

# Motion Compression for Telepresent Walking in Large Target Environments

---

## Abstract

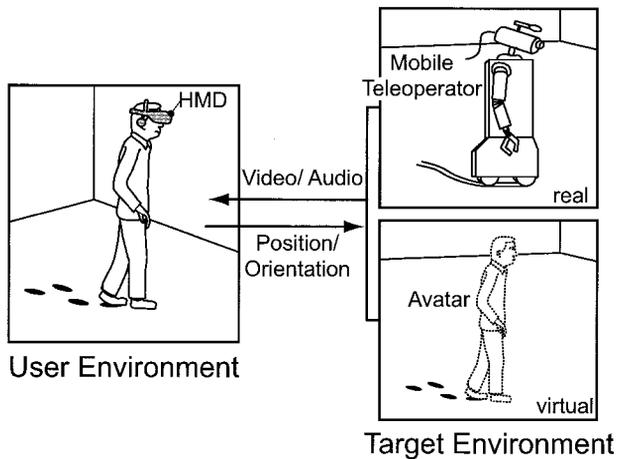
Telepresent walking allows visits to remote places such as museums, exhibitions, architecture, or industrial sites with a high degree of realism. While walking freely around in the user environment, the user sees the remote environment through the “eyes” of a remote mobile teleoperator. For that purpose, the user’s motion is tracked and transferred to the teleoperator. Without additional processing of the motion data, the size of the remote environment to be explored is limited to the size of the user environment. This paper proposes an extension of telepresent walking to arbitrarily large remote or virtual spaces based on compressing wide-area motion into the available user space. Motion compression is a novel approach and does not make use of scaling or walking-in-place metaphors. Rather, motion compression introduces some deviation of curvature between user motion and teleoperator motion. An optimization approach is used to find the user path of minimum curvature deviation with respect to a given predicted teleoperator path that fits inside the boundaries of the user environment. Turning angles and travel distances are mapped with a 1:1 ratio to provide the desired impression of realistic self-locomotion in the teleoperator’s environment. The effects of the curvature deviation on inconsistent perception of locomotion are studied in two experiments.

## I Introduction

Visiting remote environments from your living room is a convenient alternative to being physically present on scene. Doing so requires a mobile teleoperator that moves under user control in the remote environment. It is equipped with a pair of cameras that transfers the perceived stereoimage stream to the user’s current location.

The user’s perception of locomotion is incomplete when remaining stationary with respect to the user environment while controlling the teleoperator’s locomotion. It is also well known that users quickly get lost in complex remote environments because of inaccurate path integration. This is due to missing proprioceptive cues, that is, the “feeling” of locomotion. Proprioception is essential for the spatial perception of a remote environment and, hence, a prerequisite for navigation and wayfinding (Darken, 1999).

Greater realism is achieved by allowing the user to walk about the user environment. The motion of the teleoperator in the remote environment is then controlled by the user’s physical motion, which is sensed by some kind of tracking system. The images of the remote environment as perceived by the



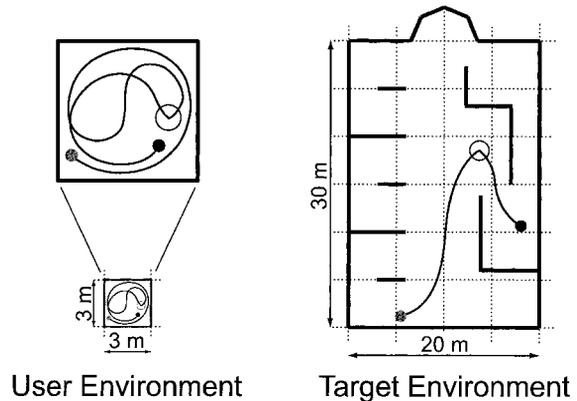
**Figure 1.** Telepresent walking. The user controls locomotion in the target environment (real or virtual) by walking in the user environment.

teleoperator are displayed to the user. As a result, the user has the impression of physically walking through the remote environment (Figure 1).

However, the available user space is usually much smaller than the considered remote environment. Space restrictions either result directly from the size of the user environment or are caused by the limited range of the tracking system. Both severely limit the size of the remote space that can be visited.

This paper introduces a new paradigm, called motion compression (MC), that allows strolling through an arbitrarily large remote environment by actually walking in a much smaller user space (Figure 2). This is achieved by transforming remote environment motion to a corresponding motion in the available user space (Figure 2). This transformation does not make use of scaling and is selected such that degradation of realism is minimized.

MC is not limited to the control of the locomotion of a mobile teleoperator. In fact, it can readily be applied to locomotion in large virtual environments. In the remainder of this paper, both remote real environments and virtual environments will be called *target environments*. The teleoperator and the virtual representation of the user in a virtual target environment (avatar) will both be denoted as *proxies*.



**Figure 2.** Example of motion compression. The user is walking along the convoluted path in the user environment (left). By visual perception, the user has the impression of walking along the intended path in the target environment (right).

## 2 Related Work

A great variety of approaches to teleoperated locomotion of mobile robots can be found in the literature. A good overview is given by Fong and Thorpe (2001). Many current systems allow the user to control robot locomotion by means of high-level commands such as target positions or desired paths, with the robot more or less autonomously executing these commands. It is not the goal of these systems to provide the user with a realistic sensation of self-motion in the target environment.

Direct interfaces for teleoperated locomotion as conventionally used employ hand controllers like joysticks to control teleoperator velocities while visual feedback is provided from cameras mounted on the teleoperator. Such setups create the impression of driving the mobile teleoperator.

Paulos and Canny (1998) use an Internet interface to control a mobile teleoperator equipped with a digital camera, microphone, and speakers. Their goal is to create “tele-embodiment”—that is, the user identifies with the teleoperator. However, in our opinion, the locomotion interface and the kinematics of the mobile teleoperator prohibit a realistic impression of self-motion.

