Taxonomy and Implementation of Redirection Techniques for Ubiquitous Passive Haptic Feedback

Frank Steinicke*, Gerd Bruder*, Luv Kohli†, Jason Jerald†, and Klaus Hinrichs*

*Visualization and Computer Graphics (VisCG) Research Group
Department of Computer Science
Westfälische Wilhelms-Universität Münster, Germany
{fsteini,g_brud01,khh}@uni-muenster.de

†Effective Virtual Environments (EVE) Group
Department of Computer Science
University of North Carolina at Chapel Hill, USA
{luv,jjerald}@cs.unc.edu

ABSTRACT

Traveling through immersive virtual environments (IVEs) by means of *real walking* is an important activity to increase naturalness of VR-based interaction. However, the size of the virtual world often exceeds the size of the tracked space so that a straightforward implementation of omni-directional and unlimited walking is not possible. *Redirected walking* is one concept to solve this problem of walking in IVEs by inconspicuously guiding the user on a physical path that may differ from the path the user visually perceives. When the user approaches a virtual object she can be redirected to a real *proxy object* that is registered to the virtual counterpart and provides passive haptic feedback. In such passive haptic environments, any number of virtual objects can be mapped to proxy objects having similar haptic properties, e.g., size, shape and texture. The user can sense a virtual object by touching its real world counterpart. Redirecting a user to a registered proxy object makes it necessary to predict the user's intended position in the IVE. Based on this target position we determine a path through the physical space such that the user is guided to the registered proxy object. We present a taxonomy of possible redirection techniques that enable user guidance such that inconsistencies between visual and proprioceptive stimuli are imperceptible. We describe how a user's target in the virtual world can be predicted reliably and how a corresponding real-world path to the registered proxy object can be derived.

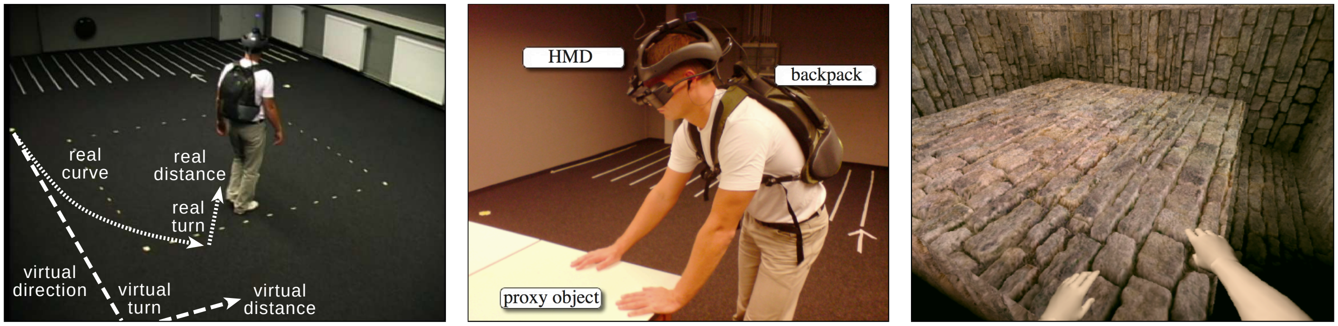
Keywords: Virtual Reality, Locomotion Interface, Generic Redirected Walking, Dynamic Passive Haptics

1 INTRODUCTION

Walking is the most basic and intuitive way of moving within the real world. Keeping such an active and dynamic ability to navigate through large-scale immersive virtual environments (IVEs) is of great interest for many 3D applications demanding locomotion, such as in urban planning, tourism, 3D entertainment etc. Head-mounted display (HMD) and tracking system represent typical instrumentation of an IVE. Although many domains are inherently three-dimensional and advanced visual simulations often provide a good sense of locomotion, most applications do not support VR-based user interfaces, least of all *real walking* is possible [33]. However, real walking in IVEs can be realized. An obvious approach is to transfer the user's head movements to changes of the virtual camera in the IVE by means of a one-to-one mapping. This technique has the drawback that the user's movements are restricted by the limited range of the tracking sensors and a rather small workspace in the real world. Therefore concepts for virtual locomotion interfaces are needed that enable walking over large distances in the virtual world while remaining within a relatively small space in the real world.

Many hardware-based approaches have been presented to address this issue [1, 15, 16, 26]. Since most of them are very costly and support only walking of a single user they may not get beyond a prototype stage. However, cognition and perception research suggests that more cost-efficient alternatives exist. Psychologists have known for decades that vision usually dominates proprioceptive, i. e., vestibular and kinesthetic, sensation when the two disagree [7]. While graphics may provide correct visual stimuli of motion in the IVE, it can only approximate proprioceptive stimuli. Experiments demonstrate that the user tolerates a certain amount of inconsistency between visual and proprioceptive sensation [28, 32, 17, 22, 18, 4, 24]. Moreover users tend to unwittingly compensate for small inconsistencies making it possible to guide them along paths in the real world which differ from the path perceived in the virtual world. This so-called *redirected walking* enables users to explore a virtual world that is considerably larger than the tracked lab space [24] (see Figure 1 (a)).

