

Chapter 11

Stepping-Driven Locomotion Interfaces

Mary C. Whitton and Tabitha C. Peck

Abstract Walking-in-place and real-walking locomotion interfaces for virtual environment systems are interfaces that are driven by the user's actual stepping motions and do not include treadmills or other mechanical devices. While both walking-in-place and real-walking interfaces compute the user's speed and direction and convert those values into viewpoint movement between frames, they differ in how they enable the user to move to any distant location in very large virtual scenes. Walking-in-place constrains the user's actual movement to a small area and translates stepping-in-place motions into viewpoint movement. Real-walking applies one of several techniques to transform the virtual scene so that the user's physical path stays within the available laboratory space. This chapter discusses implementations of these two types of interfaces with particular regard to how walking-in-place interfaces generate smooth motion and how real-walking interfaces modify the user's view of the scene so deviations from her real motion are less detectable.

11.1 Designing Stepping-Driven Locomotion for Virtual Environment Systems

Arguably, the locomotion interfaces for Immersive Virtual Environment (IVE) systems that are most natural are those that employ a stepping metaphor, i.e., they require that users repeatedly move their feet up and down, just as if walking in the real world.

M. C. Whitton (✉)

Department of Computer Science, The University of North Carolina at Chapel Hill,
Chapel Hill, NC, USA
e-mail: whitton@cs.unc.edu

T. C. Peck

Event Lab, Faculty of Psychology, University of Barcelona,
Barcelona, Spain
e-mail: tabitha.peck@gmail.com

Such interfaces give users a locomotion experience that is close to natural walking in the real world. Chapter 9 of this volume, Technologies of Locomotion Interface, describes mechanically-assisted walking interfaces such as treadmills and cycles. This chapter is about stepping-driven interfaces that are not mechanically assisted.

In *walking-in-place (WIP) interfaces*, users make stepping motions but do not physically move forward. Sensor data, captured from the user's in-place stepping motions and other sensors, are used to control the movement of the user's viewpoint through the virtual scene. The primary technical challenge in WIP systems is *controlling the user's speed* so that it is both responsive and smooth; direction can be set with any of a number of techniques. Using the taxonomy in Bowman et al. [4], WIP is a hybrid interface: physical because the user makes repeated movements, and virtual because the user does not move through physical space.

In *real-walking interfaces*, a purely physical interface in Bowman et al.'s taxonomy, users really walk to move through the virtual scene and the physical (lab) environment. The easy case is when the virtual scene fits within the lab: There is a one-to-one mapping between the change in the user's tracker-reported pose (position and orientation) and the change in viewpoint for each frame. Speed and direction are controlled by how fast and in what direction the user moves. This is just as in *natural walking*.

The more difficult real-walking case is when the virtual scene is larger than the lab: The mapping between changes in tracker-reported pose and changes in viewpoint can no longer be one-to-one if the user is to travel to areas in the virtual scene that lie outside the confines of the lab. Thus, the primary technical challenge in real-walking interfaces for large scenes is *modifying the transform applied to the viewpoint (or scene)* so that the user changes her real, physical direction in a way that keeps her path through the virtual scene within the physical lab space. Recent locomotion taxonomies have added categories for new real-walking techniques: Arns' taxonomy includes interfaces using *scaled rotation* and/or *scaled translation* [1] and Wendt's taxonomy includes interfaces that *recenter* users via *redirection* techniques [40].

In this chapter we discuss only stepping-driven locomotion interfaces for virtual scenes that are larger than the lab's tracked space. The locomotion interface techniques reported here were developed for IVE systems that use tracked head-mounted display devices (HMDs). With some adaptation, walking-in-place can be used in single- or multi-wall projection display systems. Redirected-walking, one of the techniques for real-walking in large scenes, has also been employed in multi-wall display systems [28]. The interfaces described here do not require stereo-viewing.

Research has shown that locomotion interfaces that require the user to make stepping movements induce a higher sense of presence, are more natural, and enable better user navigation than other interfaces [22, 35]. These benefits make stepping-driven interfaces a worthy subject of study. We conclude this introduction with general goals for locomotion interfaces in IVEs and specific goals for setting locomotion speed and direction. We then discuss walking-in-place and real-walking virtual locomotion interfaces in depth.

General goals for locomotion interfaces. To be widely adopted, a locomotion interface for IVEs has more requirements than simply enabling movement from place to place. Other desirable features of locomotion interfaces include:

- Is easy to learn and easy to use; incurs low cognitive load;
- Leaves the user’s hands free so she can use task-related tools;
- Does not increase occurrence or severity of simulator sickness;
- Prevents users from running into real-world obstructions and walls;
- Minimizes encumbrances
 - Is easy to don and doff;
 - Ensures that equipment, including safety equipment, does not interfere with other task-related gear the user may be wearing;
- Minimizes required supporting infrastructure, e.g., tracking systems, for portability and cost.

Goals for setting speed. The notional speed versus time profile (Fig. 11.1a) is a standard against which to compare similar speed/time profiles for our interfaces. Figure 11.1b shows an actual profile generated from (noisy) head-tracker data. The same development, rhythmic, and decay phases are visible in both profiles. We propose four design goals for setting user speed:

- Starting and stopping latency should be minimized. Movement in the virtual scene should begin as soon as the user initiates a step and stop when the user stops stepping. Starting latency is annoying for casual walking and interferes with the timing of quick movements. Stopping latency can result in overshooting the desired stopping location leading to unintended collisions with or interpenetration of objects in the scene.
- Users should be able to adjust their speed continually during a step, as we can with natural walking. If speed is controlled by data measured only once per step, e.g., foot-strike or foot-off speed, continuous control of speed is not possible.

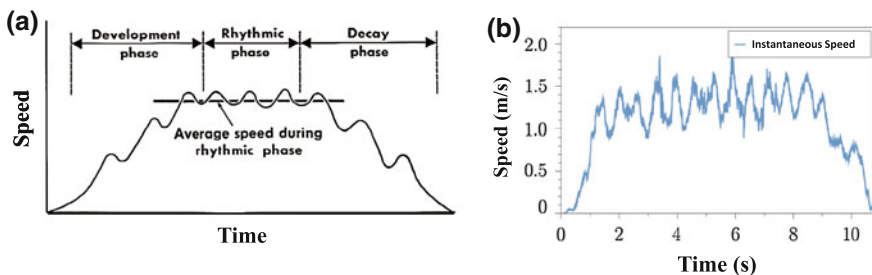


Fig. 11.1 **a** Notional speed versus time plot for one start-to-stop walking event. **b** Speed versus time plot computed from head-tracker data. The higher frequency variations in speed are caused by fore-aft head bob and roughly correspond to steps. Notice that during the rhythmic phase, speed stays well above zero. We try to replicate the general shape of the speed profile with our interfaces. (**a** Adapted from Inman [14]; **b** reproduced from Wendt [40])

- Virtual walking speed should stay relatively constant during the rhythmic phase to avoid detectable variations in optic flow—the change in patterns of light on the retina occurring during movement.
- The system should allow fine positioning or maneuvering steps that do not initiate a full step’s movement.