Locomotion in large virtual environments is also ad-

dressed in a variety of research projects. In many cases, the focus is on designing realistic interfaces trying to create a “perfect” sensation of self-motion. Examples are 2D treadmills (Iwata, 1999; Darken, Cockayne, & Carmein, 1997) or the tracking of physical user locomotion (Welch et al. 2001; Nitzsche, Hanebeck, & Schmidt, 2001). Other researchers employ more-abstract metaphors to move through virtual environments. Tracking physical in-place walking and stepping motions and extracting the user’s intended locomotion is described by Slater, Usoh, and Steed (1994). LaViola, Feliz, Keefe, and Zeleznik (2001) implemented a Step WIM (world in miniature) to allow users to travel large distances by tapping on target locations in a reduced view of the virtual environment projected on the floor.

The work reported on by Razzaque, Kohn, and Whitton (2001) is similar to motion compression as it is based on the inconsistency between visual and proprioceptive perception of self-motion. However, this method relies on predefined target paths with a sufficient number of on-the-spot direction changes. By amplifying angular motion primarily during these changes of direction, turning angles are not mapped with a 1:1 ratio as in our approach.

The discussion of advantages and disadvantages of the various locomotion interfaces is continuing (Ruddle & Jones, 2001) and indicates that a universal solution is not available yet.

The effects of the various stimuli conveying information about self-motion on the human navigation and orientation capabilities have been studied by many researchers. The results of Bakker, Werkhoven, and Passenier (1999) show that kinesthetic feedback among vestibular feedback and visual flow provides the most reliable and accurate data for path integration. Nevertheless, vestibular perception is essential for accurate path integration, too (Becker, Nasios, Raab, & Jürgens, 2002; Glasauer, Amorim, Viaud-Delmon, & Berthoz, 2002). The work of Chance, Gaunet, Beall, and Loomis (1998) suggests locomotion interfaces that use real translational and rotational user motion as input for tasks involving spatial orientation.

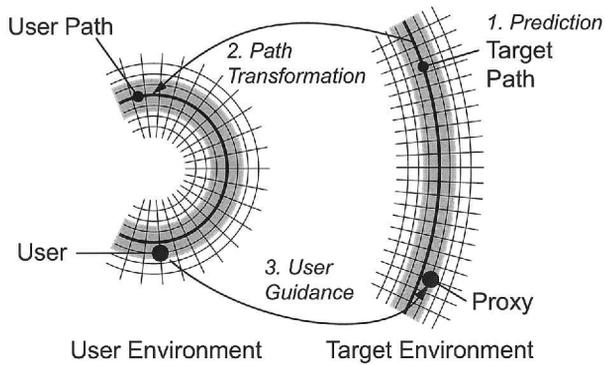
The human ability to detect the curvature deviation inherent to MC is related to the human vestibular and

podo-kinesthetic sensation. Kolev, Mergner, Kimming, and Becker (1996) found a detection threshold of  $1.2^\circ/\text{s}$  for rotational velocities detectable with the vestibular system. At an assumed walking speed of  $1.0 \text{ m/s}$ , this value corresponds to a calculated curvature detection threshold of  $1/50 \text{ m}^{-1}$ . However, this and most other investigations concerning the human sensation of self-motion have always tried to separate the various stimuli to study their individual contribution. In contrast, motion compression makes use of a certain amount of inconsistency between these stimuli, thus, their effects must be studied collectively.

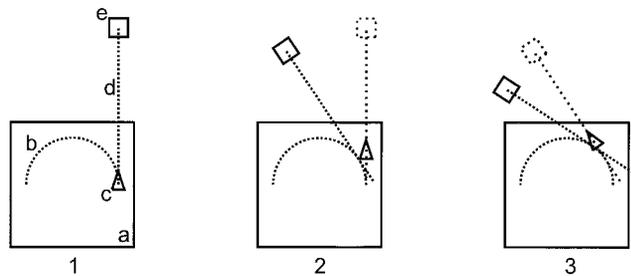
### 3 Basic Concept of Motion Compression

A user who is wearing a head-mounted display (HMD) and whose physical head motion is tracked in six degrees of freedom (DOF) can be immersed into a target environment by providing visual cues consistent with the detected head motion. The user receives the impression of walking in the target environment. MC extends this most realistic locomotion interface to target environments much larger than the available size of the user environment.

The basic concept of MC is to guide the user on a path in the user environment (user path), which is a transformed version, that is, an image of the path the proxy travels in the target environment (target path) (Figure 3). The user’s visual sensation is consistent with the motion of the proxy in the target environment and creates the impression of walking through the target environment. In contrast, proprioceptive (i.e., vestibular and kinesthetic) sensation reflects motion in the user environment. In other words, MC provides the correct visual stimuli corresponding to motion in the target environment but can only approximate proprioceptive stimuli. As an example, the user walking along a straight line in the target environment is guided along a circle in the user environment. Experiments demonstrate that the user tolerates a certain amount of inconsistency between visual and proprioceptive sensation of motion. (See Section 6.) This inconsistency results from the dif-



**Figure 3.** Motion compression. User locomotion is mapped from the user environment to the target environment in such a way that the user's path-tracking feedback control results in the user being guided on the user path. The condition of angle and distance preservation holds only in a narrow region around the nominal paths (shaded area).



**Figure 4.** User guidance as the basic principle of motion compression. a: user environment; b: user path; c: user; d: target path; e: object in target environment.

ferences in curvature between the target and the user path.

User guidance is a result of path-tracking control performed by the user who tries to stay on the intended target path. The user estimates his position and orientation in the target environment from the visual cues provided via the visual display. MC transforms user motion into the target environment in a way that results in visual cues guiding the user on the user path. Figure 4 illustrates the fundamental concept of user guidance, again using the example of straight line motion compressed into circular motion.

1. The user (c) is standing in the user environment (a) at the beginning of the user path (b). Wearing an HMD, the user sees the object (e) in the target environment and decides to walk towards the object on the target path (d).
2. The user has moved a short distance straight ahead in the user environment. However, the user's new position in the target environment reflects a circular motion resulting in deviation from the intended target path.
3. The user compensates for the perceived deviation (path-tracking control).

