Camera-based OBDP Locomotion System

Minghadi Suryajaya*
School of Mining Engineering, University of NSW

Tim Lambert
School of Computer Science and Engineering, University of NSW

Chris Fowler
School of Mining Engineering, University of NSW
Sydney, 2052, Australia

Abstract

In virtual reality, locomotion is a key factor in making a simulation immersive. Actually walking is the most intuitive way for people to move about, providing a better sense of presence than walking-in-place or flying [Usoh et al. 1999]. We have built a locomotion system with a ball-bearing platform that allows the user to walk in a natural fashion in any direction. The user's leg motion is tracked with two cameras and turned into locomotion in the simulation. We also track upper body motion and use this to animate the user's avatar.

Our approach is less expensive than systems that involve complex mechanical arrangements, such as an omnidirectional treadmill [Darken et al. 1997], and more immersive than simple switch mechanisms such as the Walking-Pad [Bouguila et al. 2004]. Our system delivers real-time performance on mid-tier hardware computer and webcams.

CR Categories: H.5.1 [Information Systems]: Information Interfaces and Presentation—Artificial, augmented, and virtual realities; I.2.10 [Computing Methodologies]: Artificial Intelligence—Vision and Scene Understanding3D/stereo scene analysis I.3.7 [Computing Methodologies]: Computer Graphics—Three-Dimensional Graphics and RealismVirtual Reality

Keywords: locomotion interface, walking, stereo camera, mean-shift algorithm, virtual reality

1 Introduction

In virtual reality, locomotion is a key factor in making a simulation immersive. Usoh et al. [1999] found that real walking produced a substantially higher sense of presense than virtual walking. Zanbaka et al. [2005] found that real walking increases learning and understanding of an environment.

The primary objective of our research is developing a locomotion device to improve the user's sense of walking in the virtual environment. We also want a system that is robust and can be deployed in an actual training simulation.

An omni-directional ball-bearing disc platform (OBDP) provides a locomotion platform that conforms to the kinesiology of the human body [Huang 2003]. Huang [2003] demonstrated the advantages of an OBDP interface in a training simulator. However the device employed hundreds of custom-made ball bearing sensors, which are expensive and difficult to install and maintain.

*e-mail:minghadi@cse.unsw.edu.au

Our system uses an OBDP, but employs standard ball transfer units [Blackwood 2007] rather than custom-made ones. Instead of ball-bearing sensors, we use computer vision to track user motion, resulting in a robust and relatively inexpensive locomotion system.

2 Previous Work

A linear treadmill has been used in a number of working simulators. Brooks [1986] used a steering bar for changing direction in a treadmill setup. Noma and Miyasato [1998] employed a series of linear actuators underneath the treadmill belt for simulating the slope of virtual ground. Christensen [1998] used a large manipulator connected to the walker to provide a sense of gravity. Finally, the Sarcos treadport [Hollerbach et al. 2000] combined a tilting treadmill, a mechanical tether and a CAVE-like visual display.

Omni-directional treadmills have also been used. Darken et al. [1997] developed an early version which uses two perpendicular treadmills woven together in a mechanical fabric. Iwata [1999] improved upon this, developing the Torus treadmill. It employs twelve treadmills mounted on two orthogonal rails actuated by four chains. It reduced the noise level and improved safety.

Iwata et al. [2001, 2005, 2006] developed several systems using a motorized platform that followed foot movement, but could not achieve sufficient walking speed due to difficulties in tracking foot movement [Iwata et al. 2007]. Iwata et al. [2007] improved the walking speed with strings attached to shoes with low friction soles. The system pulls on the strings to keep the user walking in place.

The Omni-direction Ball-bearing disc platform (OBDP) [Huang 2003] uses 975 custom-made ball bearing sensors embedded on a disc to detect the user's feet. The curvature of the disc and the ball-bearings allows the user's foot to slip back to the center of the disc.

There are other passive locomotion devices, where no motor is required. The Sarcos Uniport [Darken et al. 1997] is a unicycle-like pedaling device for a battlefield simulator project where direction of motion is controlled by twisting of the user's waist and thighs. The Virtual Perambulator [Iwata and Fuji 1996] uses roller skates equipped with four casters and magnetic sensors to measure the location of the feet. The drawbacks of such device is that the users need to slide their feet by themselves.