Besides natural navigation, multi-sensory perception of an IVE increases the degree of presence [10]. Whereas graphics and sound rendering have matured so much that realistic synthesis of real world scenarios is possible, generation of haptic stimuli still represents a vast area for research. Tremendous effort has been undertaken to support *active* haptic feedback by specialized hardware which generates certain haptic stimuli [5]. These technologies such as force feedback devices can provide compelling haptic feedback, but are expensive and limit the size of the user's working space due to devices and wires. A simpler solution is to use *passive* haptic feedback: physical props registered to virtual objects provide real haptic feedback to the user. By touching such a prop the user gets the impression of interacting with an associated virtual object seen in an HMD [19] (see Figure 1 (b)). Passive haptic feedback is very compelling, but a different physical object is needed for each virtual object requiring haptic feedback [9]. Since the interaction space is constrained, only a few physical props can be supported, thus the number of virtual objects that can be touched by the user is limited. Moreover, the presence of physical props in the interaction space prevents exploration of other parts of the virtual world not represented by the current physical setup. Thus exploration of large scale environments and support of passive haptic feedback seem to



(a) **(b)** **(c)**

Figure 1: Combining several redirection techniques and dynamic passive haptics. (a) A user walks in the physical environment on a path that is different from the visually perceived path. (b) A user touches a table serving as proxy object for (c) a stone block displayed in the virtual world.

be mutually exclusive.

Recently redirected walking and passive haptics have been combined in order to address both problems [18, 28]. If the user approaches an object in the virtual world she is guided to a corresponding physical prop. Otherwise the user is guided around obstacles in the working space in order to avoid collisions. Props do not have to be aligned with their virtual world counterparts nor do they have to provide haptic feedback identical to the visual representation. Experiments have shown that physical objects can provide passive haptic feedback for virtual objects with a different visual appearance and with similar, but not necessarily the same, haptic capabilities [28] (see Figure 1 (b) and (c)). Hence, virtual objects can be sensed by means of real *proxy props* having similar haptic properties, i. e., size, shape and texture. The mapping from virtual to real objects need not be one-to-one. Since the mapping as well as the visualization of virtual objects can be changed *dynamically* during runtime, usually a small number of proxy props suffices to represent a much larger number of virtual objects. By redirecting the user to a preassigned proxy object that represents a virtual counterpart, the user gets the illusion of interacting with a desired virtual object.

We present a taxonomy of potential redirection techniques which guide users to corresponding proxy props, and we show how the required transformation of virtual to real paths can be implemented. The remainder of this paper is structured as follows. Section 2 summarizes previous work about redirection techniques and passive haptic feedback. Section 3 provides a taxonomy of redirection techniques which can be used to guide users to registered proxy props. Section 4 explains how a virtual path is mapped to a physical path on which users are guided. Section 5 concludes the paper and gives an overview about future work.

2 Previous Work

Currently locomotion and perception in virtual worlds are in the focus of many research groups. To address natural walking in IVEs, various prototypes of interface devices have been developed to prevent a displacement in the real world. These devices include torus-shaped omni-directional treadmills [1, 2], motion foot pads [15], robot tiles [14, 16] and motion carpets [27]. All these systems are costly and support only a single user. For multi-walker scenarios, it is necessary to equip each user with a separate device. Although these hardware systems represent enormous technological achievements, most likely they will not get beyond a prototype stage in the foreseeable future due to the described limitations. Hence there is a tremendous demand for alternative approaches.

As a solution to this challenge, traveling by exploiting walk-like gestures has been proposed in many different variants, giving the user the impression of walking. For example, the walking-in-place approach exploits walk-like gestures to travel through an IVE, while the user remains physically at nearly the same position [13, 31, 27, 29, 34, 6]. Real walking has been shown to be a more presence-enhancing locomotion technique than any other navigation metaphors [31].

Research has analyzed perception in both real as well as virtual worlds. For example, many researchers have described that distances in virtual worlds are underestimated in comparison to the real world [11, 12]. Furthermore, it has been discovered that users have difficulty orienting themselves in virtual worlds [25].

Visual dominance over the proprioception has been examined for hand-based interaction tasks [4]. Redirected walking [24] is a promising solution to the problem of limited tracking space and the challenge of providing users with the ability to explore an IVE by walking. The technique redirects the user by manipulating the displayed scene, causing users to unknowingly compensate by repositioning or reorienting themselves.

Different approaches to redirect a user in an IVE have been suggested. The most common approach is to scale translational movements, for example, to cover a virtual distance that is larger than the distance walked in the physical space. Interrante et al. suggest to apply the scaling exclusively to the main walking direction in order to prevent unintended lateral shifts [13]. With most reorientation techniques, the virtual world is imperceptibly rotated around the center of a stationary user until she is oriented such that no physical obstacles are in front of her [22, 24, 18]. Then, the user can continue to walk in the desired virtual direction.

Alternatively, reorientation can also be applied while the user walks [8, 28]. For instance, if the user wants to walk straight ahead for a long distance in the virtual world, small rotations of the camera redirect her to walk unconsciously on a circular arc in the opposite direction in the real world. When redirecting a user, the visual sensation is consistent with visual motion in the IVE, but proprioceptive sensation reflects motion in the physical world. However, if the induced manipulations are small enough, the user has the impression of being able to walk in the virtual world in any direction without restrictions. Until now not much research has been undertaken in order to identify thresholds which indicate the tolerable amount of deviation between vision and proprioception [32, 28, 17]. Redirection techniques have been applied particularly in the field of robotics for controlling a remote robot by walking [8]. For such scenarios much effort has been undertaken to prevent collisions–sophisticated path prediction is therefore essential [8, 21]. These techniques guide users on physical paths for which lengths as well as turning angles of the visually perceived paths are maintained. Hence, omni-directional and unlimited walking is possible. However, passive haptics feedback has not been considered in this context.

