane often happens due to the insufficient training on handling the hook or controlling the overhand crane. Moreover, the overhead crane training itself is a dangerous process which makes it a perfect case for virtual reality application.

In order to simulate the process of following the overhead crane on foot while controlling the lift hook, the overhead crane training simulator requires an input device to detect the walking

Fig. 14. Overhead crane.

pace and direction of the trainee. The locomotion is a perfect input device to fulfill such a requirement and the omnidirectional ball-bearing disc is then designed to meet such a demand.

In addition, in order to fully immerse the trainee into the simulated scenario, the overhead crane simulator uses the head mount display (HMD) with the 3-D tracker to enable the trainee to observe the virtual scene. Besides, to further provide a realistically simulated environment, the control panel from the actual overhead crane is adapted and refitted as another input device for the trainee. Hence, the overhead crane training simulator is composed of the following six functional modules.

1) The Overhead Crane Control Module: This module is to receive commands from the control panel to change the states of the simulated overhead crane, such as moving the crane left or raising the lift hook. There are six buttons on the control panel, and it is refitted from an actual overhead crane system. These six buttons control the east, west, north, and south movement of the crane, and up and down of the lift hook. In order to increase the flexibility of our system, we use two joystick inputs to simulate these buttons. The game port on the conventional personal computer can support two X-axes and two Y-axes along with four buttons. Since we need to simulate six buttons, an X-axis and a Y-axis are used to simulate the fifth and sixth buttons respectively.

2) OBDP Module: This module receives signals from the ball bearing sensors and executes the gait sensing algorithm discussed in Section III-C. This module interactively computes the maneuvering direction and distance according to the data received from the omnidirectional ball-bearing disc.

3) A 3-D Tracker Module: The main purpose of this module is to allow the user to change his view by moving his head. A crucial factor to build an effective virtual training system is to allow the user to easily observe the virtual scene. The overhead crane training system uses the Polhemus tracker [20] mounted on the HMD to detect the user's head movement. The Polhemus tracker uses the electromagnetic wave to detect the six-degree of motion of the receiver which allows the trainee to have a great flexibility to browse the virtual scene.

4) Sound Module: The sound module is responsible for producing, inside the virtual scene, 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 overhead crane training system uses the Microsoft DirectSound library [21] to implement the sound module. With this sound module, a realistic training scenario can be designed for the trainee.

5) A 3-D Scene Management Module: The function of the 3-D scene management module is to efficiently manipulate the objects inside a virtual scene [22]. The 3-D scene management module is an important factor to increase the performance of a virtual reality system, especially when complex interactions among objects are simulated. In order to fully immerse the user into the virtual environment, the virtual simulation system must be able to effectively simulate various physical phenomena, for example, collision detection among objects, and inertia oscillation of the lift hook.

When the overhead crane or its lift hook is moving in the virtual environment, the 3-D scene management uses the multilevel collision detection mechanism [23] to effectively perceive the collision if there is any. The 3-D scene management 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 3-D scene management module first animates the collision event and then sends messages to the sound module and the rendering module to play a collision sound and display the scene respectively.

In addition, in order to realistically simulate the overhead crane, the 3-D scene management module also computes the inertia oscillation of the lift hook during and after the overhead crane is moving. When the overhead crane is moving, the 3-D scene management 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 overhead crane 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 3-D rendering module to realistically display the physical phenomenon inside the virtual scene. However, constrained by the availability of computation power, the cable is assumed to be an inflexible wire to simplify the computation.

6) A 3-D Rendering Module: The 3-D rendering module is implemented by Microsoft Direct3D library. The Direct3D library automatically builds a tree of objects in the scene when a scene file is imported [24]. However, this scene tree is an exclusive data structure and it is difficult to control an individual object in the scene tree. In order to integrate with the 3-D scene management module, an object table is designed as the interface between the 3-D scene management module and the Direct3D scene tree. Hence, this module displays the virtual scene according to the user's head movement, the gait data from the OBDP and the control panel. Fig. 16 is the snapshot of the scene when the trainee, as shown in Fig. 17, is playing with the con-