Goals for setting direction. The goals for direction setting are to make it as easy as natural walking and to avoid introducing sensory conflict.

- Users should be able to move in any direction—forward, backward, sideways, or at any angle.
- As in natural walking, the direction of movement should be independent of user’s view direction and body orientation. Reinforcing the results reported in Bowman et al. [3], the description of the *Pointman* interface includes a cogent argument for independence of these parameters for tactical movements [37].
- Direction setting should be hands-free, as it is in natural walking, so the hands can be used for application-specific interactions with the environment.

11.2 Walking-in-Place Interfaces

Walking-in-place (WIP) is a locomotion interface technique for Immersive Virtual Environment systems that uses data describing the stepping-in-place gesture to control locomotion speed and uses any one of a number of techniques or input devices to set locomotion direction.

11.2.1 Setting Speed: Interpreting Stepping Gestures

Repeated stepping gestures have several distinct, observable, and measurable phases. Starting from the eight-phase human gait cycle, [41] proposed the six-phase walking-in-place gait cycle shown in Fig. 11.2. There are three events associated with each leg’s step: foot off, maximum step height, and foot strike. With appropriate sensors, it is possible to detect each of these events, make measurements about them, and apply time stamps to them. The resulting data are what is available to determine whether the user is moving, and, if she is moving, how fast. The question of whether the user is moving includes both whether the user is starting to move and whether the user is stopping.

11.2.1.1 Detecting Foot-Strike Events

The earliest WIP interfaces computed forward motion based on indirect or direct detection of foot-strikes: each time a foot strike was detected, the user’s viewpoint

was moved forward by some amount. The faster the foot strikes occurred, the faster the user moved through the virtual scene.

A very early walking-in-place system, called a virtual treadmill, applied a neural network to head tracker data to detect local maxima in stepping-related vertical head-bob [35]. A set amount of forward movement, inserted over several frames, was added between detected steps. The neural network required four positive “step” signals before initiating movement and two “no step” signals before stopping. Starting latency was about two seconds; stopping, about one second.

Other methods of foot-strike detection include pressure sensors in shoes [36], a floor-based array of pressure sensors [2], and head-worn accelerometers [44]. Unlike the first two methods which produce a binary variable when a step is detected, the latter technique generates a stream of accelerometer data in which foot-strikes are detected as local maxima.

Starting latency is a problem for foot-strike techniques: a step is not recognized until the foot has been lifted and returned to the ground. For a casual walking speed of 3 mph and a 24” step length, this latency is around half a second.

Movement can be implemented by choosing a moderate base speed and computing the distance the viewpoint must be moved for each foot strike to achieve that speed through the scene. That incremental distance is added to the viewpoint pose over one or more frames. Stepping faster or slower changes speed, but it is not possible to adjust speed between foot-strikes. Maneuvering is not possible unless the algorithm includes a sensor-signal threshold so that it ignores small foot movements or light floor strikes.

Moving the user forward a set distance for each foot-strike generally does not lead to a relatively constant speed for rhythmic-phase walking even if the total distance to be moved is spread over several frames. In an exaggerated fashion, Fig. 11.3 shows a speed profile for distance (a) added uniformly over several frames and (b) added in a sawtooth pattern in order to avoid multi-frame pauses in the optic flow occurring when speed goes to zero or near zero between steps. Comparing these profiles to Fig. 11.1 reveals that neither waveform is a good approximation of natural walking. Overcoming the limitations of discrete-step based interfaces—latency, speed variations during rhythmic-phase, inability to maneuver and adjust speed—requires additional data about the user’s stepping motion.

11.2.1.2 Continuously Measuring Leg Position

The addition of trackers to the front or back of the user’s legs (or knees, shins, ankles, or feet) provides a continuous stream of time-stamped tracker data from which the six events in the walking-in-place cycle can be detected: motion of one leg begins at foot-off, motion reverses direction when the tracker reaches its maximum extent, and motion of that leg stops at foot-strike; then similarly for the other leg. Leg speed can be computed from the tracker data.

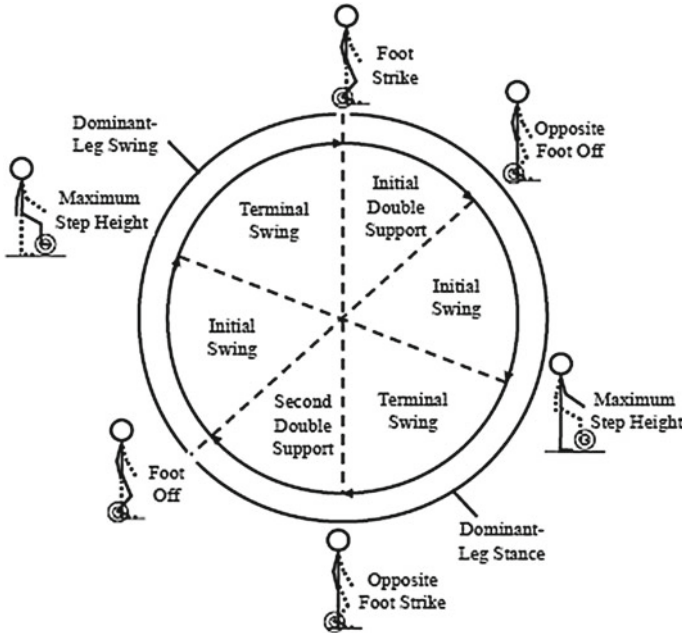


Fig. 11.2 Six-phase walking-in-place gait cycle. Stepping gestures can be quantified by detecting the events, measuring how high the legs are raised, and noting the timing between the events. Note that (1) the phase when both feet are on the floor is called *double support*, and (2) stepping frequency can be calculated from the time stamps of *any* three successive events. (Reproduced from Wendt [40])

Gaiter is a WIP system enabling locomotion in a virtual scene of unlimited size with some limited real-space maneuvering [36]. Knee excursion in the horizontal plane, measured by shin-worn trackers, differentiates virtual and real steps. In a virtual step, i.e., stepping-in-place, the knee moves out (and up) and back again; in a real step the knee moves out and stays out as the user takes the real step. Startup latency is half a step since the system cannot tell if the step is real or virtual until the knee has reached its maximum extent and either stopped or begun to travel back.

Yan et al. designed a system that set locomotion speed based on leg speed during the period of high leg acceleration occurring just after foot-off [45]. Using results from the biomechanics literature and experimentally developed relationships among leg-lift speed, step frequency, and forward velocity for natural walking and for stepping-in-place, the team developed a user-specific linear function relating the stepping-in-place leg-lift speed and forward velocity. Speed was set once per step using this function. Motion did not begin until a leg-lift speed threshold was exceeded, resulting in a starting latency of approximately one-quarter of a step. The threshold prevented false steps and allowed (slow) maneuvering steps. Per-step movement was spread across frames and a Kalman filter was used to smooth forward movement between leg-lifts.