Active haptic feedback is often supported by expensive haptic hardware, such as Phantom devices [5] or specialized data gloves, but only few devices can be worn comfortably without any wires that provide at least a sufficient sense of touch. Passive haptic feedback has been used effectively to provide the natural sensation of touch [10]. The main idea is to replicate counterparts of virtual objects such as walls and tables in the physical space and to arrange them correspondingly. It has been shown that this increases the immersion in the IVE significantly [31, 9]. As mentioned in Section 1, the mapping between virtual objects and proxy props need not necessarily be one-to-one. In this context McNeely has presented the concept of robotic graphics [20]. The main idea is that a robot is equipped with a haptic feedback device attached to its end effector. The robot takes the device to the location where the haptic feedback should be presented. This concept has been extended by Tachi et al. with their Shape Approximation Device [30]. The device can exchange the surface touched by the user's finger, and hence different shapes and textures can be simulated. Kohli et al. suggest the inverse idea [18]. They use a static proxy prop to provide passive haptic feedback for several virtual objects. Their prototype setup was limited to symmetrical cylinders, but recent research results indicate that visual and kinesthetic information may be discrepant without users observing the inconsistencies [28, 17, 4].

In summary, considerable effort has been undertaken in order to enable a user to walk through a large-scale IVE while presenting continuous passive haptic stimuli.

3 TAXONOMY OF REDIRECTION TECHNIQUES

A fundamental task of an IVE is to synchronize images presented on the display surface with the user's head movements in such a way that the elements of the virtual scene appear stable in space. Redirected walking and reorientation techniques take advantage of the imperfections of the human visual-vestibular system by intentionally injecting imperceivable motions of the scene. When a user navigates through an IVE by means of real walking, motions are composed of translational and rotational movements. Translational movements are used to get from one position to another, rotational movements are used to reorient in the IVE. By combining both types of movements users can navigate on curve-like trajectories. We classify redirection techniques with respect to these types.

3.1 User's Locomotion Triple

Redirected walking can be applied via *gains* which define how tracked real-world movements are mapped to the virtual environment. These gains are specified with respect to a coordinate system. For example, gains can be applied by uniform or non-uniform scaling factors applied to the scene coordinate system. Previous research approaches suggest defining locomotion gains with respect to the user's walking direction [13].

We introduce the user's *locomotion triple* (s,u,w) defined by three vectors: the *strafe* vector s, the *up* vector u and the *direction of walk* vector w. The user's direction of walk can be determined by the actual walking direction or using proprioceptive information such as the orientation of the limbs or the viewing direction. In our implementation we define w by the actual walking direction tracked by the tracking system. The strafe vector is orthogonal to the direction of walk and parallel to the walking plane. Since from the user's perspective the strafe vector points to the right, it is sometimes denoted as *right vector*. While the direction of walk and the strafe vector are orthogonal to each other, the up vector u is not constrained to the crossproduct $s \times w$. For instance, if a user walks a slope the user's direction of walk is defined according to the walking plane's orientation, whereas the up vector is not orthogonal to the tilted walking plane. When walking on slopes humans tend to lean forward, so the up vector remains orthogonal to the virtual world's (x,z)-plane. Even on tilted planes the user's up vector may be defined by $s \times w$. This can be useful, for example, if the user is located in another reference system, such as driving a car. However, while walking the user's up vector is usually given by the inverse of the gravitation direction, e. g., the scene's up vector.

In the following sections we describe how gains can be applied to such a locomotion triple.

3.2 Translation gains

Assume that the tracking and virtual world coordinate systems are calibrated and registered. When the tracking system detects a change of the user's position defined by the vector $translation = pos_{cur} - pos_{pre}$, where pos_{cur} is the current position and pos_{pre} is the previous position, $translation$ is applied one-to-one to the virtual camera, i. e., the virtual camera is moved by $|translation|$ units in the corresponding direction in the virtual world coordinate system. The tracking system updates the change of position several times per second as long as the user remains within the range of the tracking system.

A translation gain $g_{trans} \in \mathbb{R}$ is defined by the quotient of the applied virtual world translation $translation_{\text{virtual}}$ and the tracked real world translation $translation_{\text{real}}$, i. e., $g_{trans} := \frac{translation_{\text{virtual}}}{translation_{\text{real}}}$. When a translation gain g_{trans} is applied to a translational movement $translation_{\text{real}}$ the virtual camera is moved by the vector $g_{trans} \cdot translation_{\text{real}}$ in the corresponding direction. This is particularly useful if the user wants to explore IVEs whose size differs significantly from the size of the tracked space. For instance, if a user wants to explore molecular structures, movements in the real world must be scaled down when they are mapped to virtual movements, e. g., $g_{trans} \approx 0$. In contrast, the exploration of a football field by means of real walking in a working space requires a translation gain $g_{trans} \approx 10$. Such uniform gains allow exploration of IVEs whose sizes differ from the size of the working space, but often restrict natural movements.

