A DESIGN OF THE GAIT SENSING ALGORITHM FOR THE LOCOMOTION-BASED VIRTUAL REALITY SYSTEM

*Jiung-yao Huang、Kuei-ping Shih、Yung-ting Lin
Department of Information Engineering, Tamkang University, Tamsui 251, Taiwan
*E-Mail：jhuang@mail.tku.edu.tw

Chi-fu Tai and +Chung-yun Gau
Institute of Occupational Safety and Health,
Council of Labor Affairs, Executive Yuan, Taiwan
+E-Mail：gau@mail.cla.gov.tw

ABSTRACT

This paper present an algorithm to enable the user to navigate the virtual world by walking. The presented algorithm detect the user's moving distance and direction by sensing his walking paces only. With this algorithm, the user can freely navigate the virtual environment by locomotion without wearing any tracking devices to detect his motion. To achieve this goal, a gait model, that analyzes the states of the walking posture, is presented in this paper. Compared to other researches on the locomotion devices, the gait sensing algorithm presented in this paper provides a more natural human-machine mechanism to navigate the virtual world. The algorithm presented in this paper is based upon a locomotion mechanism called Omni-direction Ball-bearing Disc Platform(OBDP). The OBDP is a round shape disc laid with arrays of ball-bearing point-sensors to detect the user's footstep. At the end, this paper presents a prototype of an overhead crane simulator that is designed using the presented gait sensing algorithm to allow the user to navigate the virtual world.

1. INTRODUCTION

The research of the virtual reality is to study the man-machine interface for the user to fully immerse into the simulated world to accomplish the designated task. The locomotion interface is a result of such efforts. 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. The locomotion device provides a natural way for the user to maneuver the virtual environment on foot. Different locomotion devices have been developed over the years, including treadmill, pedaling device and motion capture.[1]

The treadmill, which was originally used for the fitness, is the simplest way to build a locomotion device for the virtual environment maneuvering. UNC[2] developed the first locomotion device in 1986 to study 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 the bicycle, to enable the user to change his walking direction.

To accurately simulate the human pedestrian behavior, Utah University designed another type of treadmill, Treadport[3], to provide the inertia force feedback to the user. One of the major problems of design the locomotion interface is that, it can not accurately simulate the walking behavior which hinders the user to fully immersed into the virtual space. For example, when the human walking on an uphill, gravity and inertia will act on body and ankle to counteract his forward motion. This situation is only one of the examples that diverge the real world walking from the virtual space locomotion.

To revoke the restriction on the freedom of navigation that is imposed on previous treadmill systems, the Naval Postgraduate school proposed an omni-directional treadmill(ODT)[4] 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 his moving 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[5], which is built by the Institute of Engineering Mechanics, University of Tsukuba, is an enhancement of the omni-directional treadmill. The Torus treadmill employs twelve treadmills moving the walker in 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 six degree-of-freedom 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.

The UniPort[6] is a variation of the legacy treadmill-based locomotion device. The UniPort is very similar to a unicycle and it is a bipedal locomotion device. The user pedals to simulate walking or running. The moving direction is controlled by the seat of the UniPort. That is, the user uses the waist and thigh to change the direction of locomotion.

This paper presents a completely new type of locomotion device, called 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 3D tracker, arrays of ball-bearing sensors on a disc are used to detect the pace and an orbiting frame to identify the walking direction. 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[4][5], the OBDP is the locomotion interface that is more conform with the kinesiology of the human. Stem from the specific feature of the OBDP, a gait sensing algorithm is designed and presented in this paper. In addition, the gait model of the human walking posture is also proposed in this paper. With the algorithm presented in this paper, the user can freely walk on the OBDP to roam inside the virtual world without physically leaving the platform.