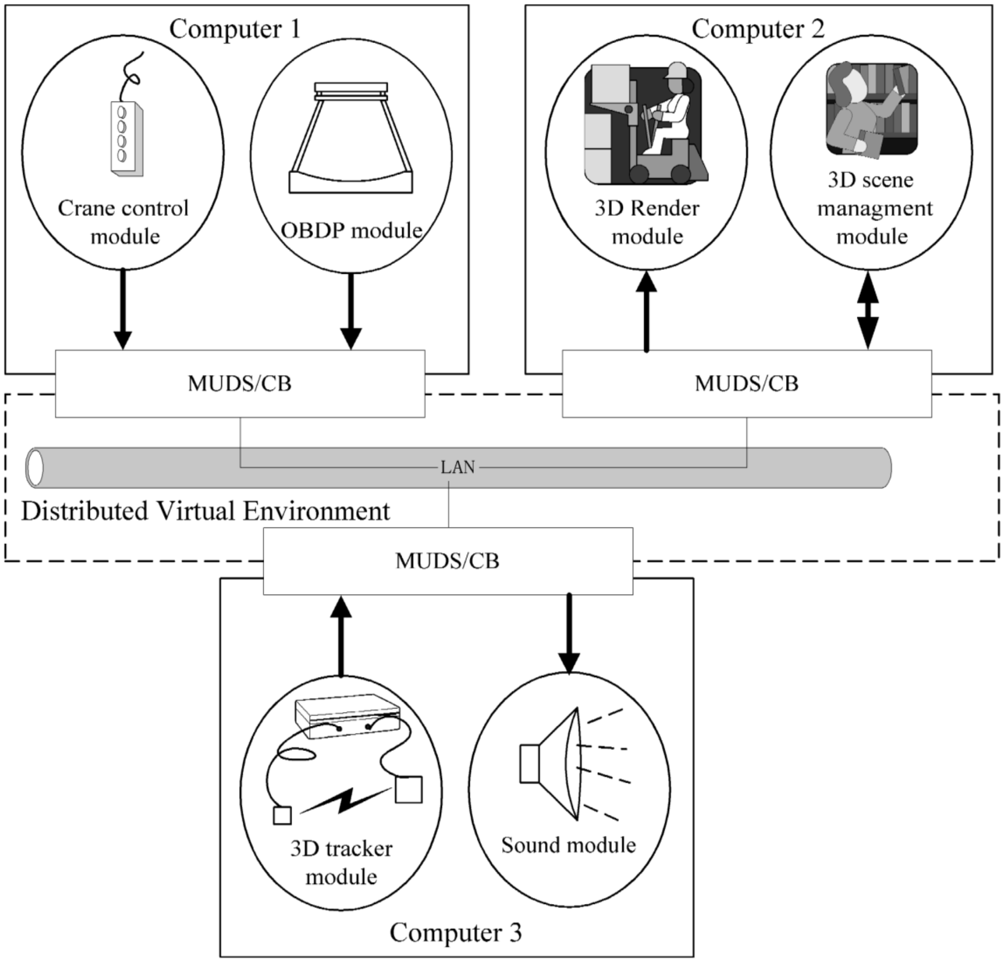


Fig. 15. Architecture of the overhead crane training system.

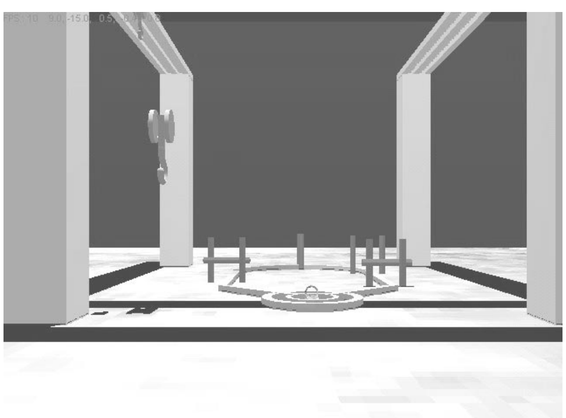


Fig. 16. Snapshot of the virtual scene.