Besides scaling movements in the direction of walk, lateral and vertical movements are affected by uniform gains. In most VR-based scenarios users benefit from the ability to explore close objects via head movements which may be hindered by scaling vertical or lateral movements, and therefore uniform gains are often inadequate. Non-uniform translation gains are used to distinguish between movements in the main walking direction, lateral movements and vertical movements [11]. Translation gains are defined with respect to the user's locomotion triple (see Section 3.1) and are designated by $g_{trans_s}, g_{trans_w}, g_{trans_u}$, where each component is applied to its corresponding vector s, u or w.

3.3 Rotation gains

A real-world head turn can be specified by a vector consisting of three angles, i. e., $rotation := (yaw, pitch, roll)$. The tracked orientation change is applied to the virtual camera.

Analog to translation gains, a rotation gain g_{rot} is defined by the quotient of the considered component (yaw/pitch/roll) of a virtual world rotation $rotation_{\text{virtual}}$ and the real world rotation $rotation_{\text{real}}$, i. e., $g_{rot} := \frac{rotation_{\text{virtual}}}{rotation_{\text{real}}}$ and $rotation \in \{yaw, pitch, roll\}$. When a rotation gain $g_{rotation}$ is applied to a real world rotation α the virtual camera is rotated by $g_{rotation} \cdot \alpha$ instead of α. This means that if $g_{rot} = 1$ the virtual scene remains stable considering the head's orientation change. For $g_{rot} > 1$ the virtual scene appears to rotate against the direction of the head turn, and $g_{rot} < 1$ causes the scene to rotate in the direction of the head turn. For instance, if the user rotates her head by 90° degree, a gain $g_{rot} = 1$ maps this motion one-to-one to the VE. The appliance of a gain $g_{rot} = 0.5$ means that the user has to rotate the head by 180° physically in order to achieve a 90° virtual rotation; a gain $g_{rot} = 2$ means that the user has to rotate the head by 45° physically in order to achieve a 90°

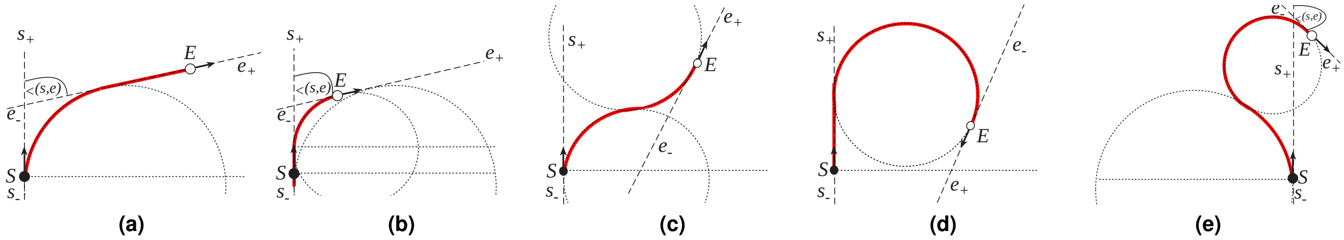


Figure 2: Generated paths for different poses of start point S and end point E.

virtual rotation. Again, gains are defined for each component of the rotation, i. e., yaw, pitch, and roll, and are applied to the axes of the locomotion triple.

Thus, generic gains for rotational movements can be expressed by $g_{rot_s}, g_{rot_w}, g_{rot_u}$, where the gain g_{rot_w} specified for roll is applied to w, the gain g_{rot_s} specified for pitch is applied to s and the gain g_{rot_u} specified for yaw is applied to u.

3.4 Curvature gains

Instead of multiplying gains to translational or rotational movements, they can be added as offsets to real world movements. Camera manipulations are applied if the user turns the head, but does not move, or the user moves straight without turning her head. If the camera manipulations are reasonably small, the user will unknowingly compensate for these offsets and walk on a curve. The gains can be applied in order to inject rotations, while users virtually walk straight, or they can be applied as offsets, while users only rotate their heads.

The curvature gain g_{cur} denotes the bending of a real path. For example, when the user moves straight ahead a curvature gain that causes reasonably small iterative camera rotations to one side forces the user to walk along a curve in the opposite direction in order to stay on a straight path in the virtual world. The curve is determined by a circular arc with radius r, where $g_{cur} := \frac{1}{r}$. The resulting curve is considered for a reference distance of $\frac{\pi}{2}$ meters. In the case that no curvature is applied $r = \infty$ and $g_{cur} = 0$, whereas if the curvature causes the user to rotate by $90°$ clockwise after $\frac{\pi}{2}$ meters the user has covered a quarter circle and $g_{cur} = 1$. Alternatively, a curvature gain can be applied as translation offset while the user turns the head and no translational movements are intended.

While the user turns, such a gain causes the camera to shift to one direction. This camera shift prompts the user to unknowingly move into the opposite direction in order to compensate an unintended displacement in the virtual world. Potentially, such gains can be applied to each permutation of axes of the locomotion triple. However, the common procedure is to enforce users to walk on a curve as described above.

4 Implementation of Redirection Techniques

In this section we present how the redirection techniques described in Section 3 can be implemented such that users are guided to particular locations in the physical space, e g., proxy props, in order to support passive haptic feedback. This is done by applying the gains to tracked data as described in Section 2. Therefore, we explain how a virtual path along which a user walks in the IVE is transformed to a path on which the user actually walks in the real world (see Figure 2).

4.1 Target Prediction

Before a user can be redirected to a proxy prop, the target in the virtual world which is represented by the prop has to be predicted. In most redirection techniques [21, 24, 29] only the walking direction is considered for the prediction procedure.