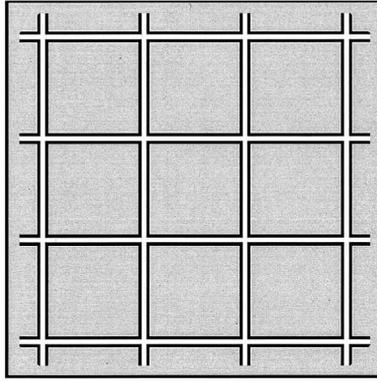
Although motion is 3D in both the user and the target environment, MC considers (and influences) only the 2D part of motion, that is, translation in the horizontal plane ( $x, y$ ) and rotation about the vertical axis ( $z$ ). Translation along the  $z$ -axis and rotations about the  $x$ - and  $y$ -axes remain unchanged. Planar transformations (other than scaling by a factor of 1) can precisely preserve angle and distance only for one-dimensional regions (i.e., lines). Thus, MC is based on the mapping between a narrow region in the target environment and a narrow region in the user environment. A target environment that resembles a narrow strip or a network of narrow strips (e.g., aisles) can be transformed statically into the user environment, as will be shown in Section 4.

In general, the target environment is not limited to narrow regions formed by physical constraints like walls and other obstacles. In this case, the transformation needs to be "dynamic." The angle and distance preserving condition cannot be met for the entire target environment by one single mapping. However, mappings for subspaces defined by narrow regions around the user's intended path can be approximately angle and distance preserving.

Therefore, MC comprises three main components. First is prediction of the intended locomotion of the user in the target environment (path prediction). The result is a predicted target path, a subspace of the target environment for which the transformation will be angle and distance preserving. Second is transformation of the predicted target path into the user environment (path transformation). The resulting user path is an image of



User Environment



Target Environment

**Figure 5.** Target environment with a finite number of discrete predefined paths. For targets that simple, static transformations from user to target environment can be found.

the target path with equal length, minimum curvature deviation, and fitting inside the user environment boundaries. Third is tracking of the user position and orientation with respect to the user environment and transformation back into the target environment. This transformation is defined by the user and the target path and is angle and distance preserving for positions close to the predicted path. As a result of this transformation, while having the visual impression of walking along the target path, the user is guided along the user path (user guidance) (Figure 3).

#### 4 Motion Compression for Target Environments with a Finite Number of Discrete Predefined Paths

Consider a simple target environment like the one in Figure 5, such as the aisles between bookshelves in a large library. An even simpler example would be just one straight aisle in the target environment that could be mapped to just one circle in the user environment.

The previous examples are special cases of MC because the target environment allows locomotion along only a finite number of well defined paths. Therefore, a specialized MC algorithm for such targets has to pro-

vide transformations that are valid for these predefined paths only. In terms of the general MC components introduced in Section 3, this means that path prediction is trivial because paths are known beforehand. In such cases, the predefined target paths can be transformed offline into the user environment resulting in a static mapping from the user to the target environment.

In the example in Figure 5, each aisle segment between two intersections is mapped to one of the circles in the user environment. The target path intersections all map to the upper left intersection of the two user path circles.

User guidance is equivalent to the general case of motion compression and will be the focus of the next section.

## 5 Motion Compression for General Target Environments

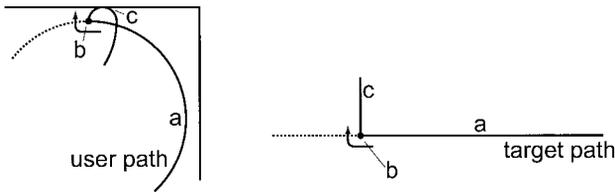
A general target environment has a continuum of paths that the user may move along. Offline transformation of all possible paths becomes impractical. Hence, one path or one set of paths is considered at a time, and path prediction as well as path transformation must be performed online.

### 5.1 Path Prediction

Reliable path prediction is vital for MC with minimum degradation of realism. This statement is illustrated by the example given in Figure 6, which represents a worst-case situation. The user leaves the predicted target path (unpredicted change of direction) at a point when he/she is very close to the boundary of the user environment and turning towards this boundary. To prevent the user from leaving the user environment, a user path with very strong curvature is required. Strongly curved user paths, however, are experienced as the target environment spinning around the user. This undesired behavior creates confusion and disorientation due to a large inconsistency between vision and proprioception.

Two approaches exist for path prediction. First, a

boundary of user environment



**Figure 6.** *Worst-case situation. The user leaves the predicted target path (a) turning towards the limit of the user environment at (b). This results in a strongly curved user path (c) and thus a large inconsistency between vision and proprioception.*

path prediction algorithm can be based on current and past position/orientation data of the proxy. Such an algorithm would then make use of some model of human locomotion to predict future motion. It is obvious that this locomotion-based approach can offer only short-term prediction as long as the locomotion model does not supply information about the user's global goal of locomotion. Second, long-term prediction is possible if the user's intention to head for a particular point or landmark in the target environment can be recognized. Intention recognition must be based on a model of the target environment from which information about potential goals of locomotion can be extracted.

The locomotion-based approach can be applied to structured as well as unstructured target environments because it does not rely on any information about the target environment. This is advantageous in the case of real target environments that are to be explored by means of a mobile teleoperator as is in military reconnaissance applications. However, MC based on short-term path prediction cannot explicitly avoid the worst-case situation mentioned previously. Sudden changes of the user's direction of locomotion can hardly be detected. Therefore, the path transformation component must try to keep clear from the boundaries of the user environment at all times to implicitly reduce the frequency of high curvature deviation. This results in stronger overall curvature but only reduces, and does not eliminate, the incidence of peak curvatures.

Target-based path prediction provides more informa-

tion about the user's future locomotion. However, it can be applied only to target environments of which a model is available. This is the case with virtual environments. Using a target environment model, the target-based approach can predict further into the future. Once the algorithm has recognized the user's intention to walk up to a particular object, a predicted path of a few meters may be the result. Such a predicted path may include, (unlike locomotion-based prediction) on-the-spot turns that can then be considered in the path transformation component. In addition, a target-based prediction usually gives information about the ultimate goal of the path. Assuming that the user stays on the path until reaching the goal, the goal is the only part of the path where unpredicted changes of direction may occur. Thus, only the end point of the path has to be transformed to a location in the user environment with considerable clearance from its boundaries to avoid strong curvature when the user starts out from there. This results in a considerable reduction of overall curvature.