3 System Description

3.1 The Omnidirectional Stroll-based Platform

We have improved the original OBDP to make it more robust, maintainable, and practical to use in day-to-day deployment.

Instead of custom-made ball-bearing rollers, we used the widely-available ball-transfer conveyor [Blackwood 2007]. These conveyors are often used for transferring heavy goods between conveyor lines. Each can support a maximum capacity of 45kg in a 36cm^2 space. A foot size of 20cm $\times$ 8cm dimension can be supported by

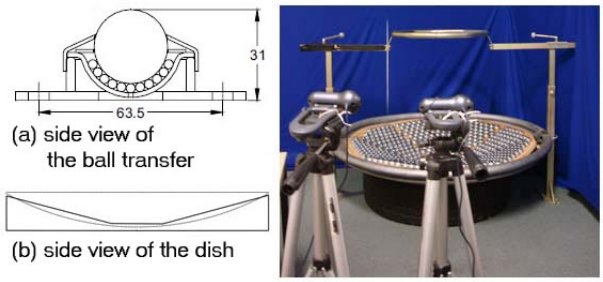

Figure 1: *Dish platform*

a minimum of four of these ball-bearings. It can support a person with a maximum weight of 180kg standing on one foot.

We have also modified the curvature of the dish. Testing showed that a height-proportional hemispherical platform (that is where the radius is the distance between waist and foot) does not provide sufficient inclination for the feet to slide naturally. A flat inclination angle was shown to work much better since it provides better footing for the feet to slide. Therefore, we use a flat 15 degree inclination angle. We also placed a rubber support under the dish to provide more comfortable dampening force.

Instead of using a rotatable orbiting frame, we used a non-rotatable frame. Disabling rotation allowed us to use a smaller number of cameras, reducing computation power necessary for tracking. Change of direction is available via gesture recognition. The orbiting frame is adjustable to the user's height.

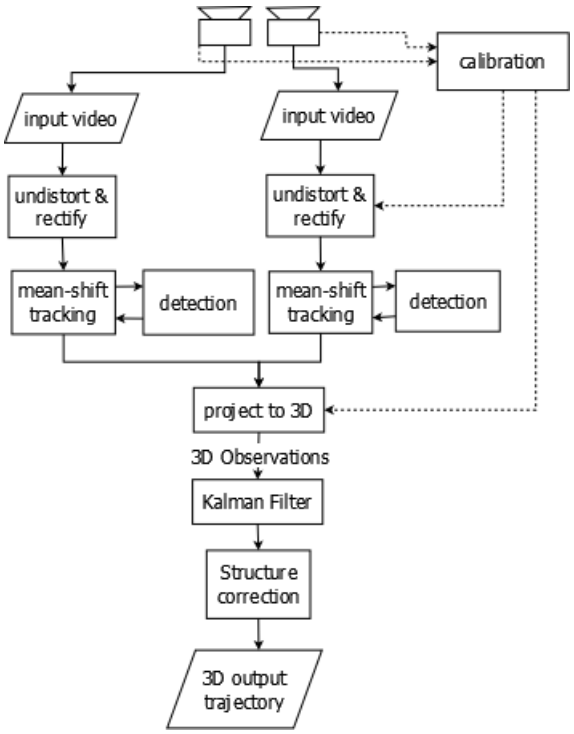

Figure 2: *System Flow Diagram*

3.2 Mean-shift Tracking and Detection of Markers

The mean-shift algorithm is often used when the motion of a tracked object can not be described by a motion model. It was originally proposed by Fukunaga [1975] and was further developed by Cheng [1995] and Comaniciu [2000].

The basic idea of mean-shift is finding the nearest peak of a probability distribution input vector in a multivariate kernel density function. OpenCV [Bradski and Kaehler 2008] implements this as calculation of the center of mass of the image pixel distribution $x_c = \frac{M_{10}}{M_{00}}$ where $M_{10} = \sum_x \sum_y xI(x,y)$, $M_{00} = \sum_x \sum_y I(x,y)$, and similarly for y_c.

Histogram backprojection is commonly used to generate probability vector input to be fed to the mean-shift algorithm [Bradski and Kaehler 2008]. Histogram backprojection generates an intensity image from the hue channel based on similarity to a color histogram. A detailed description on histogram backprojection can be found in Swain and Ballard [1990].