In contrast to these approaches our implementation also takes into account the viewing direction. The current direction of walk determines the predicted path, and the viewing direction is used for verification: if both vector's projections to the walking plane differ by more than $45°$, no reliable prediction can be made. For short-term path prediction in such a scenario the user seems to move around without specific target. Hence the user is only redirected in order to avoid a collision in the physical space or when she might leave the tracking area.

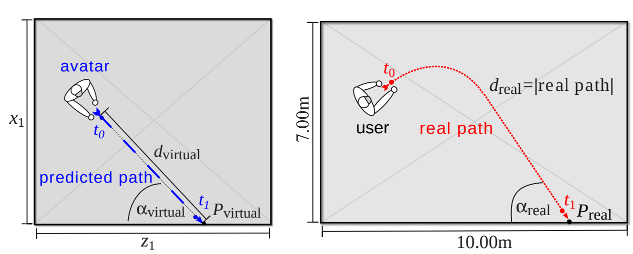
In order to prevent collisions in the physical space only the walking direction has to be considered because the user does not see the physical space due to the HMD. Therefore redirection is not necessary in order to prevent collisions in the physical world.

When the angle between the vectors projected onto the walking plane is sufficiently small ($< 45°$), the walking direction defines the predicted path. In this case a half-line s_+ extending from the current position S in the walking direction (see Figure 2) is tested for intersections with virtual objects in the user's frustum. These objects are defined in terms of their position, orientation and size in a corresponding scene description file. We use an XML-based description as explained in Section 4.5. The collision detection is realized by means of a ray shooting similar to the approaches referenced in [23]. For simplicity we consider only the first object hit by the walking direction w. We approximate each virtual object that provides passive feedback by a 2D bounding box. Since these boxes are stored in a quadtree-like data structure the intersection test can be performed in real-time (see Section 4.5).

As illustrated in Figure 3 (a) if an intersection is detected, we store the target object, the intersection angle α_{virtual}, the distance to the intersection point d_{virtual}, and the relative position of the intersection point P_{virtual} on the edge of the bounding box. From these values we can calculate all data required for the path transformation process as described in the following section.

4.2 Path Transformation

In robotics techniques for autonomous robots have been developed to compute a path through several interpolation points [21, 8]. How-



(a) Virtual environment **(b)** Real environment

Figure 3: Redirection technique: (a) a user in the virtual world approaches a virtual wall such that (b) she is guided to the corresponding proxy object, i. e., a real wall in the physical space.

ever, these techniques are optimized for static environments, and highly-dynamic scenes, where an update of the transformed path occurs approximately 30 times per second, are not considered [29]. Since the XML-based description contains the initial orientation between virtual objects and proxy props, it is possible to redirect a user to the desired proxy prop such that the haptic feedback is consistent with her visual perception. Fast memory access and simple calculations enable consistent passive feedback.

As mentioned above, we predict the intersection angle α_{virtual}, the distance to the intersection point d_{virtual}, and the relative position of the intersection point P_{virtual} on the edge of the bounding box of the virtual object. These values define the target pose E, i. e., position and orientation in the physical world, with respect to the associated proxy prop (see Figure 2). The main goal of redirected walking is to guide the user along a real world path (from S to E) which varies as little as possible from the visually perceived path, i. e., ideally a straight line in the physical world from the current position to the predicted target location. The real world path is determined by the parameters α_{real}, d_{real} and P_{real}. These parameters are calculated from the corresponding parameters α_{virtual}, d_{virtual} and P_{virtual} in such a way that consistent haptic feedback is ensured. Due to many tracking events per second the start and end points change during a walk, but smooth paths are guaranteed by our approach.

We ensure a smooth path by constraining the path parameters such that the path is C^1-continuous, starts at the start pose S, and ends at the end pose E. A C^1-continuous composition of line segments and circular arcs is determined from the corresponding path parameters for the physical path, i. e. α_{real}, d_{real} and P_{real} (see Figure 3 (b)). The trajectories in the real world can be computed as illustrated in Figure 2, considering the start pose S together with the line s through S parallel to the direction of walk in S, and the end pose E together with the line e through E parallel to the direction of walk in E. With s_+ resp. e_+ we denote the half-line of s resp. e extending from S resp. E in the direction of walk, and with s_- resp. e_- the other half-line of s resp. e.

In Figure 2 different situations are illustrated that may occur for the orientation between S and E. For instance, if s_+ intersects e_- and the intersection angle satisfies $0 < \angle(s,e) < \pi/2$ as depicted in Figure 2 (a) and (b), the path on which we guide the user from S to E is composed of a line segment and a circular arc. The center of the circle is located on the line through S and orthogonal to s, its radius is chosen in such a way that e is tangent to the circle. Depending on whether e_+ or e_- touches the circle, the user is guided on a line segment first and then on a circular arc or vice versa. If s_+ does not intersect e_- two different cases are considered: e_- intersects s_- or not. If an intersection occurs the path is composed of two circular arcs that are constrained to have tangents s and e and to intersect in one point as illustrated in Figure 2 (c). If no intersection occurs (see Figure 2 (d)) the path is composed of a line segment and a circular arc similar to Figure 2 (a). However, if the radius of one of the circles gets too small, i. e., the curvature gets too large, an additional circular arc is inserted into the path as illustrated in Figure 2 (e). All other cases can be derived by symmetrical arrangements or by compositions of the described cases.