Obviously, the assumption of the user always staying on the predicted path is questionable, but the target-based approach can also supply information about where along the path and in what direction unpredicted changes of direction may occur. An even more sophisticated algorithm determines the probability of changes of direction along the path. All this information can, if incorporated into the path transformation algorithm, reduce the incidence of peak curvature.

## 5.2 Path Transformation

The basic idea of path transformation is to bend the predicted target path such that it fits into the user environment. The result is the user path, a transformed version of the target path. As stated previously, the inconsistency between the visually perceived target path and the proprioceptively perceived user path results from the deviation of curvature. When walking along a path, human vestibular and kinesthetic senses alone are not sufficient for estimating the position and orientation on a horizontal plane (except for by path integration from a known pose). Vestibular and kinesthetic senses

can, however, detect translational and rotational speed. Provided that the lengths of target path and user path are kept equal, curvature deviation is the only difference a user can detect by means of proprioception. Curvature deviation obviously is produced by bending the path into the confined user environment and is thus inevitable. Therefore, the fundamental goal of path transformation is to minimize curvature deviation.

Path transformation could then be understood as a dynamic optimization problem with the continuous path variable  $s$  as the independent variable. The predicted path is given by the function

$$\kappa_T = \kappa_T(s) \quad 0 \leq s \leq s_E, \quad (1)$$

where  $\kappa_T$  is the target path curvature,  $s$  the distance along the path, and  $s_E$  the total length of the path. Starting position and orientation of the target path are not required for the transformation. The optimal user path to be found will be given by the function

$$\kappa = \kappa(s) \quad 0 \leq s \leq s_E \quad (2)$$

together with its starting position and orientation

$$x_0, y_0, \text{ and } \varphi_0. \quad (3)$$

An intuitive objective functional for minimizing overall curvature deviation is

$$J_1 = \int_0^{s_E} \frac{1}{2} (\kappa - \kappa_T)^2 ds. \quad (4)$$

The user path is subject to the following equality constraints describing the relationship of curvature, orientation, and position:

$$\frac{dx}{ds} = \dot{x} = \cos\varphi, \quad (5)$$

$$\frac{dy}{ds} = \dot{y} = \sin\varphi, \quad (6)$$

$$\frac{d\varphi}{ds} = \dot{\varphi} = \kappa, \quad (7)$$

and a set of inequality constraints characterizing the feasible space in the user environment,

$$\underline{g}(x,y) \geq 0. \quad (8)$$

The terminal position and orientation of the user path may either be fixed or free. If free, the terminal position and orientation become part of the optimization.

In general, the terminal position of the user path should be located as far away from the user environment boundaries as possible. This gives the user the freedom to continue locomotion in any direction from there. However, imposing a hard constraint for the terminal condition, in many cases, results in strongly curved paths. Therefore, a penalty term for the terminal condition is added to the objective functional according to

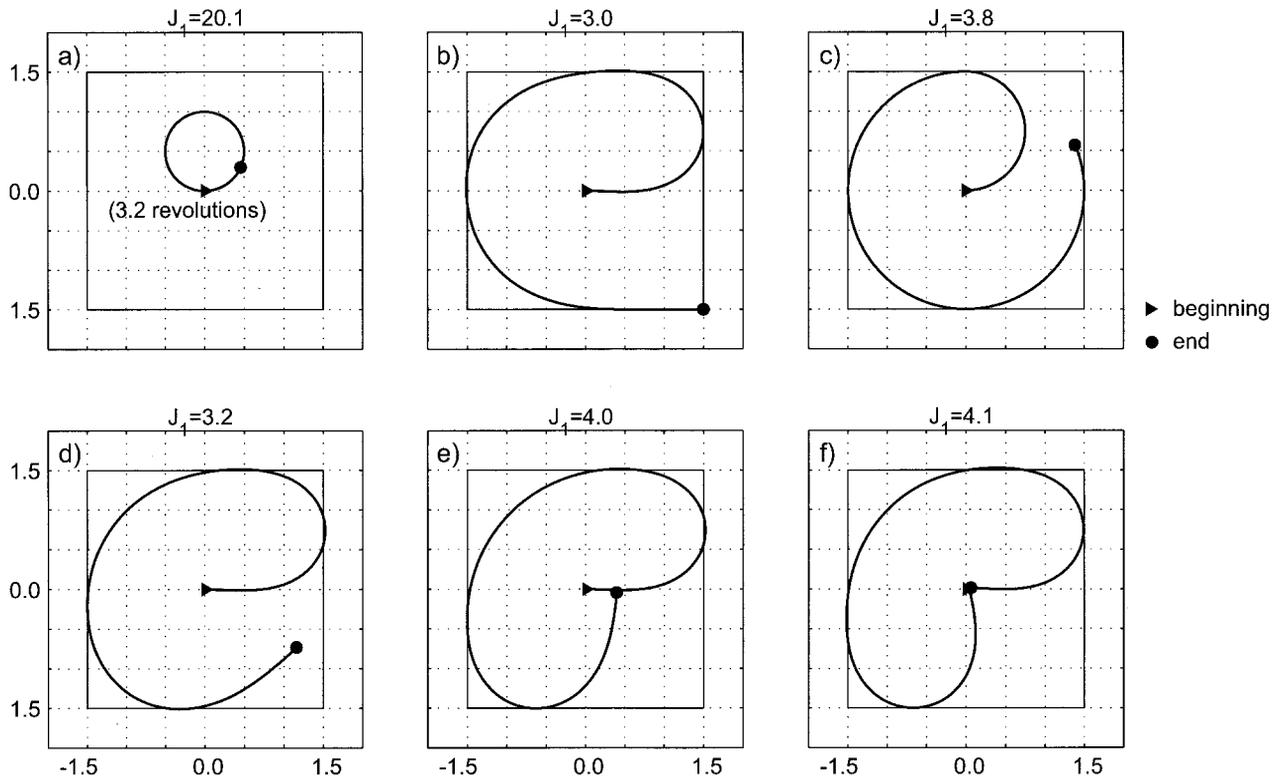
$$J_2 = J_1 + \phi[x(s_E), y(s_E), \varphi(s_E)]. \quad (9)$$