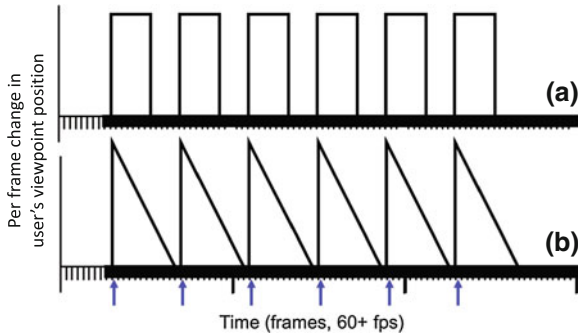


Fig. 11.3 Exaggerated examples of foot-strike driven speed profiles. Time is measured in frames; small arrows below the X-axis indicate foot-strikes every 400 ms (~ 3.6 mph). **a** User's viewpoint is moved a set distance for each foot-strike and that distance is added uniformly over several frames. **b** To avoid pauses in the optic flow while speed is zero, the set distance is added non-uniformly over several frames. The saw-tooth shape minimizes latency at the start of motion for each step. (After a figure in Feasel et al. [9])

11.2.1.3 Techniques to Smooth Speed Between Foot Strikes

Low-Latency Continuous Motion WIP (LLCM-WIP). LLCM-WIP was developed to reduce starting and stopping latency and to smooth speed during rhythmic-phase walking [9]. LLCM-WIP uses trackers placed just below the user's knees. From the tracker data it finds the location of the user's heel via a rigid body transform and calculates the speed of the user's heel in the vertical axis from that data. LLCM-WIP supports maneuvering by requiring that a heel-speed threshold be exceeded before a full step forward is taken. After some signal processing, vertical heel speeds above the threshold are mapped to locomotion speed. The locomotion speed signal is noisy and dips close to zero during the double support phases of gait. At the cost of approximately 100 ms of latency, filtering smoothes the output speed and reduces, but does not eliminate, those speed dips (Fig. 11.4). Because virtual speed is mapped continuously from heel speed, speed can be changed at any time by speeding or slowing stepping movements.

Gait-Understanding-Driven WIP (GUD-WIP). GUD-WIP addresses the problem of speed variation during rhythmic walking with a technique that updates speed six times during each two-step WIP gait cycle using a quadratic function reported in the biomechanics literature that relates stepping-frequency and speed. Figure 11.5 shows the GUD-WIP system in use.

The timing of events in the WIP gait cycle (Fig. 11.2) is discoverable from time-stamped logs of tracking data from the user's shins. The events occur when tracker position starts changing (foot off), stops changing (foot strike), changes direction (reaching maximum step height). Stepping frequency is computed from the time stamps of the three most recent WIP-cycle events. After startup, step frequency can be (re)computed six times in each two-step cycle. Startup requires three gait events,

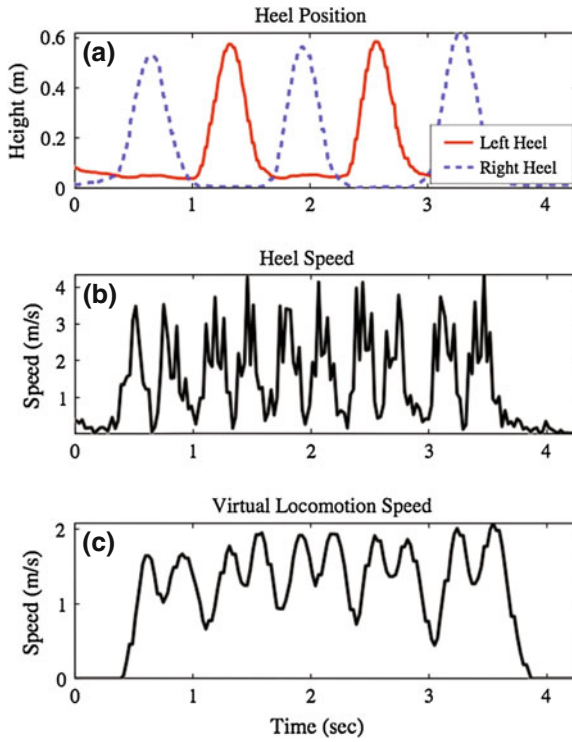


Fig. 11.4 LLCM-WIP system overview. **a** Vertical position of the user's heels; **b** Heel speed obtained through differentiation; **c** Virtual locomotion speed. (Reproduced from Feasel et al. [9])

a latency of one step. The GUD-WIP algorithm consciously traded longer stopping latency (~ 500 ms) for smoother inter-step motion.

While Yan et al. used a linear relationship between step frequency and speed, the biomechanics literature reports a quadratic relationship between these two values. Wendt used the formula reported by Dean [7] to compute virtual speed six times per 2-step cycle [41]. Figure 11.6 shows Dean's equation, a graph of its curve, and step-frequency to speed data points from other published works. The formula is partially customized with user height, (h).

Figure 11.7 shows LLCM-WIP and GUD-WIP speed profiles computed from the tracker log of the same five-step sequence from the rhythmic phase of a start-to-stop walking event. Note that unlike LLCM-WIP, GUD-WIP speed (and hence optic flow) does not approach zero during double support; however, there are discontinuities when speed is updated (3 times/step). We do not yet know if these discontinuities have perceptual or task-performance consequences.



Fig. 11.5 GUD-WIP in use. The user is free to maneuver and rotate within the PVC pipe “cage” that constrains his motion and protects the cameras. The beacons on the user’s shins (inset) are sensed by the eight PhaseSpace cameras arrayed around the user. Data from knee tracking is used for both speed and the direction setting. The HMD and head-tracker have wired connections; the PhaseSpace system is wireless. (Reproduced from Wendt [40])

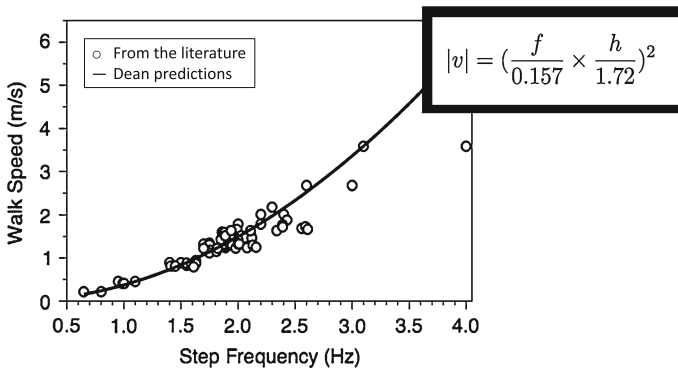


Fig. 11.6 Equation used to compute speed from step frequency and user height. The *solid line* is the curve for $h = 1.67$ m. The *open circles* represent data relating step frequency and speed gathered from the literature and cited in [40]). (Equation from Dean [7]; figure adapted from Wendt [40])

11.2.2 Setting Direction for Walking-in-Place

There is nothing particularly hard about simplistically setting the direction of movement to “forward” in a walking-in-place interface. The difficulty arises when incorporating the goals of allowing the user to move in any direction and keeping the direction of movement independent of view direction and body orientation.