Figure 3 shows how a path is transformed using the described approaches in order to guide the user to the predicted target proxy prop, i. e., a physical wall. In Figure 3 (a) an IVE is illustrated. Assuming that the angle between the projections of the viewing direction and direction of walking onto the walking plane is sufficiently small (see Section 4.3), the desired target location in the IVE is determined as described in Section 4.3. The target location is denoted by point P_{virtual} at the bottom wall. Moreover, the intersection angle α_{virtual} as well as the distance d_{virtual} to P_{virtual} are calculated. The registration of each virtual object to a physical proxy prop allows the system to determine the corresponding values P_{real}, α_{real} and

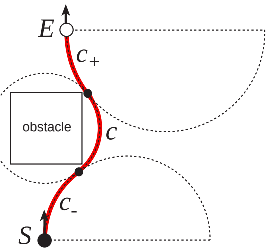


Figure 4: Corresponding paths around a physical obstacle between start- and endpoint poses S and E.

d_{real}, and thus to derive start and end pose S and E are derived. A corresponding path as illustrated in Figure 3 is composed like the paths shown in Figure2.

4.3 Physical Obstacles

When guiding a user through the real world, collisions with the physical setup have to be prevented. Collisions in the real world are predicted similarly to those in the virtual world based on the direction of walk and ray shooting approaches as described above. A ray is cast in the direction of walk and tested for intersection with real world objects represented in the XML-based description (see Section 4.5). If such a collision is predicted a reasonable bypass around an obstacle is determined as illustrated in Figure 4. The previous path between S and E is replaced by a chain of three circular arcs: a segment c of a circle which encloses the entire bounding box of the obstacle, and two additional circular arcs c_+ and c_-. The circles corresponding to these two segments are constrained to touch the circle around the obstacle. Circular arc c is bounded by the two touching points, c_- is bounded by one of the touching points and S and c_+ by the other touching point and E.

4.4 Score Function

In the previous sections we have described how a real-word path can be generated such that a user is guided to a registered proxy prop and unintended collisions in the real world are avoided. Actually, it is possible to represent a virtual path by many different physical paths. In order to select the best transformed path we define a score function for each considered path. The score function expresses the quality of paths in terms of matching visual and vestibular/proprioceptive cues:
First, we define

$$scale := \begin{cases} \frac{d_{\text{virtual}}}{d_{\text{real}}} - 1, \text{ if } d_{\text{virtual}} > d_{\text{real}} \\ \frac{d_{\text{real}}}{d_{\text{virtual}}} - 1, \text{ otherwise} \end{cases}$$

with the length of the virtual path $d_{\text{virtual}} > 0$ and the length of the transformed real path $d_{\text{real}} > 0$. Case differentiation is done in order to weight up- and downscaling equivalently. Furthermore we define the terms

$$\begin{aligned} t_1 &:= 1 + c_1 \cdot maxCurvature^2 \\ t_2 &:= 1 + c_2 \cdot avgCurvature^2 \\ t_3 &:= 1 + c_3 \cdot scale^2 \end{aligned}$$

where *maxCurvature* denotes the maximal and *avgCurvature* denotes the average curvature of the entire physical path. The constants c_1, c_2 and c_3 can be used to weight the terms in order to adjust the terms to the user's sensitivity. For example, if a user is susceptible to curvatures, c_1 and c_2 can be increased in order to give the corresponding terms more weight. In our setup we use

```
    ...
    <worldData>
      <objects number ="3">
        <object0>
 5        <boundingBox>
            <vertex0 x="6.0" y="7.0"></vertex0>
            <vertex1 x="6.0" y="8.5"></vertex1>
            <vertex2 x="8.5" y="8.5"></vertex2>
            <vertex3 x="8.5" y="7.0"></vertex3>
10        </boundingBox>
          <vertices>
            <vertex0 x="6.1" y="7.1"></vertex0>
            <vertex1 x="6.1" y="8.4"></vertex1>
            <vertex2 x="8.4" y="8.4"></vertex2>
15          <vertex3 x="8.4" y="7.1"></vertex3>
          </vertices>
        </object0>
      ...
      <borders>
20      <vertex0 x="0.0" y="0.0"></vertex0>
        <vertex1 x="0.0" y="9.0"></vertex1>
        <vertex2 x="9.0" y="9.0"></vertex2>
        <vertex3 x="9.0" y="0.0"></vertex3>
      </borders>
25  ...
```

Listing 1: Line-based description of the real world in XML format.

$c_1 = c_2 = 0.4$ and $c_3 = 0.2$. With these definitions we specify the score function as

$$score := \frac{1}{t_1 \cdot t_2 \cdot t_3} \tag{1}$$

This function satisfies $0 \leq score \leq 1$ for all paths. If $score = 1$ for a transformed path, the predicted virtual path and the transformed path are equal. With increasing differences between virtual and transformed path, the score function decreases and approaches zero. In our experiments most paths generated as described above achieve scores between 0.4 and 0.9 with an average score of 0.74. Rotation gains are not considered in the score function since when the user turns the head no path needs to be transformed in order to guide a user to a proxy prop.