2. THE OMNI-DIRECTIONAL BALL-BEARING DISC PLATFORM(OBDP)

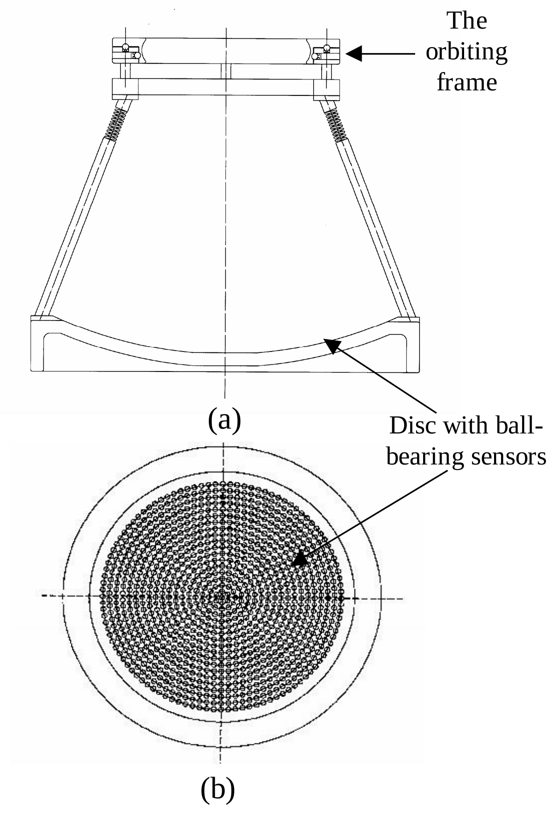

Figure 1. The omni-directional ball-bearing disc locomotion device; (a) Side view; (b) The surface of disc

The OBDP is a round-shaped disc with ball-bearing point-sensors scatter over its surface. Compared with other locomotion devices, the OBDP is unique in terms that no motor is used to provide omni-directional maneuvering. Instead, as shown in Figure 1, OBDP uses arrays of ball-bearing sensors to detect the user pace and support omni-directional navigation.

The OBDP is designed by Division of Occupational Safety, which belongs to Institute of Occupational Safety and Health, Council of Labor Affairs, Executive Yuan[7], Taiwan. 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 ease 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 the user when he is walking. Furthermore, the supporting frame can simulate the friction between the foot and the ground by counteracting a force to the body which then force the foot to slip on the surface of the OBDP.

The surface of the OBDP, as shown in Figure 1(b), is laid with arrays of ball-bearing point-sensors in rings. In addition, the surface of the OBDP was designed based upon the natural gait of the human being. While walking, the human will either swing one leg on the air with the other on the ground or have both feet touch the ground. The arc shape design in 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 OBDP. 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.

The OBDP uses a set of A/D and D/A interface cards to translate the signals of the point-sensors into the computer to analyze the gait. The surface of the OBDP 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 entire point-sensors and analyze the gait within 10 ms to realistically reflect the walking posture of the user.

3. THE GAIT SENSING ALGORITHM

The goal of this paper is to present an algorithm to recognize the user's footstep from the signals of point-sensors 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.

3.1 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 the natural motion of the human being. However, it requires complex coordination among skeleton, muscle and nervous system 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 objectivity and 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[8] 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 Figure 2, is cited from the Medical Science. The arguments from this definition for the Temporal Distance can effectively help us to model the human walking status.

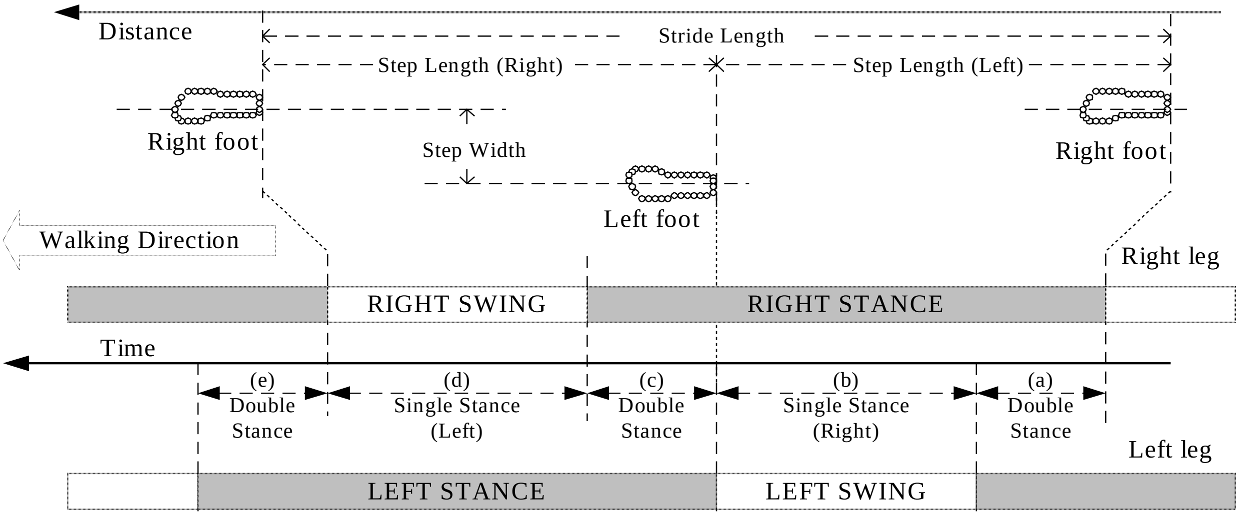

Figure 2. The definition of Temporal Distance of the walking posture

As shown in Figure 2, 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 leg as **the swing leg**. Hence, it is obvious that the human walking posture is composed of a series of leg exchange between the swing leg and the stance leg.