The general optimization problem (Equations 1–9) for finding a user path with minimum curvature deviation can be solved only numerically. DIRCOL (von Stryk, 1995, 1999) provides a powerful framework for this purpose. The sample solutions shown in Figure 7 have been computed on a Sun Ultra 450, Ultra Sparc II 400 MHz machine in about 700 ms. The target path is a straight line of 10 m. A circular path complying with the equality constraints (5), the inequality constraints (8), and the initial conditions (3) is used as an initial guess, Figure 7(a). Solution Figure 7(b) is based on the objective functional,  $J_1$ . The solutions in Figure 7(d), 7(e), and 7(f) were generated by the objective functional,  $J_2$ , and the penalty term

$$\phi[x(s_E), y(s_E), \varphi(s_E)] = \alpha_{s_E} \{x(s_E)^2 + y(s_E)^2\} \quad (10)$$

with  $\alpha = 0.02$ ,  $\alpha = 0.1$ , and  $\alpha = 1.0$ , respectively.

Figure 7(b) indicates that the endpoint of the transformed user path is very likely to be close to the user environment boundaries. In general, this may lead to peak curvatures at the beginning of the subsequent path. By introducing the penalty term  $\phi$  (see Equations 9 and 10), the resulting user path is a compromise between overall curvature and distance of the endpoint to the boundaries. See Figure 7(d), 7(e), and 7(f). The choice of one particular value for the



**Figure 7.** Sample solutions of path transformation. (a) initial guess. (b), (d), (e), and (f) numerical solutions based on various objective functionals. (c) solution found by the semicircle algorithm. (See Section 5.4.)

parameter  $\alpha$  depends primarily on the size of the user environment but is independent of the length of the target path.

### 5.3 User Guidance

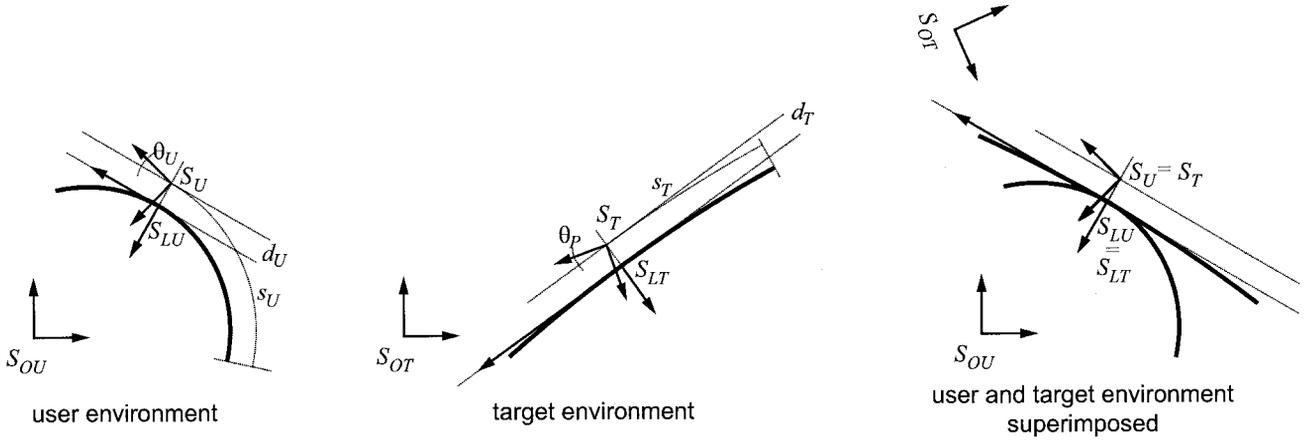
The user guidance component of MC maps the measured user position and orientation from the user environment to the position and orientation of the proxy in the target environment. The mapping is defined by the target and the user path. The basic idea of user guidance is to make use of the user's path tracking feedback control. Position and orientation in the target environment are calculated such that the resulting visual feedback guides the user on the transformed (user) path.

A person walking toward a particular goal in his/her environment acts as a human feedback controller. The

desired path to the goal is continuously compared to the actual path, and compensating locomotion is performed when a deviation between desired and actual path is perceived. The actual path is estimated from sensory input—this is, visual and proprioceptive cues. MC affects the estimation of the actual path by providing modified visual cues.

One obvious method is obtained by specifying the following properties for the mapping (see also Figure 8):

- Positions on the user path are mapped to positions on the target path, with the distance the user has moved along the user path ( $s_U$ ) equal to the distance along the target path ( $s_T$ ).
- Positions off the user path are mapped to positions off the target path, with the perpendicular distance from the user to the user path ( $d_U$ ) equal to the distance of the proxy to the target path ( $d_T$ ).



**Figure 8.** User guidance. The measured position and orientation of the user are mapped to the position and orientation of the proxy. Superimposing user and target environment illustrates the relationship between both environments.

- User orientation tangential to the user path is mapped to proxy orientation tangential to the target path, or, more generally, the angle between user orientation and the tangent of the user path ( $\theta_U$ ) equals the angle between proxy orientation and target path ( $\theta_T$ ).