4.5 Virtual and Real Scene Description

In order to register proxy props with virtual objects we represent the virtual and the physical world by means of an XML-based description in which *all* objects are discretized by a polyhedral representation, e. g., 2D bounding boxes. The degree of approximation is defined by the level of discretization set by the developer. Each real as well as virtual object is composed of line segments representing the edges of their bounding boxes. As mentioned in Section 2 the position, orientation and size of a proxy prop need not match these characteristics exactly. For most scenarios a certain deviation is not noticeable by the user when she touches proxy props, and both worlds are perceived as congruent. If tracked proxy props or registered virtual objects are moved within the working space or the virtual world, respectively, changes of their poses are updated in our XML-based description. Thus, also dynamic scenarios where the virtual and the physical environment may change are considered in our approach.

In Listing 1 part of an XML-based description specifying a virtual world is shown. In lines 5-10 the bounding box of a real world object is defined. The borders of the entire tracking space are defined by means of a rectangular area in lines 19-24.

In Listing 2 part of an XML-based description of a working space is illustrated. In lines 5-10 the bounding box of a virtual world object is defined. The registration between this object and

```
    ...
    <worldData>
      <objects number ="3">
        <object0>
 5        <boundingBox>
            <vertex0 x="0.5" y="7.0"></vertex0>
            <vertex1 x="0.5" y="9.5"></vertex1>
            <vertex2 x="2.0" y="9.5"></vertex2>
            <vertex3 x="2.0" y="7.0"></vertex3>
10        </boundingBox>
          <vertices>
            <vertex0 x="1.9" y="7.1"></vertex0>
            <vertex1 x="0.6" y="7.1"></vertex1>
            <vertex2 x="0.6" y="9.4"></vertex2>
15          <vertex3 x="1.9" y="9.4"></vertex3>
          </vertices>
          <relatedObjects number="1" obj0="0">
          </relatedObjects>
          ...
```

Listing 2: Line-based description of virtual world in XML format.

proxy props is defined in line 17. The field `relatedObjects` specifies the number as well as the objects which serve as proxy props.

5 Conclusions

In this paper we presented a taxonomy of redirection techniques in order to support ubiquitous passive haptic environments. Furthermore, we have described how we have implemented these concepts. When our redirection concepts are used in our laboratory environment, users usually do not observe inconsistencies between visual and vestibular cues.

Currently, the tested setup consists of a cuboid-shaped tracked working space ($10 \times 7 \times 2.5$ meters) and a real table serving as proxy prop for virtual blocks, tables etc. With increasing number of virtual objects and proxy props more rigorous redirection concepts have to be applied, and users tend to recognize the inconsistencies more often. However, first experiments in this setup show that it becomes possible to explore arbitrary IVEs by real walking, while consistent passive haptic feedback is provided. Users can navigate within arbitrarily sized IVEs by remaining in a comparably small physical space, where virtual objects can be touched. Indeed, unpredicted changes of the user's motion may result in strongly curved paths, and the user will recognize this. Moreover, significant inconsistencies between vision and proprioception may cause cyber sickness [3].

We believe that redirected walking combined with passive haptic feedback is a promising solution to make exploration of IVEs more ubiquitously available, e. g., when navigating in existing applications such as Google Earth or multiplayer online games. One drawback of our approach is that proxy objects have to be associated manually to their virtual counterparts. This information could be derived from the virtual scene description automatically. When the HMD is equipped with a camera, computer vision techniques could be applied in order to extract information about the IVE and the real world automatically. Furthermore we have to evaluate in how far visual representation and passive haptic feedback of proxy props may differ.

References

[1] L. Bouguila and M. Sato. Virtual Locomotion System for Large-Scale Virtual Environment. In *Proceedings of Virtual Reality*, pages 291–292. IEEE, 2002.

[2] L. Bouguila, M. Sato, S. Hasegawa, H. Naoki, N. Matsumoto, A. Toyama, J. Ezzine, and D. Maghrebi. A New Step-in-Place Loco-

motion Interface for Virtual Environment with Large Display System. In *Proceedings of SIGGRAPH*, pages 63–63. ACM, 2002.

[3] D. Bowman, D. Koller, and L. Hodges. Travel in Immersive Virtual Environments: An Evaluation of Viewpoint Motion Control Techniques. In *Proceedings of VRAIS'97*, volume 7, pages 45–52. IEEE, 1997.

[4] E. Burns, S. Razzaque, A. T. Panter, M. Whitton, M. McCallus, and F. Brooks. The Hand is Slower than the Eye: A Quantitative Exploration of Visual Dominance over Proprioception. In *Proceedings of Virtual Reality*, pages 3–10. IEEE, 2005.

[5] M. Calis. Haptics. Technical report, Heriot-Watt University, 2005.

[6] J. Feasel, M. Whitton, and J. Wendt. Llcm-wip: Low-latency, continuous-motion walking-in-place. In *Proceedings of IEEE Symposium on 3D User Interfaces 2008*, pages 97–104, 2008.

[7] J. Gibson. Adaptation, after-effect and contrast in the perception of curved lines. *Journal of Experimental Psychology*, 16(1):1–31, 1993.

[8] H. Groenda, F. Nowak, P. Rößler, and U. D. Hanebeck. Telepresence Techniques for Controlling Avatar Motion in First Person Games. In *Intelligent Technologies for Interactive Entertainment (INTETAIN 2005)*, pages 44–53, 2005.

[9] B. Insko. *Passive Haptics Significantly Enhances Virtual Environments*. PhD thesis, Department of Computer Science, University of North Carolina at Chapel Hill, 2001.