3.2 The Analysis of Walking Posture

As presented in the previous section, we can calculate the human’s walking direction, speed and distance by continuously detecting the position of both feet. That is, from Figure 2, it is clear that the walking distance can be derived from the step length. The step length is measured as the distance between two feet when both feet are touching the ground and are not side by side, as illustrated in Figure 2. The walking speed is the division of 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 right leg. Hence, the gait sensing algorithm does not need to distinguish the left foot from the right one. For example, given the following two cases:

Case 1: If the user is facing the North and he swings his right leg forward.

Case 2: If 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 leg 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 Figure 3.

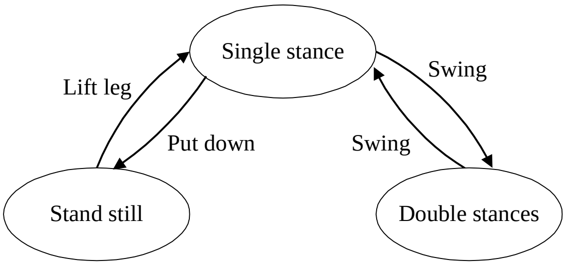

Figure 3. The state diagram for the gait

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 his 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.

3.3 The 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 model has been slightly modified. Since the OBDP always constraints the user in the center of the platform, the user can not shift his gravity and swing his other leg. Instead, the original swing leg slides back to the center of the platform as shown in Figure 4.

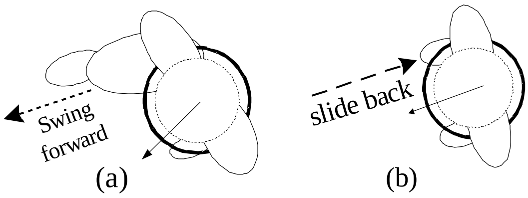

Figure 4. The walking posture on the OBDP; (a) one leg is swing forward; (b). the swung leg slide back.

To match up 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, as shown in Figure 5. 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 siding 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 is then continuously probing the sliding area to decide the distance between two feet until the swung foot slides back to the center area.

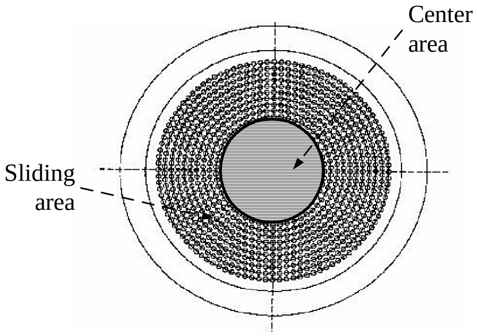

Figure 5. The center area and the sliding area of the surface of the OBDP

Hence, we can modify the state diagram in Figure 3 into the gait model as shown in Figure 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 sate. 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.

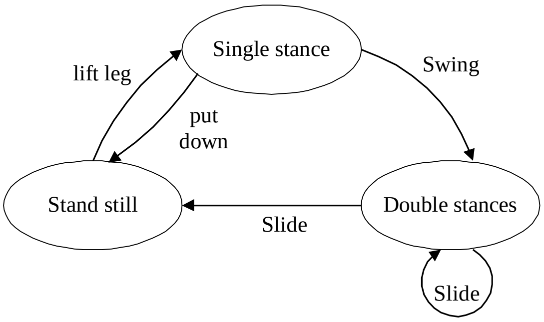

Figure 6. The gait model for OBDP

3.4 The Steps of detecting the Gait

Based upon the state diagram in Figure 6 and the feature of the OBDP, we know that the gait sensing mechanism should include the following functional modules and they are fully discussed in the following sub-sections.