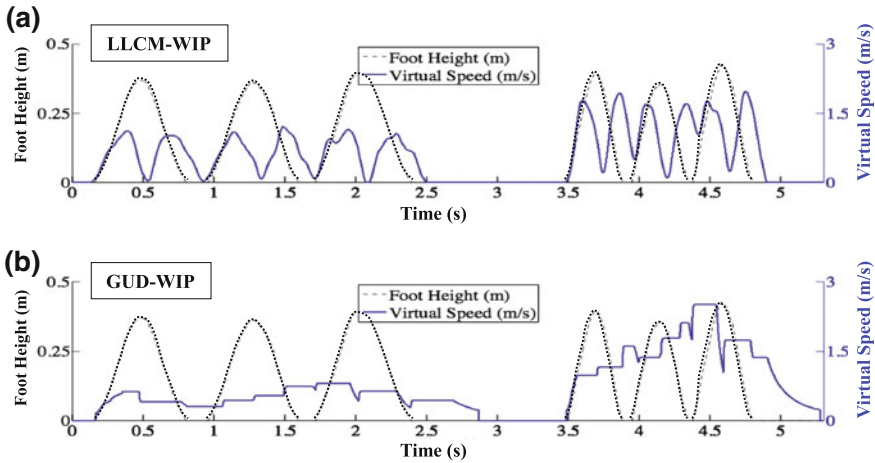


Fig. 11.7 Solid lines are the output speed profiles for (a) LLCM-WIP and (b) GUD-WIP at two walking speeds. Dotted lines are the height of *one* ankle. The trace of the other ankle’s height would be the same shape, but offset so that its peaks are in the gaps. In a, note the dips toward zero in LLCM-WIP output during the double-support event between steps, offset by the ~ 100 ms latency. In b note the three changes in speed during each step and that the speed does not dip to nearly zero. You can also see the ~ 500 ms stopping latency. (Figure adapted from Wendt et al. [41])

11.2.2.1 Hands-Free Direction Setting Techniques

Head-directed motion. Often called *gaze-directed*, head-directed motion uses the forward direction of the (head-tracked) head pose as the direction of motion. This requires no additional apparatus and is easy to implement and learn to use. However, the user cannot move and look around at the same time, as people normally do. Slater’s team’s neural-network-based WIP system used head-directed motion [35].

Torso-directed motion. Torso-directed motion is one of several direction-setting techniques that depends on data from trackers located on the user’s body. A tracker on the user’s torso (front or back; chest or hips) can be used to set “forward” to be the direction the user’s body is facing. Use of the additional tracker means that torso-directed movement is independent of head orientation, so users can walk and look around at the same time. A limitation of such body-worn tracker techniques is that users cannot move backwards or sideways, as both of those motions require decoupling direction of motion from the direction the body is facing.

Gesture-controlled direction. Gesture-controlled direction setting techniques interpret tracked movements of the user’s hands, head, legs, or feet to establish direction of movement. While we would argue that any use of gestures reduces the naturalness of walking-metaphor interfaces, gestures are frequently used. In Gaiter, sideways motion is enabled by swinging the leg to the side from the hip; backward motion is enabled by kicking backward from the knee [36].

11.2.2.2 Hand-Held Direction Setting Devices

The most common hand-held devices for setting direction are tracked wands and joysticks that may or may not be part of a game controller. While the efficacy of these interfaces is well accepted, they come at the cost of limiting how the user can use her hands to interact with the virtual scene in application tasks.

Wands and pointing. Wands typically include a tracker and one or more other input devices such as buttons. Forward direction can be set by a combination of arm gesture and a hand-held three degrees of freedom (3DOF) tracker by using the tracker-measured positions of the user's head and the wand to define the direction vector. If the tracker is 6DOF, direction of movement can be set from the tracker's coordinate system; typically movement is in the direction of the longitudinal axis of the wand. The biomechanics of human shoulders and wrists limit the range of directions that can be set with wands without repositioning the body.

Joysticks/game controllers. Joysticks/game controllers can specify motion in any arbitrary direction, so they are an attractive solution for setting direction. Most often the user wears a 6DOF tracker on her body and the joystick outputs are interpreted in that coordinate system. This means that when the user pushes the joystick perpendicularly away from herself, it causes her move in the direction her body is facing. Note that the tracker data does not restrict the direction of movement; it simply establishes a body-centric coordinate system for the joystick.

Integrated tracker/joystick and task tool. The encumbrance of the hand-held interface devices can be mitigated in part if they are integrated into the task tools used in the IVE system. A well-developed example is the instrumented rifles with integrated thumb-operated joysticks (thumb-sticks) that are used in many military training systems, including the United States Army's relatively new Dismounted Soldier Training System [26]. An evaluation of an earlier system reported both positive and negative aspects of the thumb-sticks [25].

11.2.3 *The Future for Walking-in-Place Interfaces*

Modeling human walking in ways suitable for use in WIP interfaces is not yet a solved problem. Techniques inspired by biomechanics have addressed setting virtual speed during the rhythmic phase of walking and have tried to minimize starting and stopping latency, but they have not yet addressed the shape of the velocity profile during those two phases of walking, or variations in speed that may result from turning or walking with a heavy load. We do not yet know if the discrete changes in speed that occur in GUD-WIP affect users' perception of the environment or their task performance. We do not know how the mathematical models may change if the user is running.

To be cost effective, walking-in-place techniques have often made do with very little information about the user's actual motion. In some cases, the only data available for use in the locomotion algorithm is from the head tracker. Full body tracking systems provide rich data, but also are costly, encumbering, and inconvenient. Their use has to be carefully balanced against the improvements in naturalness made possible by the richer data.

Consumer products have started to change the landscape. Applications for the Kinect™ range camera can compute and update the 3D pose of a user's skeleton each frame time. The Kinect is inexpensive and does not require the user to wear any additional gear [46]. Small wireless sensors—accelerometers, magnetometers, and gyros—will be an inexpensive and non-encumbering source of data measuring user motion that can be used as inputs to the locomotion algorithm. A proof-of-concept system using such devices is described in Kim et al. [16].

With a richer set of input data, walking-in-place locomotion techniques will be better able to model and simulate the experience of natural walking for users of IVE systems.

11.3 Real-Walking Interfaces

Real-walking interfaces enable HMD-IVE-system users to naturally walk around the virtual scene just as they would in the real world. Because the user must be tracked, restricting the size of the virtual scene to the size of the tracked space is the simplest case for real-walking. If the virtual scene fits in the tracked space, the user can freely walk about in the entire virtual space, the user's real-world speed can be mapped in a one-to-one ratio to her virtual speed, and her direction in the virtual scene can be directly controlled by her direction of motion in the real world.