In HSV color space, hue is ill-defined at low saturation and low brightness [Bradski 1998]. Therefore we built an adaptive mechanism in our tracker to filter out pixels with low saturation and value. The threshold is determined heuristically during the initialization phase. The combined result of the histogram backprojection and saturation-value thresholding for two of the red markers is illustrated in Fig. 3.

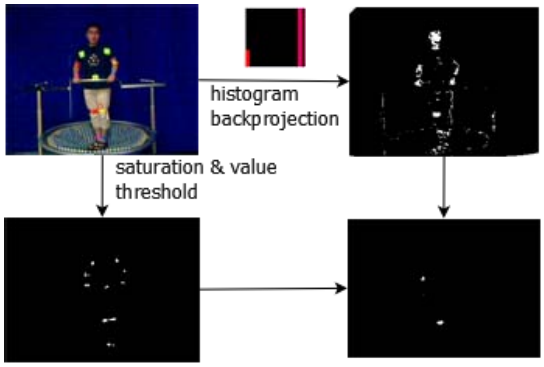

Figure 3: *Histogram backprojection and saturation-value thresholding for red markers*

We attached eleven markers of five different colors to the user's body joints (Figure 3, top-left). The colors are chosen based on the probability of markers occluding each other, for example the left and right arms.

To initiate tracking, the user adopts a T-stand shape. This generates eleven tracking objects which are organised in a tree data structure that reflects the structure of the body.

If we lose track of a marker, we do a contour finding on the area surrounding its parent marker. For example, the parent of the elbow's marker is the marker attached to the shoulder. Out of the contours we find, we pick the one closest to the parent position and mark it as the new position of the tracking window.

3.3 Stereo Image Acquisition and 3D Reconstruction

We use two side-by-side Logitech Fusion cameras. The images are captured at a size of 320x240 pixels, with a frame rate of 30 fps. We use Dom Anker's dscam library [Anker 2008] to perform software synchronization between the cameras with an average of 0.016 seconds delay between the image acquisitions by the first and second camera.

We calibrate our cameras using Bouguet's algorithm [Bradski and Kaehler 2008] which is a refinement of the method presented by

Zhang [2000] and Tsai [1987]. This calibration algorithm yields the reprojection matrix:

$$Q = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -\frac{1}{T_x} & (c_x - c_x')/T_x \end{bmatrix} \quad (1)$$

where (c_x, c_y) are the principal point in the left image, f is the focal length of the camera, c_x' is the principal point x-coordinate in the right image, and T_x is the x-translation between the cameras. We can then project the point (x, y) into 3D space using:

$$Q \times \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} \quad (2)$$

where the resulting 3D coordinate is (X/W, Y/W, Z/W) and d is the x-coordinate disparity between the left and the right images.

We use a Kalman filter to reduce the noise in the reconstructed 3D data [Bradski and Kaehler 2008]. We model the system with position as the measurement parameter(z_k) and two dynamic parameters(x_k) of position and velocity. The velocity parameter is used for gait analysis.

3.4 Physical Control of Character

We model the avatar using Cal3D [2006], an open source skeletal based 3D character animation library. Modeled, rigged, and skinned characters can be exported from 3D Studio Max through Cal3D exporter. The library also allows programmable control over the avatar skeleton and animations.

To synthesized natural-looking walk, we separated the motion tracking into two parts. The lower body parts (leg and upper leg) are instead developed as FSM-based controllers in similar mechanism as to that presented by [Raibert and Hodgins 1991].

3.4.1 Upper Body Tracking

The upper body model (hand and forearms) are controlled using 3DOF motion capture. We manipulate the avatar's hand and forearm to match the position of the tracker relative to its parent. For example, the forearm is rotated according to relative rotation between shoulder and elbow marker.

3.4.2 Gait Model

We cannot use the 3D tracking information data to directly control the animation of the legs because the movement of the legs (sliding down an inclined plane) is not the same as in a normal walk. To synthesize a robust and natural-looking walk, we use a similar mechanism to that of Raibert and Hodgins [1991]. Like them, we developed a Finite State Machine (FSM) controller for the lower body parts. On the top-level, the controller are modelled as two states; idle and walking. State transition is decided from the position and velocity of the markers. Fig. 4. shows the state transitions of the gait model.

When both feet are not moving (zero velocity), the system will default to idle state. From idle state, it can transition to walking state when an increase in velocity in either the left or right foot is detected.