3.4.1 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 signal from the point-sensors, which are meaningless unless the sizes of both feet are known. Hence, the user is requested to stand still in the center area at the initialization stage. By counting the bits clusters on the center area, the size of two feet are detected and recorded. We call this bit-clusters for two feet as **the feet-clusters**. Similarly, the bit-cluster of a single foot is called **the foot-cluster**. This sizing information is then used to detect the walking state of the user as modeled in Figure 6. Notice that we assume the tolerance of the sizing difference is 10%. That is, the size of the foot-cluster may be 10% smaller or 10% 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 algorithm is then continuously scan the sliding area to detect the foot-cluster. When the foot-cluster is detected in the sliding area, the user then enters the Double stance state as illustrated in Figure 6. Figure 7 illustrate the double stance state where, except the foot-cluster in the center area, there exists another foot-cluster in the sliding area.

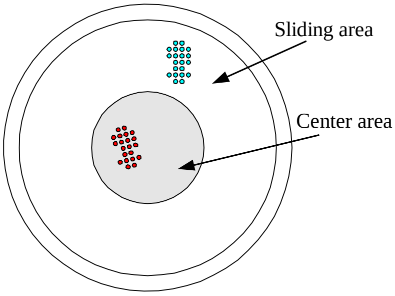

Figure 7. Gait state recognition

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 90% 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.

3.4.2 The Geometrical Center for 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 the 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 position of two feet.

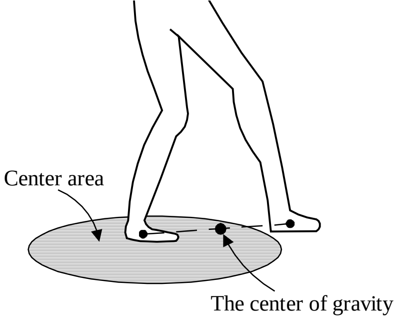

Figure 8. Simulating the gravity point of walking

As illustrated in Figure 8, we first calculate, respectively, the geometrical center points of the left and right feet, and then simulate the gravity point as the middle point between these two geometrical center points.

Further, we use the foot-cluster to calculate the geometrical center of each foot. For each foot-cluster, we sum up 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 Figure 9.

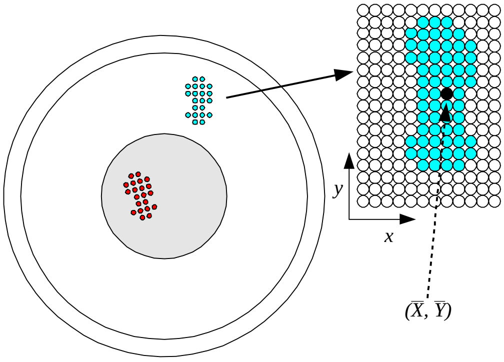

Figure 9. Computation of the geometrical center

3.4.3 Speed up the state recognition by the sectoring technique

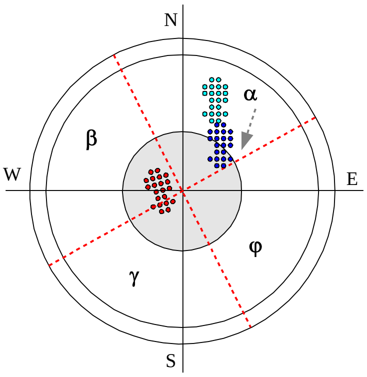

Figure 10. Speeding up the state recognition by sectoring

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 Figure 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 with the dashed line as the midline of one of the sectors, which is sector α in Figure 10. Under the normal walking posture, the foot in the sliding area will slide back to the center area within the sector area α only. Hence, we can speed up the gait state reasoning process by scanning the sector area α only, until the

Stand still state is reached.

3.4.4 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 Figure 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.

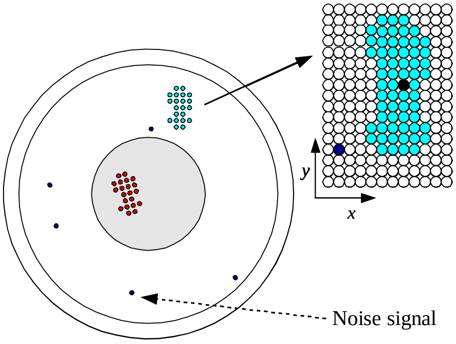

Figure 11. Noise signal on the OBDP

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 distance to the geometrical center is beyond three-fifths 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 provide a simple yet effective mechanism to filter out the noise signal.

3.4.5 Calculate the moving direction and distance

Based upon the coordinates of the geometrical centers of both feet computed in Section 3.4.2, we can derive the gravity's coordinate and the moving distance as well as the moving direction can be subsequently computed. As shown in Figure 12, assuming that L_1 and R_1 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.