Fig. 17. Trainee is immersed in the overhead crane training system.

trol panel. Fig. 16 is the snapshot of the virtual scene perceived by the player through his HMD while the observer can see the same image from the monitor next to the OBDP system.

In this training scenario, the user operates the control panel to control the lift hook of the overhead crane to clasp a heavy object. He then requires to move the hook and heavy object along the rail to cross the barriers on the rail. During this training process, the trainee needs to stroll along with the overhead crane to precisely control it. The OBDP detects the walking posture of the user and produces locomotion inside the virtual scene.

These six modules are executed on three networked personal computers which constitute a cluster environment for the interactive visual simulation. As illustrated in Fig. 15, the crane control module and the OBDP module is executed on one computer,

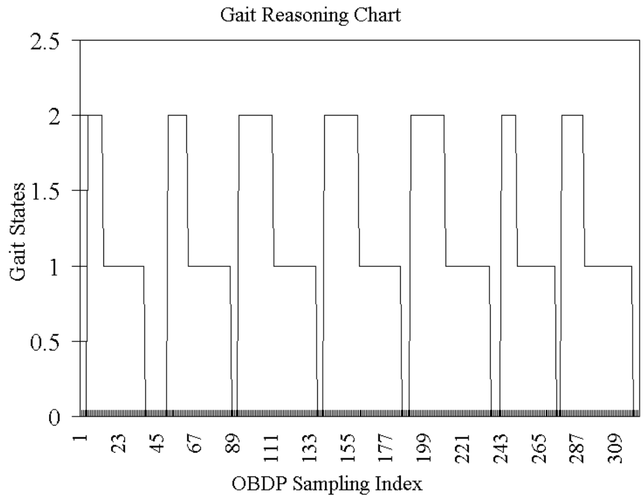


Fig. 18. Chart of the gait reasoning experiment.

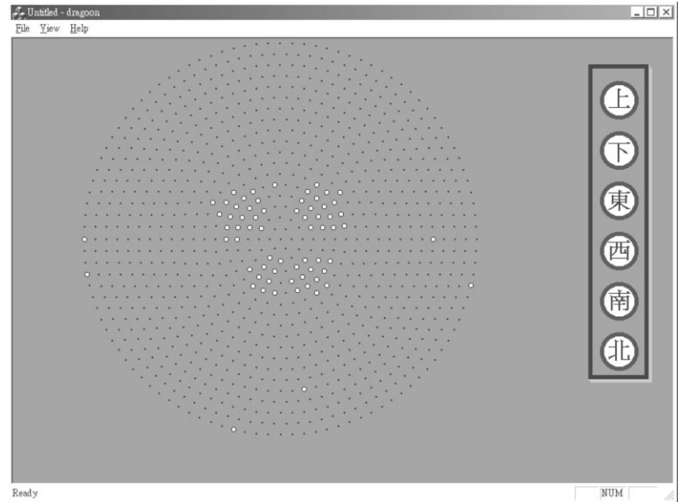


Fig. 19. Snapshot of the stand still state.

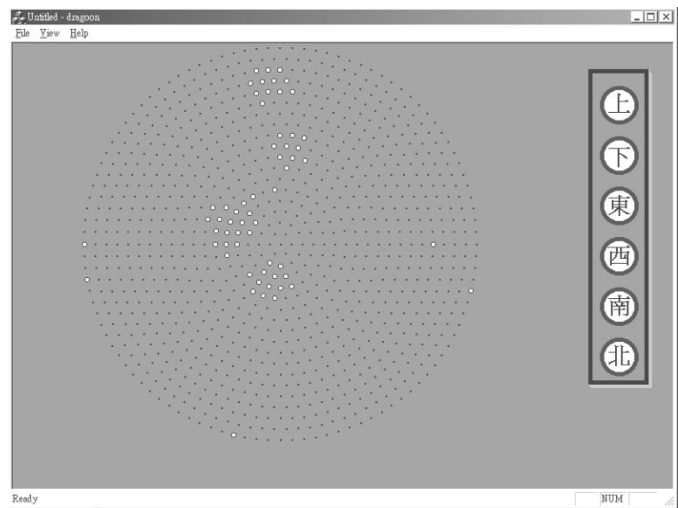


Fig. 20. Snapshot of the double stance state.