These properties are taken into account by introducing two local Cartesian coordinate frames,  $S_{LU}$  and  $S_{LT}$  in addition to the global frames,  $S_{OU}$  and  $S_{OT}$ . The origin of  $S_{LU}$  is on the user path at position  $s_U$ , and its  $x$ -axis is tangential to the user path. The origin of  $S_{LT}$  is on the target path at position  $s_T = s_U$  with its  $x$ -axis tangential to the target path (Figure 8). As a consequence of the previously specified properties, the position and orientation of the user ( $U$ ) with respect to  $S_{LU}$  equal the position and orientation of the proxy ( $T$ ) with respect to  $S_{LT}$ . Introducing homogeneous transformations to express position and orientation of one coordinate frame with respect to another, this can be expressed by

$${}^{LU}T_U = {}^{LT}T_T. \quad (11)$$

Assuming known transformations  ${}^{OU}T_{LU}$  and  ${}^{OT}T_{LT}$ , the position and orientation of the proxy with respect to  $S_{OT}$  ( ${}^{OT}T_T$ ) are calculated from the measured position and orientation of the user with respect to  $S_{OU}$  ( ${}^{OU}T_U$ ) by

$${}^{OT}T_T = {}^{OT}T_{LT} {}^{LU}T_{OU} {}^{OU}T_U. \quad (12)$$

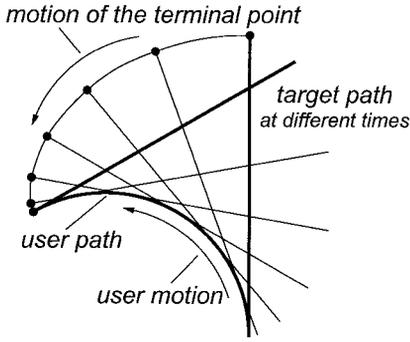
From Equation 12, the transformation from user to target environment or vice versa is obtained as

$${}^{OT}T_{OU} = {}^{OU}T_{OT}^{-1} = {}^{OT}T_{LT} {}^{LU}T_{OU}. \quad (13)$$

This transformation describes position and orientation of the target environment with respect to the user environment (Figure 8).

If user and target environment are superimposed according to this transformation, the target path will always be tangential to the user path in the point specified by  $s_U$ . As the user moves along the user path, the target environment will move relative to the user environment in a way equal to the motion resulting from the target path rolling off the user path. In the case of the target path being a straight line and the user path being a circle, each point of the target environment moves along an evolute with respect to the user environment (Figure 9).

The transformation  ${}^{OU}T_{LU}$  can be obtained from the measured position of the user in  $S_{OU}$  by dropping a perpendicular from the user position onto the user path. The foot of the perpendicular is the origin of  ${}^{OU}T_{LU}$  and thus determines the path variable,  $s_U$ . Note that the perpendicular from the user position onto the target path may not always be unique.



**Figure 9.** As the user moves along the user path, the target path rolls off the user path with points in the target environment describing evolvents with respect to the user environment.

## 5.4 Implemented Solutions

In the preceding subsections, we presented the methodology that MC is based upon. For some of the related problems, solutions are not yet available or are difficult to implement. Therefore, we will introduce some simplifying assumptions and detail practical solutions.

**5.4.1 Path Prediction.** Both a locomotion-based and a target-based algorithm for path prediction have been implemented and tested successfully.

The locomotion-based algorithm rests upon the following simplified model of human locomotion: future walking direction equals current gaze direction. In this case, the predicted path is always a straight line starting at the proxy's position and running in the direction of its gaze. The endpoint of the predicted path is unspecified. Thus, a corresponding path transformation algorithm cannot optimize the endpoint of the user path and must anticipate unpredicted changes of the direction of locomotion.

The implemented target-based algorithm is equally simple. As mentioned, it is applicable to only target environments of which a model is available. From the model, objects of interest are identified offline and manually as potential goals of locomotion. One of these potential goals is then selected online as the endpoint of the predicted path. The selection is based again on gaze

direction. The longer the user remains looking at a potential goal, the more likely the object is the user's goal. This likelihood is expressed by a coefficient,  $w_i \in [-1; 1]$ , assigned to every potential goal.  $w_i$  is increased when the corresponding object is within the field of view and decreased otherwise. The object with the highest  $w_i$  is selected, and the predicted path is the straight line from the user to the selected goal. With this prediction algorithm, optimization of the user path endpoint becomes feasible.

Path prediction is performed concurrently with user tracking. The predicted target path is updated with every incoming position and orientation measurement.

**5.4.2 Path Transformation.** The dynamic optimization problem defined in Section 5.2 can be reduced to a form for which an analytical solution exists. This solution will then be used to find suboptimal solutions for the original problem.

We temporarily neglect the inequality constraints (Equation 8). Furthermore, we assume the target path to be a straight line: that is  $\kappa_T(s) = 0$ . Then the optimization problem reduces to finding the minimum of

$$J_3 = \int_0^{s_E} \frac{1}{2} \dot{\varphi}^2 ds. \quad (14)$$

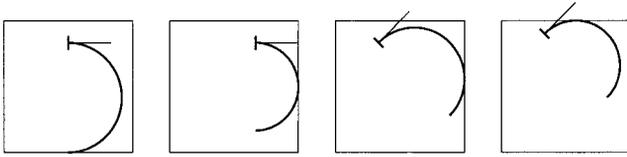
By calculus of variation, the necessary condition for a minimum of Equation 14 is given by

$$-\frac{d}{ds}\dot{\varphi} = -\frac{d^2\varphi}{ds^2} = 0. \quad (15)$$

Paths satisfying Equation 15 minimize the objective functional and are either circular arcs or straight lines. For that reason and because circles are convenient to deal with, path transformation is reduced to finding an appropriate circular arc.

A first heuristical solution, which incrementally transforms a target path to a user path, is based on the following assumptions:

- The predicted path is a straight line starting at the proxy's position.



**Figure 10.** Largest semicircles fitting into a square user environment for different given starting positions and directions.

- The predicted path has no fixed endpoint.
- The user environment is a convex polygon.

Then the user path is a circular arc satisfying the following conditions:

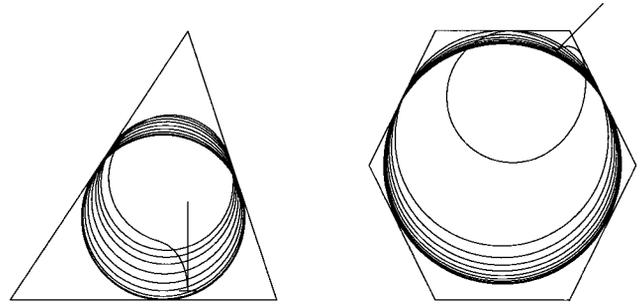
- The starting point is the user position.
- The starting direction coincides with the direction of the target path mapped into the user environment.
- The arc is the largest semicircle fitting into the user environment limits.