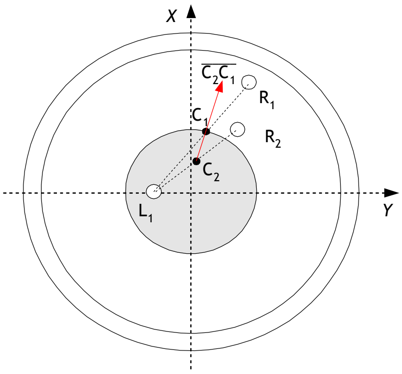

Figure 12. Computation of the moving distance and direction

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.

3.5 The Gait Sensing Algorithm

The Gait Sensing algorithm starts with the collaboration stage as discussed in Section 3.4.1 to record the sizes of the left foot and right foot, respectively, as well as the foot length. A continuous loop is then followed 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 entering 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 geometrical centers of the foot-clusters in the center area and sliding area, respectively;
Step5: 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 3.4.5.
Step 6: Go back to Step 2.
```

4. THE IMPLEMENTATION AND ITS APPLICATION TO THE OVERHEAD CRANE SIMULATOR

The Gait Sensing algorithm presented in this paper was implemented to design an overhead crane simulator. The overhead crane is a porterage device that is commonly used in the manufacturing industry. Its main structure is an I-steel frame under the roof and an alternator-driven main body running on that frame. In addition, there is another motor inside that main body, which controls a lift hook under it to porterage cargo. 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. Since the overhead crane requires the user to follow the lift hook while controlling it, the OBDP system provides an ideal vehicle to design such an interactive visual training system.

Notice that the main goal of the gait sensing mechanism is to detect the moving distance and direction of the user, a six-degrees of position tracker must be used to detect the user's head motion if the support of view change is required. Hence, the overhead crane training system uses the 3D tracker and head mount display(HMD) to enable the trainee to observe the virtual scene. Besides, in order to design a realistic training system, the control panel from the actual overhead crane is adapted and refitted as an input device for the simulator. The overhead crane training simulator is composed of the following six functional modules:

- The overhead crane control module to receive commands from the control panel to change the states of the simulated overhead crane.
- The Gait sensing module to detect and analyze the user's gait;
- A 3D scene management module to calculates and manages the collision of the lift hook with barriers and cargo.
- A 3D rendering module to displays the virtual scene and is implemented by Microsoft Direct 3D library.
- A 3D tracker module to allow the user to change view by moving his head.
- A sound module to produce, 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.

The overhead crane training system is run on three networked personal computers which constitute a virtual parallel computing environment for the interactive visual simulation. These networked computers are communicated and synchronized by a Communication Backbone(CB), which is the kernel of the Multiple User Distributed Simulation(MUDS) system.[9] The CB is a transparent layer among distributed tasks that are run on networked desktop computers. Each computer in this virtual parallel environment executes CB as its backbone, and tasks of a virtual simulation run on different computers communicate with each other via CB. 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.

As illustrated in Figure 13, the crane control module and the OBDP module are run on one computer, the 3D rendering module and 3D scene management module on the second computer, and the 3D tracker module and the sound module on the last computer.

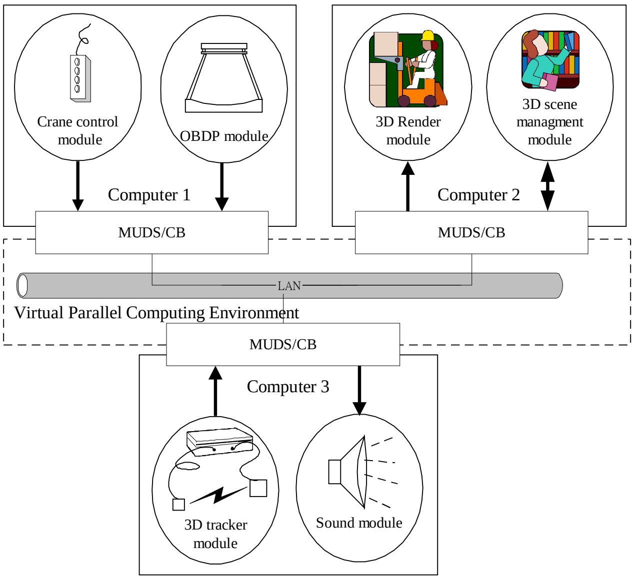

Figure 13. The architecture of the overhead crane training system

Figure 14 is the snapshot of the scene when the trainee, as shown in Figure 15, is playing with the control panel. In Figure 15, the player perceives the virtual scene through his HMD while the observer can see the same image from the monitor next to the OBDP system.