In the walking state, we compare the current location of each foot relative to its maximum displacement. This yields the current animation time t_c. Immediately changing the animation time to t_c would result in jerky movement, so instead, the animation is allowed to run until it reaches t_c, though this target will change as the foot moves. This introduces a slight lag, but users find this acceptable.

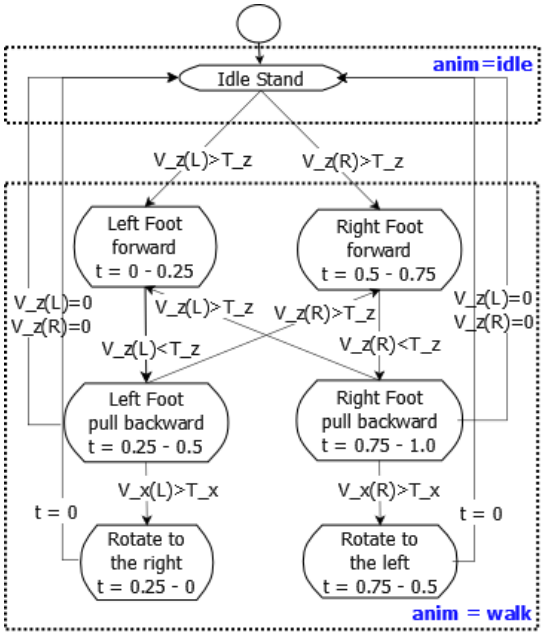

Figure 4: *Animation state*

3.4.3 Changing the Direction of Walking

To change the walking direction, the user needs to adopt a side-sliding gait that is similar to the natural way of turning while walking. For example, for turning left the walker would swing forward the right foot then slide it to the right and then slide it back again.

The system will track the sliding motion and rotate the avatar accordingly. Once a change of direction of gait is recognised, instead of continuing with the animation, we run the animation in reverse. Since when we are changing the direction of walk, the back foot usually acts as the pivot point where the forward foot will be pulled back to.

4 Experiment Results

We have conducted a number of preliminary tests to evaluate the performance of the system. We built a simple virtual environment consisting of grid floors so that the walking gait can be clearly viewed. The setup and results are shown on Fig. 5.

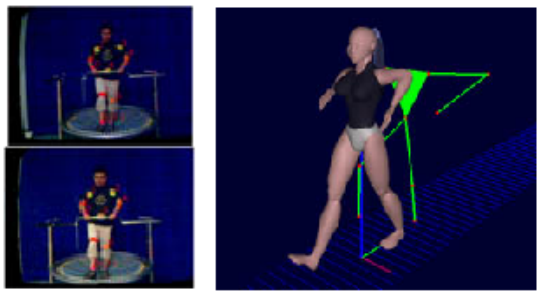

Figure 5: *Experiment result*

If the user moves the markers too quickly or a marker is occluded the system loses track, but recovers instantly when it finds them again. In the meantime it assumes the marker is at the previous location.

We have tested the system to run on hardware with a 2.4GHz Intel Core2 Duo processor, 3 GB DDR2, and NVidia GeForce 8600 GT. The system was capable of performing in real-time with image acquisition rate of around 30 fps. Image processing and 3D projection produce total latency between 20-50ms. Most of the latency (15-45ms) occurred during mean-shift tracking depending on the quality of the markers images.

5 Conclusion and Future Work

Walking on foot is the most intuitive way to move around the real world. We have presented a new framework that combines OBDP and computer vision tracking. The improved OBDP platform is robust so it can be used regularly. The tracking mechanism is efficient enough to allow real-time performance on mid-tier hardware. The system is also capable of 3DOF motion capture for the upper body parts.

We will continue to develop the system by: improving the tracking mechanism so that color markers are no longer necessary, improving the display system to use a hemispherical screen instead of screen projector to reduce motion sickness, and adding recognition of other gestures such as sidestepping and backstepping.

References

ANKER, D., 2008. `http://tech.groups.yahoo.com/group/OpenCV/files/dscam.zip`.

BLACKWOOD. 2007. *Blackwoods Catalog 2007 release*. J Blackwood & Son Limited.

BOUGUILA, L., EVEQUOZ, F., COURANT, M., AND HIRSBRUNNER, B. 2004. Walking-pad: a step-in-place locomotion interface for virtual environments. In *Proceedings of the 6th international conference on Multimodal interfaces*, ACM.

BRADSKI, G., AND KAEHLER, A. 2008. *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc.