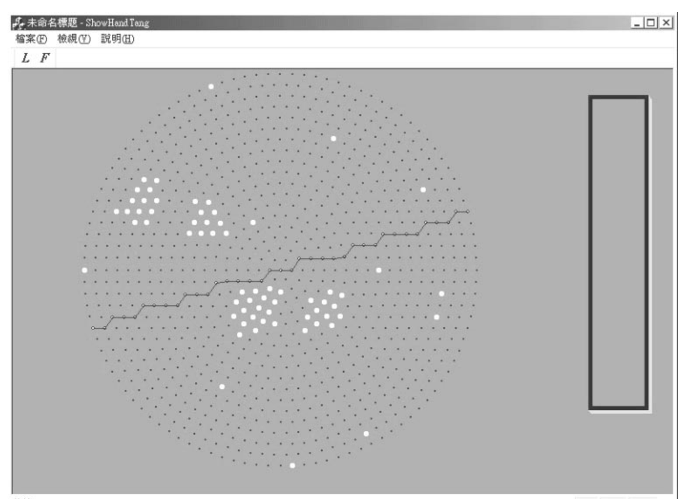


Fig. 21. Snapshot of the raw data from OBDP.

the 3-D rendering module and 3-D scene management module on another computer, and the 3-D tracker module and the sound module on the other computer.

B. Experiments of the Gait Sensing Algorithm

When the user is navigating the virtual world with the ball-bearing disc, the OBDP system scans the ball-bearing sensors to collect the data of the user's motion. The collected data is then input to the gait sensing algorithm to compute the walking status and the pace of the user. Finally, the virtual scene is updated based upon the gait analysis result to respond to the user's motion instantaneously.

Experiment shows that the resulted system reached the frame rate of 17 frames/s. Since the presented locomotion system takes the user's footstep to simulate his gate and thus enables him to navigate the virtual world, the accuracy of the gate sensing algorithm then becomes the crucial factor of the effectiveness of this type of locomotion system. Moreover, the gait reasoning and noise filtering techniques are two essential issues for the gait sensing algorithm. Only if the gait status is successfully detected and the noise signal is correctly filtered, the status of the user can be accurately simulated. Hence, experiments are conducted for gait reasoning and noise filtering methods.

For the experiments of the gait state reasoning, the user is asked to stroll on the OBDP and the gait states are continuously computed and logged into a file for analysis. The logged data are then compared with the state model depicted in Fig. 6 and the result is shown in Fig. 18.

The horizontal axis in Fig. 18 represents the times of sampling the OBDP and the vertical axis is the result of gait state reasoned. The scales 0, 1, and 2 on the vertical axis represent the Stand still, the Double stance, and the Single stance states, respectively. Fig. 19 is the snapshot of the Stand still state as collected from the OBDP and the Stand still state is represented as scale 0 in Fig. 18. Similarly, Fig. 20 is the snapshot of the Double still state as collected from the OBDP and the Double still state is represented as scale 1 in Fig. 18.

Finally, Fig. 21 shows the raw data collected from the OBDP and Fig. 22 is the result after the noise filter technique is applied.

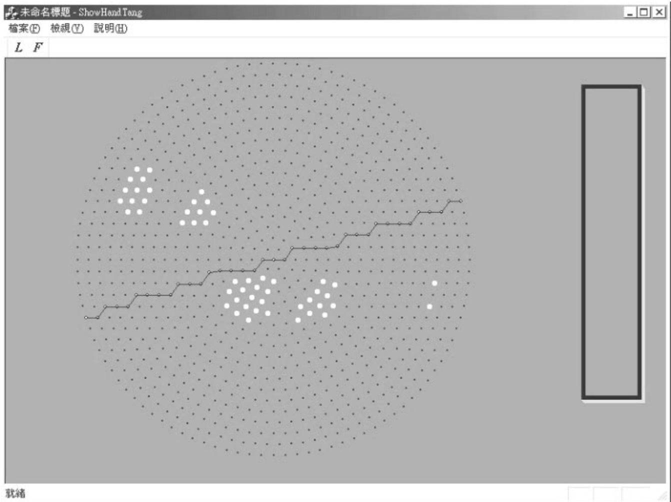


Fig. 22. Snapshot of the result after being filtered.