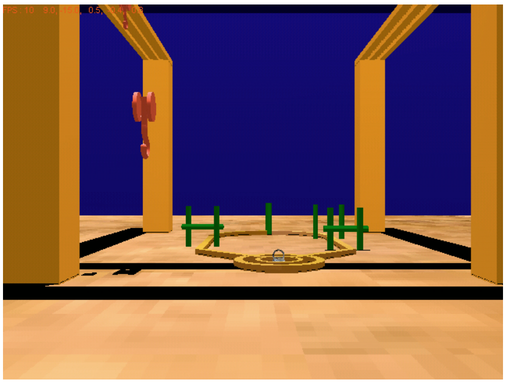

Figure 14. The snapshot of the virtual scene

Figure 15. A trainee is immersed in the overhead crane training system

5. CONCLUSION AND FUTURE WORKS

The main contribution of this paper is to provide a new research direction on designing a mechanism to free the user on navigating the virtual world. The OBDP is the first kind of locomotion device that does not require any position tracker to detect the user's motion as well as any motor to enable the user to roam around the virtual environment. Based upon the special feature of the OBDP, a gait sensing algorithm is designed and presented in this paper. This paper present a gait model to analyze the human walking posture for the Locomotion research. In addition, a gait sensing algorithm that is based upon the gait model is designed and implemented. Most importantly, this paper presents a prototype of an overhead crane simulator that employs the gait sensing algorithm to provide locomotion within the virtual environment.

Although the presented algorithm was implemented and used as part of a overhead crane simulator, more studies are required to verify the accuracy and performance of the gait sensing algorithm. The experiments, including testing the stability, verify the performance, and validate the accuracy of moving distance and direction, are currently under investigation. Noise filtering is another important issue that requires further study. Since the point-sensor used in the OBDP is a pressure-based ball-bearing switch, dust and rust as well as malfunction of the spring are unavoidable situations. Those undesirable situations will induce the noise signal and affect the accuracy of the algorithm. Further study on designing an efficient noise diagnostic algorithm may be required, without the user intervene, if the presented noise signal is over an acceptable range.

ACKNOWLEDGEMENT

This research is founded by Institute of Occupational Safety and Health, Council of Labor Affairs, Executive Yuan, under the contract number: IOSH89-S131.

REFERENCES

[1] S. S. Intille, J. W. Davis and A. F. Bobick, "Real-Time Closed-World Tracking", IEEE Conference on Computer Vision and Pattern Recognition, November 1996, pp.607-703.

[2] F. P. Brooks Jr., "A dynamic graphics system for simulating virtual building", Proceedings of the 1986 Workshop on Interactive 3D Graphics, October 1986, pp. 9-21.

[3] R. R. Christensen et al., "Inertial Force Feedback for a Locomotion Interface", , Presence: Teleoperators and Virtual Environments, Vol 9, pp. 1-14, 2000. Also available at http://www.cs.utah.edu/~jmh

[4] R. P. Darken , et al, "The Omni-directional treadmill : a locomotion device for virtual worlds", Proceedings of the 10th annual ACM symposium on user interface software and technology, 1997, pp. 213-221.

[5] H. Iwata, "Walking About Virtual Environments on an Infinite Floor", Proceedings IEEE Virtual Reality, March 1999, pp. 286-293.

[6] D. R. Pratt et al., "Insertion of an Articulated Human into a Networked Virtual Environment", Proceedings of the 1994 AI, Simulation, and Planning in High Autonomy Systems Conference, University of Florida, Gainesville, December, 1994. Available at http://www.npsnet.org/~npsnet/publications.html

[7] J.Y. Huang, W.H. Chiu, Y.T. Lin, M.T. Tsai, H.H. Bai, C.F. Tai and C.Y. Gau, H.T. Lee, "The gait sensing disc -- a compact locomotion device for the virtual environment", The 8-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media'2000 (WSCG 2000), Volume II, Feb. 7-11, 2000, pp.290-297.

[8] E.Y. Chao, et al., "Normative Data of Knee Joint Motion and Ground Reaction Forces in Adult Level Walking" Journal of biomechanics, Vol 16,No 3, pp 219-233, 1983

[9] 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", IEEE Real-Time Computing Systems and Applications (RTCSA), October 1997, pp.160-168.