Figure 10 presents some examples with varied user positions and target path directions in a square user environment.

User path arcs are determined successively according to these conditions for short sections of the target path. Thus, only a short part of each user path arc is actually used, and the total user path is a chain of short circular arcs. Thanks to the second condition, transitions between all circular arcs forming the user path are smooth with continuous orientation but discontinuities in curvature.

Figure 7 shows a user path created with the semicircle algorithm. The target path is a straight line with a length of 10 m. The transformation was achieved by successively transforming fractions of 1 cm. For more examples of the semicircle algorithm, see Figure 11.

Although not proven yet, simulations of this simple path transformation algorithm lead to the conclusion that the resulting user paths always converge to the maximum inscribed circle of the user environment for straight target paths of sufficient length and convex user environments (Figure 11). This property is rather favorable if the predicted path in fact is a straight line and no



**Figure 11.** User paths planned with the semicircle algorithm in different convex user environments. For sufficiently long straight target paths, the resulting user path converges to the maximum inscribed circle of the user environment polygon.

change of direction must be expected. In this case, the maximum inscribed circle of the user environment is the optimal user path with minimum curvature.

However, if the user does not stay on a predicted straight target path, the worst-case situation introduced in Section 5.1 is quite likely to occur. This is because of the fact that user paths planned with the semicircle algorithm tend to tangent the limits of the user environment.

Two simple strategies have been implemented to remedy this disadvantage. The common goal is to position the endpoint of a user path closer to the center of the user environment to provide the freedom of motion to set off in a new direction from there. Both strategies rely on the path prediction algorithm providing the length of the user path in addition to the path direction.

One strategy is to temporarily contract the user environment and simply use the semicircle algorithm as described. This results in spiral paths tending towards the center of the user environment. The alternative strategy is to find a circular arc that brings the user to the center of the user environment. This arc must satisfy the following conditions:

- Starting point is the user position.
- Starting direction coincides with the direction of the target path mapped into the user environment.
- Endpoint is the center of the user environment.
- The length of the arc equals the lengths of the target path.

It is obvious that these conditions overdetermine the circular arc. Thus, the semicircle algorithm is used for path transformation until a circular arc is found that satisfies all of the aforementioned conditions.

In the experimental setup presented in Section 6, the semicircle algorithm is executed concurrently with user tracking. A new circular arc is determined with every new measurement from the tracking system and every update of the predicted target path. Thus, the user path is planned incrementally and not in advance. Nevertheless, the semicircle algorithm features predictive behavior that is accounted for by considering the boundaries of the user environment before they are actually hit.

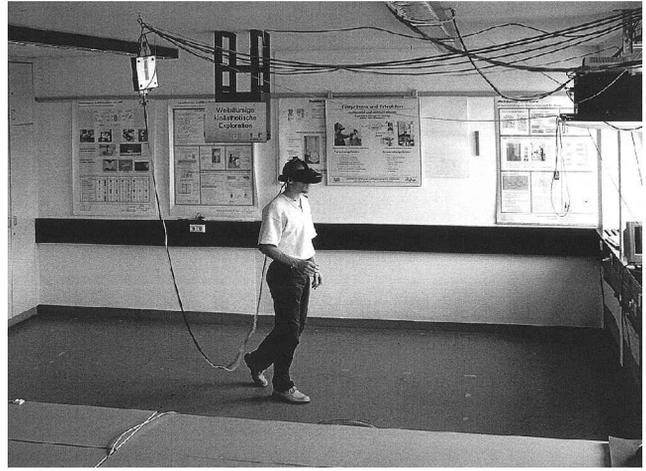
**5.4.3 User guidance.** The mapping from measured user coordinates to the proxy's position and orientation follows exactly the method described in Section 5.3.

## 6 Experiments

Two experiments with differing objectives have been performed with the current implementation of the MC approach. In the first experiment, the goal was to find the minimum deviation of curvature a human user can detect by the inconsistency between proprioceptive and visual stimuli. The objective of the second experiment was to show that higher deviations of curvature are tolerable if the target environment provides sufficient distraction to the user.

### 6.1 Experimental Setup

Experiments described in the following sections have been conducted in a user environment with a  $4 \times 4$  m floor space. MC makes use of only a  $3 \times 3$  m area to ensure some safety margin to the physical walls. An Ascension Flock of Birds magnetic tracking system with a long-range transmitter is used for user tracking. It supplies position and orientation of the user's head with a sampling rate of 100 Hz. No measures for compensation of metal distortion have been implemented.



**Figure 12.** User environment used in the experiments. Available floor space is approximately  $4 \times 4$  m. The user wears an HMD, and head motion is tracked by an extended range magnetic tracker.

The user wears a V8 head-mounted display (HMD) by Virtual Research Systems, Inc., which displays stereo images of the target environment at standard VGA resolution. The textured geometric models of the target environments are implemented with Maverik. The visual rendering of the target environments is done on a Pentium III/800 equipped with two Voodoo2 3D accelerators. Figure 12 shows the user environment.

### 6.2 Experiment I: Maximum Allowed Curvature

The inequality of target path and user path curvature results in an inconsistency between the visual and the proprioceptive—that is, kinesthetic and vestibular—sensation. The objective of this experiment was to find the perceptual threshold,  $\Delta\kappa_b$  of curvature deviation,  $\Delta\kappa_s$  below which this inconsistency is no longer detectable.

**6.2.1 Experiment Design.** The goal of the experiment is to find the probability,  $p$ , of a person correctly deciding whether the user path was curved left or right as a function of curvature. For curvatures below the perceptual threshold, a value of  $p = 0.5$  is expected,

which means that test persons are equally likely to give correct and incorrect answers. Note that this probability is different from the probability of a person correctly detecting the direction of curvature.