Complications with real-walking interfaces arise when the virtual scene is larger than the tracked lab area. Mapping the user's actual speed and direction one-to-one with virtual speed and direction no longer enables the user to travel through the entire scene, as to do so would require leaving the tracked area. Numerous techniques, most of which exploit the imprecision of human perception, have been developed to make real-walking a viable locomotion technique for larger-than-tracked-space virtual scenes. Initial implementations focused on transformations of the scene model or the user's motion by manipulating the ratio between the user's real and virtual speeds and directions. A newer technique changes the structure of the scene model [34]. We discuss both approaches.

11.3.1 *Manipulating Speed*

Manipulating speed in real-walking interfaces can be thought of as altering the ratio between the user's real walking speed and virtual speed so that it is no longer one-to-one.

11.3.1.1 Perceptual Foundation

As people move, their view of their surroundings changes, and information about the layout of the environment and the shape of surfaces, as well as their relative position within the environment, is revealed.

The illusion of self-motion, known as *vection*, can be produced by visual stimulation alone. For example vection can occur when a person is sitting in a stationary car and the adjacent car starts to move forward, causing the person in the stationary car to perceive the sensation of backwards motion.

Movement, essential for accurate perception of the environment, causes *optic flow*, the changing pattern of light on the optic array caused by the relative motion of the observer and environment. Optic flow patterns contain information about self-motion, the motion of objects, and the environment's three-dimensional (3D) structure. If an observer is moving forward, the optic flow will radiate outward from the *center of expansion*—the point toward which the person is moving; if a person is riding in a train and looking out the window, the optic flow will move horizontally across the observer's retina producing *lamellar flow*.

The results of a study by Warren led him to speculate that optical information could be exploited to control locomotion [38]. An experiment by Konczak found that as optic flow slowed, subjects' walking speed slightly increased; however increasing the speed of optic flow appeared to have no effect on participants' real speed [17]. Konczak's results suggest that increasing the ratio between the users' virtual and real walking speeds (i.e., increasing optical flow speed relative to walking speed) could be employed to enable users to travel greater virtual distances in the same number of steps.

11.3.1.2 Interfaces that Manipulate Speed

Real-walking locomotion techniques that alter the ratio between the user's real and virtual speeds, thus altering optic flow, include Seven League Boots [13, 30] and Scaled Translational Gain [42]. Each of these methods maps the user's real translation into increased virtual translation. For example, when the user takes one step in the real world she is translated two or three steps in the virtual world.

Altering the ratio between the user's real and virtual speed enables the size of the virtual scene to be scaled to a multiple of the size of the tracked space, based on the ratio between real and virtual speeds. However problems can occur if the ratio becomes very large. For example, if the user's motion is increased by a factor of 100, then when the user takes one real step she travels 100 steps forward in the virtual scene. This motion, although smooth and in the same direction as the user's motion, may cause disorientation as it places the user far away from their starting location. This rapid change in the user's location is similar to teleportation which is known to disorient the user [3].

An additional problem with speed-scaling methods arises because people move their heads side-to-side as well as forward-to-backward as they walk. When the ratio between real and virtual motion is large, the side-to-side motions are also multiplied and can cause the scene to appear unstable. To eliminate the side-to-side motion, Interrante et al. computed the user's forward direction and scaled user motion only in this predicted direction [13].

Another potential problem with altering user speed is that when the difference between physical and virtual speeds is large, people will be able to notice the discrepancy. A method introduced by Bruder et al. uses change blindness techniques to effectively move the user forward in the VE while the user is unaware of it [5]. Change blindness theory posits that people are unaware of changes made in their view when the changes occur during saccadic eye movements. Change blindness is discussed further in Chap. 14. As is common in change blindness techniques, Bruder et al. display a blank screen that flashes in the HMD for 60–100 ms. While the screen is blanked, the virtual scene is translated in the user's direction, thus altering the ratio between the user's real and virtual speed. Due to change blindness, the user is less aware of the alterations that have occurred.

11.3.2 Manipulating Direction

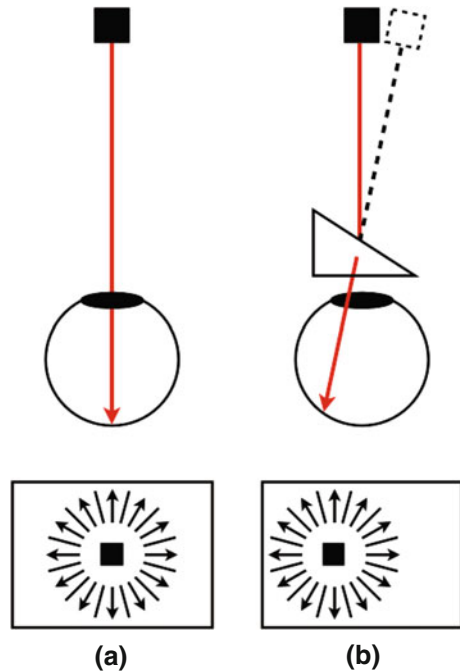
Manipulating direction for real-walking techniques can be thought of as altering the ratio between real world direction and virtual world directions of movement.

11.3.2.1 Perceptual Foundation

Altering the ratio between real and virtual directions is possible because vision guides *heading direction*, the user's direction of motion. The egocentric direction hypothesis and Gibson's theories about optic flow [10] provide theoretical support for locomotion systems that guide user direction by manipulating the user's view of the virtual scene as generated by the IVE system.

The *egocentric direction hypothesis* states that heading direction is determined by the anterior-posterior axis of the body. This theory was explored by Rushton et al. after observing a subject who suffers from *unilateral visual neglect* (UVN)—damage to one side of the cerebral hemisphere and the inability to respond to stimuli on the side opposite the lesion [31]. UVN is often associated with a misperception of location. Rushton et al. observed the subject walking in curved paths to reach target objects. To simulate the misperception of the target location for individuals without UVN, Rushton et al. had participants wear prisms in front of their eyes and found participants walked a curved path toward the target. The prism translates not only the target object, but also the optic flow produced when the participant walked toward the target (Fig. 11.8).

Fig. 11.8 a As the person looks directly along the heading vector, optic flow radiates outward from the center of vision. **b** A prism is placed in front of the eye, which shifts the visual location of the goal and the location of the radial optic flow. The optic flow on the retina is the same pattern as on the left but shifted due to the prism. (Adapted from Rushton et al. [31])



Gibson's theories [10] suggest that heading is determined from the center of expansion of optic flow. When people walk toward a target, they adjust their movements to align heading direction with the intended goal. Warren et al. [39] further investigated whether the egocentric direction hypothesis or the optic flow hypothesis dominates. They had people walk through virtual scenes with different textures to create different amounts of optic flow to see if the amount of optic flow affected participants' heading direction as they moved to a target. Their results show that with no optic flow participants followed the egocentric direction hypothesis, however when optic flow was added to the ground plane, participants initially followed the egocentric direction hypothesis, and then after traveling a few meters participants adjusted their heading and used optic flow to aid their guidance.