[10] B. Insko, M. Meehan, M. Whitton, and F. Brooks. Passive Haptics Significantly Enhances Virtual Environments. In *Proceedings of 4th Annual Presence Workshop*, 2001.

[11] V. Interrante, L. Anderson, and B. Ries. Distance Perception in Immersive Virtual Environments, Revisited. In *Proceedings of Virtual Reality*, pages 3–10. IEEE, 2006.

[12] V. Interrante, B. Ries, J. Lindquist, and L. Anderson. Elucidating the Factors that can Facilitate Veridical Spatial Perception in Immersive Virtual Environments. In *Proceedings of Virtual Reality*. IEEE, 2007.

[13] V. Interrante, B. Riesand, and L. Anderson. Seven League Boots: A New Metaphor for Augmented Locomotion through Moderately Large Scale Immersive Virtual Environments. In *Proceedings of Symposium on 3D User Interfaces*, pages 167–170. IEEE, 2007.

[14] H. Iwata. The Trous Treadmill: Realizing Locomotion in VEs. *IEEE Computer Graphics and Applications*, 9(6):30–35, 1999.

[15] H. Iwata, Y. Hiroaki, and H. Tomioka. Powered Shoes. *SIGGRAPH 2006 Emerging Technologies*, (28), 2006.

[16] H. Iwata, H. Yano, H. Fukushima, and H. Noma. CirculaFloor. *IEEE Computer Graphics and Applications*, 25(1):64–67, 2005.

[17] J. Jerald, T. Peck, F. Steinicke, and M. Whitton. Sensitivity to scene motion for phases of head yaws. In *ACM Proceedings of Applied Perception in Visualzation and Graphics*, (in press), 2008.

[18] L. Kohli, E. Burns, D. Miller, and H. Fuchs. Combining Passive Haptics with Redirected Walking. In *Proceedings of Conference on Augmented Tele-Existence*, volume 157, pages 253 – 254. ACM, 2005.

[19] R. W. Lindeman. *Bimanual Interaction, Passive-Haptic Feedback, 3D Widget Representation, and Simulated Surface Constraints for Interaction in Immersive Virtual Environments*. PhD thesis, The George Washington University, Department of EE & CS, 1999.

[20] W. A. McNeely. Robotic graphics: A new approach to force feedback for virtual reality. In *Proceedings of IEEE Virtual Reality Annual International Symposium (VRAIS)*, pages 336–341, 1993.

[21] N. Nitzsche, U. Hanebeck, and G. Schmidt. Motion Compression for Telepresent Walking in Large Target Environments. In *Presence*, volume 13, pages 44–60, 2004.

[22] T. Peck, M. Whitton, and H. Fuchs. Evaluation of reorientation techniques for walking in large virtual environments. In *Proceedings of IEEE Virtual Reality (VR)*, pages 121–128, 2008.

[23] M. Pellegrini. Ray Shooting and Lines in Space. *Handbook of discrete and computational geometry*, pages 599–614, 1997.

[24] S. Razzaque, Z. Kohn, and M. Whitton. Redirected Walking. In *Proceedings of Eurographics*, pages 289–294. ACM, 2001.

[25] B. Riecke and J. Wiener. Can People not Tell Left from Right in VR? Point-to-Origin Studies Revealed Qualitative Errors in Visual Path Integration. In *Proceedings of Virtual Reality*, pages 3–10. IEEE, 2007.

[26] M. Schwaiger, T. Thümmel, and H. Ulbrich. A 2d-motion platform: The cybercarpet. In *Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 2007.

[27] M. Schwaiger, T. Thümmel, and H. Ulbrich. Cyberwalk: Implementation of a Ball Bearing Platform for Humans. In *Proceedings of Human-Computer Interaction*, pages 926–935, 2007.

[28] F. Steinicke, G. Bruder, T. Ropinski, and K.Hinrichs. Moving towards generally applicable redirected walking. In *Proceedings of the Virtual Reality International Conference (VRIC)*, pages 15–24, 2008.

[29] J. Su. Motion Compression for Telepresence Locomotion. *Presence: Teleoperator in Virtual Environments*, 4(16):385–398, 2007.

[30] S. Tachi, Maeda, R. Hirata, and H. Hoshino. A construction method of virtual haptic space. In *Proceedings of International Conference on Artificial Reality and Tele-existence (ICAT)*, pages 131–138, 1994.

[31] M. Usoh, K. Arthur, M. Whitton, R. Bastos, A. Steed, M. Slater, and F. Brooks. Walking $>$ Walking-in-Place $>$ Flying, in Virtual Environments. In *International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 359 – 364. ACM, 1999.

[32] H. Wallach. Perceiving a stable environment when one moves. *Anual Review of Psychology*, 38:127, 1987.

[33] M. Whitton, J. Cohn, P. Feasel, S. Zimmons, S. Razzaque, B. Poulton, and B. M. und F. Brooks. Comparing VE Locomotion Interfaces. In *Proceedings of Virtual Reality*, pages 123–130. IEEE, 2005.

[34] B. Williams, G. Narasimham, T. P. McNamara, T. H. Carr, J. J. Rieser, and B. Bodenheimer. Updating Orientation in Large Virtual Environments using Scaled Translational Gain. In *Proceedings of the 3rd Symposium on Applied Perception in Graphics and Visualization*, volume 153, pages 21–28. ACM, 2006.