BRADSKI, G. R. 1998. Real time face and object tracking as a component of a perceptual user interface. In *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, 214–219.

BROOKS, F. P. 1986. A dynamic graphics system for simulating virtual buildings. In *Proceedings of the 1986 Workshop on Interactive 3D Graphics*, ACM, 9–21.

CAL3D, 2006. Cal3d character animation library. `http://home.gna.org/cal3d/`.

CHENG, Y. 1995. Mean shift, mode seeking, and clustering. In *IEEE Trans. Pattern Analysis Machine Intelligence*, vol. 17, 790–799.

CHRISTENSEN, R. 1998. Inertial force feedback for a locomotion interface. In *Proceedings ASME Dynamic Systems and Control Divisions*, vol. 64, 119–126.

COMANICIU, D., RAMESH, V., AND MEER, P. 2000. Real-time tracking of non-rigid objects using mean shift. In *Computer Vision and Pattern Recognition, 2000. Proceedings on IEEE Conference*, vol. 2, 142–149.

DARKEN, R., COCKAYNE, W., AND CARMEIN, D. 1997. The omni-directional treadmill:a locomotion device for virtual worlds. In *Proceedings of UIST'97*.

HOLLERBACH, J. M., AND YANGMING XU, R. R. C., AND JACOBSON, S. 2000. Design specifications for the second generation sarcos treadport locomotion interface. In *In Haptics Symposium, Proc. ASME Dynamic Systems and Control Division*, vol. 69, 1293.

HUANG, J.-Y. 2003. An omnidirectional stroll-based virtual reality interface and its application on overhead crane training. In *IEEE Tran. Multimedia*, vol. 5, 39.

IWATA, H., AND FUJI, T. 1996. Virtual perambulator: A novel interface device for locomotion in virtual environment. In *Proceedings of IEEE 1996 Virtual Reality Annual International Symposium*.

IWATA, H., YANO, H., AND NAKAIZUMI, F. 2001. Gait master: A versatile locomotion interface for uneven virtual terrain. In *Proceedings of IEEE Virtual Reality 2001 Conference*, 131–137.

IWATA, H., YANO, H., FUKUSHIMA, H., AND NOMA, H. 2005. Circula floor. In *IEEE Computer Graphics and Applications*, vol. 25, 64–67.

IWATA, H., YANO, H., AND TOMIOKA, H. 2006. Powered shoes. In *SIGGRAPH 2006 Conference DVD*.

IWATA, H., YANO, H., AND TOMIYOSHI, M. 2007. String walker. In *International Conference on Computer Graphics and Interactive Techniques*, no. 20.

IWATA, H. 1999. The trous treadmill: Realizing locomotion in ves. In *IEEE Computer Graphics and Applications*, vol. 9(6), 30–35.

K FUKUNAGA, L. H. 1975. The estimation of the gradient of a density function, with applications in pattern recognition. In *IEEE Trans. Information Theory*, vol. 21, 32–40.

NOMA, H., AND MIYASATO, T. 1998. Design for locomotion interface in a large scale virtual environment. atlas: Atr locomotion interface for active self motion. In *Proc. ASME Dynamic Systems and Control Division*, 111–118.

RAIBERT, M. H., AND HODGINS, J. K. 1991. Animation of dynamic legged locomotion. In *In Proc. ACM SIGGRAPH 1991*, vol. 25, 349–358.

SWAIN, M. J., AND BALLARD, D. H. 1990. Indexing via color histograms. In *Computer Vision, 1990. Proceedings, Third International Conference on*, 390–393.

TSAI, R. Y. 1987. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off -the-shelf tv cameras and lenses. In *IEEE Journal of Robotics and Automation*, vol. 3, 323–344.

USOH, M., ARTHUR, K., WHITTON, M. C., BASTOS, R., STEED, A., SLATER, M., AND BROOKS, F. P. 1999. Walking $>$ walking-in-place $>$ flying, in virtual environments. In *SIGGRAPH 1999*, 359–364.

ZANBAKA, C. A., LOK, B. C., BABU, S. V., BABU, C. V., ULINSKI, A. C., AND HODGES, L. F. 2005. Comparison of path visualizations and cognitive measures relative to travel technique in a virtual environment. In *IEEE TVCG*, vol. 11, 694–705.

ZHANG, Z. 2000. A flexible new technique for camera calibration. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, 1330–1334.