V. Conclusion and Future Works

As pointed out in [9], locomotion is the most natural way for the user to navigate the virtual environment. Different types of locomotion devices have been developed for the user over these couple years. The OBDP is the first kind of the locomotion interface that does not require any motor to enable the user to roam around the virtual environment. The OBDP is designed based upon the walking posture of the human being and it allows the user to walk on it in a more natural way. The OBDP uses a set of ball-bearing sensors to detect the user's footstep and uses the curvature of the disc surface along with the ball-bearing sensors to slip the user's foot back to the center of the disc. This mechanism makes the OBDP an effective yet the smallest locomotion interface ever been designed. In summary, the contributions of this article are as follows.

- Illustrate the possibility, besides using optical tracking technique, of navigating the virtual world without wearing any tracking device.
- Present a new research on the locomotion interface that can detect the user's foot step.
- Exhibit a gait model to analyze the human walking posture for the Locomotion research.
- Demonstrate a gait sensing algorithm, which enables the user to navigate the virtual world only by his/her footsteps.

In addition, none of the current researches answers the question of what kind of virtual reality application that the locomotion interface fits best. The overhead crane training system is the virtual reality system that fully explores the advantage of the locomotion device in the virtual reality research. The overhead crane is the most frequently used but dangerous porterage machinery in the manufacturing industry. To operate an overhead crane, the user has to follow it which makes the overhead crane training a perfect case to build a locomotion-based virtual environment. Although the overhead crane training system presented in this article is a prototype system, it shows a promising impact on the locomotion interface in virtual reality application. Further study is to improve this training system, so that the trainee can be fully immersed in the training environment.

Future researches on the OBDP will be conducted in the following directions. First, the orbiting frame on the user's waist will put some constraint on user's motion. For example, the user cannot jog on the OBDP. Secondly, the ball-bearing position sensor currently used on the OBDP is a custom-made sensor that is difficult to be maintained and installed. A new type of position sensor is required to increase the sensitivity and ease of maintenance.

Finally, the gait analysis algorithm presented in this article can not detect the sidestepping at this moment. Further improvement of the gait analysis algorithm is currently under investigation so that the OBDP can detect the walking direction as well as sidestep.

Acknowledgment

The purpose of this overhead crane system project is to design a training system that the Council of Labor Affairs can actually use for training courses and permit examinations in the near future.

References

[1] W. V. Tucker and L. D. Stuckey, Jr, "The roles of modeling and simulation at boeing," *Trans. Soc. Comput. Simul. Int.*, vol. 1, pp. 3–9, Jan. 1998.

[2] J. Kuhl, D. Evans, Y. Papelis, R. Romano, and G. Watson, "The Iowa driving simulator—An immersive research environment," *IEEE Computer*, vol. 28, pp. 35–41, July 1995.

[3] M. R. Stytz, "Distributed virtual environments," *IEEE Comput. Graph. Applicat.*, vol. 16, pp. 19–31, May 1996.

[4] J. Y. Huang and H. H. Bai, "Experience of building a high-fidelity mobile crane simulator with cluster of desktop computers," in *Proc. IEEE ICDCS-21W*, 2001, pp. 402–407.

[5] F. P. Brooks, Jr, "Walkthrough—A dynamic graphics system for simulating virtual building," in *Proc. Workshop on Interactive 3D Graphics*, 1986, pp. 9–21.

[6] R. P. Darken, W. R. Cockayne, and D. Carmein, "The omni-directional treadmill: A locomotion device for virtual worlds," in *Proc. ACM UIST*, 1997, pp. 213–221.

[7] H. Iwata, "Walking about virtual environments on an infinite floor," *Proc. IEEE Virtual Reality*, pp. 286–293, 1999.

[8] D. R. Pratt *et al.*, "Insertion of an articulated human into a networked virtual environment," in *Proc. 1994 AI, Simulation, and Planning in High Autonomy Systems Conf.*, 1994, pp. 34–45.