The results of Warren et al. demonstrated that humans rely on both optic flow and egocentric direction to guide locomotion. These results suggest that manipulations of the visual representation of the scene can guide the user so she walks a straight path in the virtual scene concurrently with walking a curved path in the laboratory.

Slight manipulation of optic flow may go unnoticed by a user; however, extreme changes will be detectable. Studies from aircraft simulation provide further understanding of ways that IVE system and scene designers can manipulate rendered visuals without the user noticing. Research by Hosman and van der Vaart determined the sensitivity of the visual and vestibular senses to different rotation frequencies or speeds, i.e., the frequency response of the two senses [12]. The results suggest

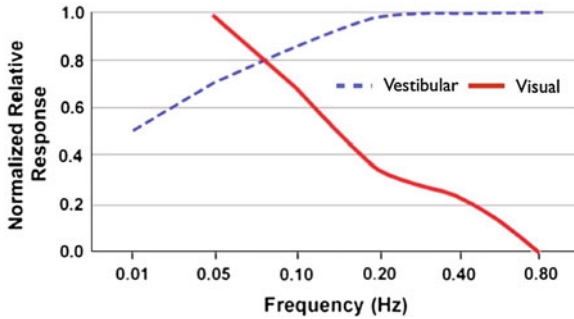


Fig. 11.9 The visual-vestibular crossover. This graph shows, in the frequency domain, the relative contributions of visual and linear vestibular cues to postural stability. (Adapted from Duh et al. [8], reproduced from Razaque [27])

that visual perception is more sensitive at low frequencies of motion and vestibular perception (sensed by the otoliths and semicircular canals) is more sensitive at higher frequencies (Fig. 11.9). These results suggest that when the head is not moving or is moving at slow frequencies, that the visual system is dominant. As head angular velocity increases, the vestibular sense comes to dominate the visual.

The important outcome of Hosman and van der Vaart’s research is the observation that when people turn their heads, the vestibular system dominates and visual manipulation may go unnoticed. Rotation of the virtual scene during head turns is therefore less likely to be detected because when people turn their heads at normal angular velocities the vestibular system dominates the visual system. As a point of reference, an angular rotation of 0.5 Hz corresponds to taking 2 s to rotate your head all the way from one side to the other and back; note that higher angular rotation frequencies (and faster head turns) are further to the right in Fig. 11.9 where vestibular cues almost totally dominate visual.

The egocentric direction hypothesis, Gibson’s theories of optic flow, and studies about the visual-vestibular crossover all provide theoretical support for manipulating the views of the virtual scene to cause the user’s virtual direction to differ from her real direction. These techniques are employed in the following locomotion interfaces.

11.3.2.2 Interfaces

Motion compression (MC) [19, 33] has a misleading name because it does not in fact compress motion. Instead, MC rotates the virtual scene around the user and remaps areas of the scene that were outside of the tracked-space into the tracked space. The MC algorithm predicts a user’s goal location based on points of interest in the scene toward which the user may be walking. The algorithm then maps the straight line of the path from the user to the predicted goal location onto the largest possible arc that will fit into the tracked space. MC continuously updates the goal location and

the rotation of the virtual scene relative to the tracked space. It is not a goal of MC to make the rotation undetectable by users.

Redirected walking (RDW) [27–29] is a technique that exploits the imprecision of human perception of self-motion—the motion of humans based on sensory cues other than vision. RDW modifies the direction of the user’s gaze by imperceptibly rotating the virtual scene around the user and redirecting the user’s (future) path back into the tracked space. Unlike MC, RDW was designed to make rotation undetectable to the user. RDW achieves undetectable rotation by exploiting the visual vestibular crossover described above. The vestibular system is dominant over the visual system at head frequencies greater than 0.07 Hz, approximately one head turn over a 14 s period, causing users to not notice unmatched real and scene rotation while turning their heads at frequencies greater than 0.07 Hz. For this reason, an integral part of the design for RDW was to make users frequently turn their heads.

Razzaque’s environments and tasks depended on static waypoints, locations that defined the user’s virtual route within the VE, for two reasons. First, a series of waypoints predetermined the user’s sequence of goal locations. Knowledge of the future goal locations enables the system to always know what part of the virtual scene should be rotated into the tracked space. Second, waypoints are a mechanism designed to make people look around. That is, users had to turn their heads to find the next waypoint. This enabled the RDW algorithm to rotate the virtual scene (during head turns) and redirect the user’s next-path-direction, i.e., the path to the next waypoint, into the tracked space.

Waypoints provided a simple answer for one of the most challenging parts of implementing a redirection system: predicting the user’s future direction. Although waypoints enable RDW, they limit applications to those that have predetermined paths and task-related reasons for users to turn their heads.

Newer implementations of redirection have added dynamic controllers: Peck and her colleagues controlled the amount of rotation added to the virtual scene based on the rotation speed of the user’s head [21, 22]; Neth et al. controlled the curvature gain based on the user’s walking speed [18]; and Hodgson et al. altered the redirection amounts based on both the user’s linear and angular velocities [11]. Chapter 10 provides a detailed description of how to modify the view transformation in redirection systems.

Additional studies and techniques have explored determining the appropriate amount of redirection that can be added at any instant [15, 32], how to steer the user within the environment [11, 21, 22, 27], and how to predict the user’s future direction [13, 21, 22].

Finally, a method presented by Suma et al. harnesses change blindness techniques by altering part of the scene model when the user is not looking at that part of the scene [34]. For example, the location of a door to a room may change from one wall to another while the user is not looking at it, thus guiding the user to walk in a different direction in the physical space by walking a different direction in the virtual space.

11.3.3 Reorientation Techniques

Many of the locomotion techniques presented in Sects. 11.3.1.1 and 11.3.1.2 use a *reorientation technique* (ROT) to handle the situation when large-area real-walking techniques fail and the user is close to walking out of the tracked space (and possibly into a wall or other obstruction). ROTs discourage the user from leaving the tracked space and rotate the virtual scene around her current virtual location. This moves the user's predicted next-path-direction into the tracked space. The user must also reorient her body by physically turning in the real environment so she can follow her desired path in the newly rotated virtual scene. Some techniques require the user to stop; others do not. As a design goal, ROTs should interfere with the virtual experience as little as possible.

In addition to waypoints, redirected walking [27–29] uses a ROT that employs a loudspeaker in the virtual scene, played through user-worn headphones, that asks the user to stop, turn her head back and forth, and then continue walking in the same direction. During the head turning the virtual world can be undetectably rotated such that the future virtual path lies within the real-world tracked space.

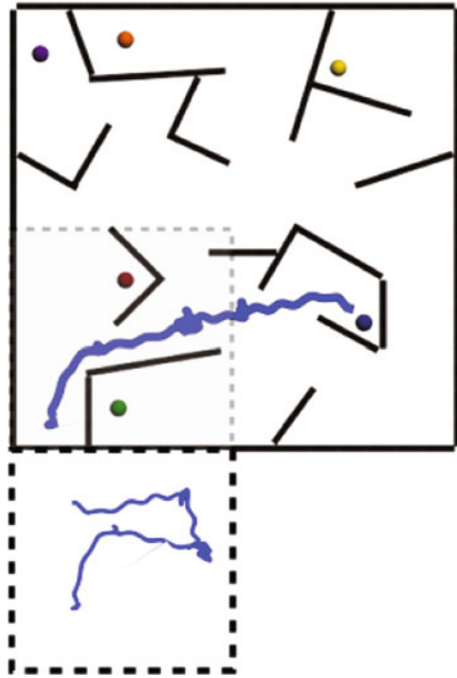
The ROT used in motion compression [19, 33] is built into the motion compression algorithm itself: as the user approaches the edge of the tracked space the arc of minimum curvature grows quite small causing the scene rotation to be large. These large rotations cause the user to feel that the scene is spinning around [19]. This method does not require the user to stop.

In the method presented by Hodgson et al. when the user is about to leave the tracked space the experimenter physically stops the user and physically turns the user back into the tracked area [11]. The HMD visuals are frozen during the turn so that the user can continue walking in the same virtual direction after the turn.

Williams et al. explored three *resetting* methods for manipulating the virtual scene when the user nears the edge of the tracked space [43]. One technique involves turning the HMD off, instructing the user to walk backwards to the middle of the lab, and then turning the HMD back on. The user will then find herself in the same place in the scene but will no longer be near the edge of the laboratory's tracked space. The second technique turns the HMD off, asks the user to turn in place, and then turns the HMD back on. The user will then find herself facing the same direction in the virtual scene, but she is facing a different direction in the tracked space.

Preliminary research suggests that the most promising is a third technique that uses an audio request for the user to stop and turn 360° [43]. The virtual scene rotates at twice the speed of the user and stops rotating after a user turn of 180°. The user is supposed to reorient herself by turning only 180° but should think she has turned 360°. This ROT attempts to trick the user into not noticing the extra rotation; however, results from Peck et al. noticed that few participants were tricked into thinking they turned 360° after only turning 180° [20, 24].

Fig. 11.10 An example of a user's path (the wiggly blue lines) through a virtual and real space. The virtual space is represented by the solid box and the real space is denoted by the dashed box. The dashed box shows relative size of real and virtual spaces. (Reproduced from Peck et al. [23])



With reorientation and/or redirection, the paths in the virtual and real world have different shapes and, as is the goal, the real world path covers less area than the virtual. Figure 11.10 shows an example.

Peck et al. introduced *distractors* which are visual objects or sounds in the virtual scene used to stop the user and elicit head rotations. Devoting attention to distractors appears to make people less aware of scene rotation while they are turning their heads [20, 24]. Distractors have been used in conjunction with redirection [21, 22], and users of the combined system scored significantly higher on a variety of navigation metrics than users of walking-in-place and joystick interfaces.

The locomotion interface implemented by Neth et al. used avatars as distractors, and when combined with their implementation of dynamic curvature gain, enabled people to successfully explore a large virtual city [18].

Alternatives to distractors include *deterrents* [22] and *Magic Barrier Tape* [6]. Both techniques display a virtual barrier to mark the real boundaries of the tracked space. The implementation from Cirio et al. uses a joystick method to move the unreachable portions of the virtual scene into the tracked space [6], whereas the implementation by Peck et al. uses distractors and redirection to rotate the unreachable part of the scene back into the tracked space [22].

11.3.4 *The Future for Real-Walking Interfaces for IVE Systems*

Manipulation of user direction should not be obtrusive to the point that it causes a break in presence. Though not yet studied, it has been proposed that

- For novice users direction manipulation should be undetectable.
- For experienced users direction manipulation should be bounded by the likelihood of increasing cognitive load and/or simulator sickness.

Large-scale real-walking techniques take advantage of the imprecisions of human perception to alter the user's perceived virtual speed and direction compared to the real world speed and direction. Newer techniques are combining multiple manipulations to enable the most usable interface possible. Different combinations of redirection and reorientation techniques are likely to enable different results and experiences.

In addition to combining redirection techniques, the current implementations can be refined and improved. The most challenging and unanswered design decisions for real-walking interfaces include how to:

- Determine an appropriate amount of speed and direction manipulation for both experienced and novice users;
- Determine the most effective way to direct the user away from the edges of the tracked space;
- Predict the user's future virtual direction.

Promising future work would compare different combinations of techniques to guide the VE designer. For training transfer applications where fatigue is important, scaled translational gain methods may not be feasible, however scaled translational gain may be most appropriate for a novice user walking through a virtual city. Possible design goals may include: accurate development of a mental model, usability, user enjoyment, speed of travel, training transfer, and designing for experienced versus novice users.

Acknowledgments Whitton's work on this chapter was supported in part by the NIH National Institute of Biomedical Imaging and Bioengineering and the Renaissance Computing Institute, and Peck's in part by the European grant VERE, an Integrated Project funded under the European Seventh Framework Program. We both thank our colleagues whose work is reported here for the pleasure and stimulation of working with them on this topic.

References

1. Arns L (2002) A new taxonomy for locomotion in virtual environments. Ph.D. Dissertation, Iowa State University, Ames, Iowa, 2002. Accessed 10 Sept 2012 from ProQuest Dissertation and Theses
2. Bouguila L, Evequoz F, Courant M, Hirsbrunner B (2004) Walking-pad: a step-in-place locomotion interface for virtual environments. In: Proceedings of the 6th international on multimodal interfaces, ACM, New York, pp 77–81. doi:[10.1145/1027933.1027948](https://doi.org/10.1145/1027933.1027948)