[9] M. Usoh *et al.*, "Walking $>$ walking-in-place $>$ flying in virtual environments," in *Proc. SIGGRAPH '99*, 1999, pp. 359–364.

[10] R. Christensen, J. M. Hollerbach, Y. Xu, and S. Meek. Inertial force feedback for the Treadport locomotion interface. presented at Presence: Teleoperators and Virtual Environments. [Online]. Available: http://www.cs.utah.edu/~jmh

[11] H. Yano, H. Iwata, H. Noma, and T. Miyasato, "Shared walk environment using locomotion interfaces," in *Proc. ACM Conf. Computer Supported Cooperative Work*, 2000, pp. 163–170.

[12] S. S. Intille, J. W. Davis, and A. F. Bobick, "Real-time closed-world tracking," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1996, pp. 607–703.

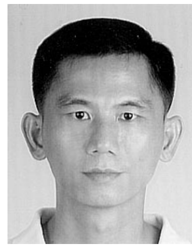
[13] M. Ward, R. Azuma, R. Bennett, S. Gottschalk, and H. Fuchs, "A demonstrated optical tracker with scalable work area for head-mounted display systems," in *Proc. 1992 Symp. Interactive 3D Graphics*, 1992, pp. 43–52.

[14] M. Slater, A. Steed, and M. Usoh, "The virtual treadmill: A naturalistic metaphor for navigation in immersive virtual environments," in *Proc. First Eurographics Workshop on Virtual Reality*, 1993, pp. 71–83.

[15] H. Iwata and T. Fuji, "Virtual perambulator: A novel interface device for locomotion in virtual environment," in *Proc. VRAIS'96*, 1996, pp. 60–65.

[16] J. Y. Huang *et al.*, "The gait sensing disc—A compact locomotion device for the virtual environment," in *Proc. WSCG2000*, 2000, pp. 290–297.

[17] E. Y. Chao *et al.*, "Normative data of knee joint motion and ground reaction forces in adult level walking," *J. Biomechanics*, vol. 16, no. 3, pp. 219–233, 1983.
[18] J. Y. Huang, C. T. Fang-Tson, S. J. Wang, and W. C. Wang, "A model and design of a fully distributed computing environment for virtual reality," *Proc. IEEE RTCSA*, pp. 160–168, 1997.
[19] S. Srinivasan and P. Reynold, "Communications, data distribution and others goodies in the HLA performance model," in *Proc. 1997 Spring SIW*, 1997, pp. 329–338.
[20] [Online]. Available: http://www.polhemus.com/isotrkds.htm
[21] [Online]. Available: http://www.msdn.microsoft.com/library/en-us/directx9_m/directx/sound/directsound.asp
[22] J. Y. Huang and J. L. Lin, "Design of 3D Real Time Database," in *Proc. 1996 Computer Graphics Workshop*, 1996, pp. 81–83.
[23] M. Moore and J. Wilhelms, "Collision detection and response for computer animation," in *Proc. 15th Annu. Conf. Computer Graphics*, 1988, pp. 289–298.
[24] [Online]. Available: http://www.microsoft.com/library/en-us/wced3d/htm/_cxconDirect3D.asp



Jiung-Yao Huang (M'90) received the B.S. degree in 1983 from the Department of Applied Mathematics at Chung-Hsing University, Taiwan, R.O.C., the M.S. degree in computer science from Tsing-Hua University, Taiwan, in 1988, and the Ph.D. degree in electrical and computer engineering from the University of Massachusetts, Amherst, in 1993.

He is currently a Professor in the Department of Computer Science and Information Engineering at Tamkang University, Taipei, Taiwan, R.O.C. He was the Director of the Advanced Digital Research Laboratory(ADSL), Information Processing Center, Tamkang University, Taiwan. In addition, he also serves as the Executive Editor of the *Tamkang Journal of Science and Engineering*. His research interests include computer graphics, networked virtual reality, distributed systems, and multimedia systems.

Dr. Huang has been a member of ACM since 1990 and joined the Society for Computer Simulation in 1996.