3. Bowman DA, Koller D, Hodges, LF (1997) Travel in immersive virtual environments: an evaluation of viewpoint motion control techniques. In: Proceedings of the virtual reality annual international symposium (VRAIS 1997), IEEE Press, Washington, pp 45–52, 215. doi:[10.1109/VRAIS.1997.583043](https://doi.org/10.1109/VRAIS.1997.583043)
4. Bowman DA, Kruijff E, LaViola JJ et al (2005) 3D user interface: theory and practice. Addison-Wesley, Boston
5. Bruder G, Steinicke F, Wieland P (2011) Self-motion illusions in immersive virtual reality environments. In: Proceedings of IEEE virtual reality, pp 39–46
6. Cirio G, Marchal M, Regia-Corte T, Lécuyer A (2009) The magic barrier tape: a novel metaphor for infinite navigation in virtual worlds with a restricted walking workspace. In: Proceedings of the ACM symposium on virtual reality software and technology (VRST 2009), pp 155–162
7. Dean GA (1965) An analysis of the energy expenditure in level and grade walking. *Ergonomics* 8(1):31–47
8. Duh HB, Parker DE, Phillips J, Furness TA (2004) Conflicting” motion cues at the frequency of crossover between the visual and vestibular self-motion systems evoke simulator sickness. *Hum Factors* 46:142–153
9. Feasel J, Wendt JD, Whitton MC (2008) LLCM-WIP: low-latency, continuous-motion walking-in-place. In: Proceedings of IEEE symposium 3D user interfaces’08, pp 97–104
10. Gibson JJ (1950) The perception of the visual world. Houghton Mifflin, Boston
11. Hodgson E, Bachmann E, Waller D (2011) Redirected walking to explore virtual environments: assessing the potential for spatial Interference. *ACM Trans Appl Percept* 8(4):1–22 (Article 22)
12. Hosman R, Van der Vaart J (1981) Effects of vestibular and visual motion perception on task performance. *Acta Psychol* 48(1–3):271–287
13. Interrante V, Ries B, Anderson L (2007) Seven league boots: a new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In: Proceedings of the IEEE symposium on 3D user interfaces, pp 167–170
14. Inman V (1981) Human walking. Williams & Wilkins, Baltimore
15. Jerald J, Peck TC, Steinicke F, Whitton MC (2008) Sensitivity to scene motion for phases of head yaws. In: Proceedings of the ACM symposium on applied perception in graphics and visualization (APGV), pp 155–162
16. Kim J-S, Gracanin D, Quek F (2012) Sensor-fusion walking-in-place interaction technique using mobile devices. In: Proceedings of IEEE virtual reality, pp 39–42
17. Konczak J (1994) Effects of optic flow on the kinematics of human gait—a comparison of young and older adults. *J Motor Behav* 26:225–236
18. Neth CT, Souman JL, Engle D, Kloos U, Bühlhoff HH, Mohler BJ (2011) Velocity-dependent dynamic curvature gain for redirected walking. In: Proceedings of IEEE virtual reality, pp 151–158
19. Nitzsche N, Hanebeck UD, Schmidt G (2004) Motion compression for telepresent walking in large target environments. *Presence Teleoper Virtual Environ* 13(1):44–60
20. Peck TC, Fuchs H, Whitton MC (2009) Evaluation of reorientation techniques and distractors for walking in large virtual environments. *IEEE Trans Vis Comput Graph* 15(3):383–394
21. Peck TC, Fuchs H, Whitton MC (2010) Improved redirection with distractors: a large-scale-real-walking locomotion interface and its effect on navigation in virtual environments. In: Proceedings of IEEE virtual reality, pp 35–38
22. Peck TC, Fuchs H, Whitton MC (2011) An evaluation of navigational ability comparing redirected free exploration with distractors to walking-in-place and joystick locomotion interfaces. In: Proceedings of IEEE virtual reality, pp 55–62
23. Peck TC, Fuchs H, Whitton MC (2012) The design and evaluation of a large-scale real-walking locomotion interface. *IEEE Trans Vis Comput Graph* 18(7):1053–1067
24. Peck T, Whitton M, Fuchs H (2008) Evaluation of reorientation techniques for walking in large virtual environments. In: Proceedings of IEEE virtual reality, pp 121–127
25. Pleban, RJ, Eakin DE, Slater MS et al (2001) Research report 1767: training and assessment of decision-making skills in virtual environments. <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA389677>. Accessed 19 Aug 2012

26. Quinn K (2011) U.S. Army to get dismounted soldier training system. Def News Train Simul J (Online). <http://www.defensenews.com/article/20110825/TJSJ01/108250309/U-S-Army-get-dismounted-soldier-training-system>. Accessed 19 Aug 2012
27. Razzaque S (2005) Redirected walking. Dissertation and computer science technical report TR05-018, University of North Carolina at Chapel Hill
28. Razzaque S, Kohn Z, Whitton MC (2001) Redirected walking. In: Proceedings of the eurographics workshop on virtual reality, pp 289–294
29. Razzaque S, Swapp D, Slater M, Whitton MC, Steed A (2002) Proceedings of the eighth eurographics workshop on virtual environments, pp 123–130
30. Robinett W, Holloway R (1992) Implementations of flying, scaling and grabbing in virtual worlds. In: ACM Symposium on interactive 3D graphics, pp 189–192
31. Rushton SK, Harris JM, Lloyd MR, Wann JP (1998) Guidance of locomotion on foot uses perceived target location rather than optic flow. *Curr Biol* 8:1191–1194
32. Steinicke F, Bruder G, Jerald J, Frenz H, Lappe M (2010a) Estimation of detection thresholds for redirected walking techniques. *IEEE Trans Vis Comput Graph* 16(1):17–27
33. Su J (2007) Motion compression for telepresence locomotion. *Presence Teleoper Virtual Environ* 16(4):385–398
34. Suma E, Clark S, Krum D, Finkelstein S, Bolas M, Warte Z (2011) Leveraging change blindness for redirection in virtual environments. In: Proceedings of IEEE virtual reality, pp 159–166
35. Slater M, Usoh M, Steed A (1995) Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Trans Comp-Hum Interact (TOCHI)* 2(3):201–219
36. Templeman JN, Denbrook PS, Sibert LE (1999) Virtual locomotion: walking in place through virtual environments. *Presence Teleoper Virtual Environ* 8(6):598–607
37. Templeman JN, Sibert LE et al (2007) Pointman—a new control for simulating tactical infantry movements. *Proc IEEE Virtual Real* 2007:285–286
38. Warren WH (2004) Optic flow. Chpt 84. MIT Press, Cambridge, pp 1247–1259
39. Warren WHJ, Kay BA, Zosh WD et al (2001) Optic flow is used to control human walking. *Nature* 4(2):213–216
40. Wendt JD (2010) Real-walking models improve walking-in-place systems. Dissertation and Computer Science Technical Report TR10-009, University of North Carolina at Chapel Hill
41. Wendt JD, Whitton MC, Brooks FP (2010) GUD-WIP: Gait-understanding driven walking-in-place. In: Proceedings of IEEE virtual reality, pp 51–58
42. Williams B, Narasimham G, McNamara TP, Carr TH, Rieser JJ, Bodenheimer B (2006) Updating orientation in large virtual environments using scaled translational gain. In: Proceedings of the 3rd ACM symposium on applied perception in graphics and visualization (APGV 2006), pp 21–28
43. Williams B, Narasimham G, Rump B, McNamara TP, Carr TH, Rieser J, Bodenheimer B (2007) Exploring large virtual environments with an HMD when physical space is limited. In: Proceedings of the 4th ACM symposium on applied perception in graphics and visualization (APGV 2007), pp 41–48
44. Whitton MC, Cohn J, Feasel J et al (2005) Comparing VE locomotion interfaces. In: Proceedings of IEEE virtual reality, pp 123–130
45. Yan L, Allision RS, Rushton SK (2004) New simple virtual walking method-walking on the spot. In: Proceedings of 9th annual immersive projection technology (IPT) symposium
46. Zheng Y, McCaleb M, Strachan C et al (2012) Exploring a virtual environment by walking in place using the Microsoft Kinect (Poster Abstract). In: Proceedings of the ACM symposium on applied perception, p 131