At a given user path curvature  $\kappa_i$ ,  $m_i$  denotes the number of correct answers, and  $n_i = n$  denotes the number of trials. Then,  $m_i$  is distributed with a binomial distribution, and the probability to achieve a trial result,  $m_i$ , is given as

$$P(m_i) = \binom{n}{m_i} p_i^{m_i} (1 - p_i)^{n - m_i}. \quad (16)$$

The maximum likelihood estimator for  $p_i$  given a trial result  $m_i$  is

$$\hat{p}_i = \frac{m_i}{n}. \quad (17)$$

$\hat{p}_i$  is approximately characterized by a Gaussian distribution with variance

$$\sigma^2 = \frac{p(1 - p)}{n}. \quad (18)$$

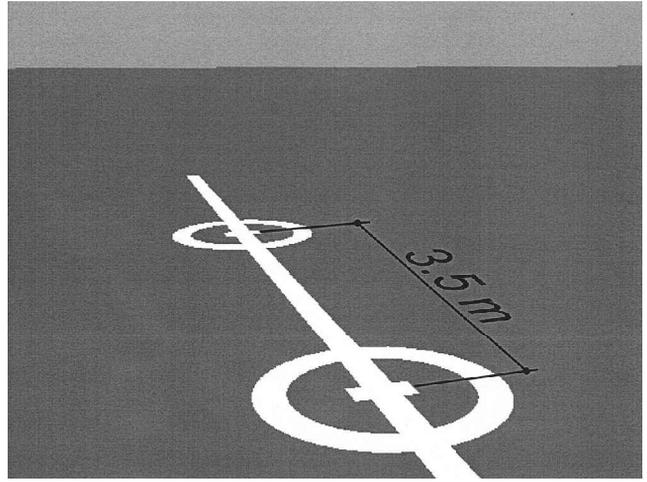
With a given upper limit  $\sigma_{max}^2$  for the variance,  $\sigma^2$ , a value ( $n_{min}$ ) can roughly be estimated. Demanding  $\sigma < 0.1$  and assuming  $p = .5$ , one obtains

$$n_{min} \approx 25.$$

In the experiment,  $n = 24$  was chosen based on this estimation and because  $n$  is required to be even.

Because  $\Delta\kappa_l$  was expected to be close to  $0.1 \text{ m}^{-1}$ , tested curvatures ranged from  $0.0$  to  $0.2 \text{ m}^{-1}$ . Twenty discrete curvature values were chosen from this range. Each curvature (except  $\Delta\kappa = 0.0$ ) was tested  $n/2$  times as a left turn and  $n/2$  times as a right turn, the whole set of trial curvatures comprising  $N = n/2 + n \cdot 19 = 468$  members. Results with equal absolute value of curvature but different sign (left and right curved) are treated as one curvature, thus supplying  $n$  trials per curvature.

**6.2.2 Test Procedure.** From preliminary studies, the threshold to be found was expected in the neighborhood of  $\Delta\kappa = 0.1 \text{ m}^{-1}$ . Full circles with corre-



**Figure 13.** Target environment for determining the perceptual threshold of curvature deviation. Infinite plane with markings.

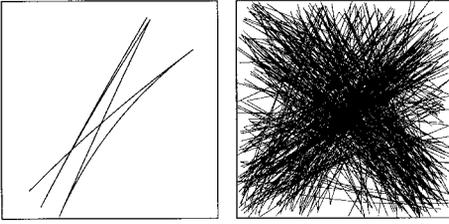
sponding radii would not fit into the available floorspace of the laboratory used and would, in addition, exceed the limits of the tracking system. Therefore, the user paths are only circular arcs of a certain curvature. The target path is a straight line in all the experiments.

The target environment is kept as simple as possible to avoid the influence of distraction. It comprises only an untextured green plane of infinite size and white markings to indicate to test participants where to walk. Figure 13 shows a screenshot of the target environment. The distance between the two endpoints of the target path is 3.5 m. Test persons walk back and forth between the two endpoints  $N$  times during the experiments.

Corresponding user paths are created by randomly choosing curvatures from a predefined set that includes a preset number of instances of each curvature to be tested. By using curvatures at random order, the influence of accommodation is minimized. Figure 14 shows the user paths for the first five trials and the total set of  $N = 468$  user paths.

Before the actual experiment, test subjects walked along the target path with zero user path curvature, maximum user path curvature, and with an intermediate value to accustom them with the procedure and the user path curvatures to be expected in the experiment.

Subjects were repeatedly directed to walk from one



**Figure 14.** The first five user paths from the trial set (left) and the total set used in the experiment (right).

end of the target path to the other. At the end of each walk, the subject was asked to rank realism on a scale from 1 to 5 and to assess whether the user path was curved left or right. The answer to the realism question was not evaluated.

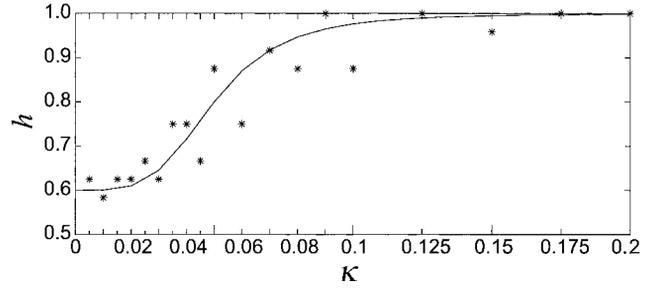
To eliminate position and orientation cues gained by auditory stimuli from the user environment, subjects wore headphones emitting white noise. Communication from the experimenter to the subject was via the headphones and prerecorded messages for the same reason.

The total experiment took approximately 4 hours to complete. Subjects were allowed to take breaks of 2 min after each set of 50 trials and a 30 min intermission after about 50% of the trials.

**6.2.3 Results.** Due to the immense effort for testing  $N = 468$  individual user paths, only one subject was used for the final experiment. Preliminary studies trying to distribute the whole set onto several subjects did not yield any useful results.

Figure 15 shows the relative frequency,  $h$ , of correctly detected direction of curvature as a function of  $\kappa$ . As stated,  $h$  is the maximum likelihood estimate for  $p$ . Each individual,  $\hat{p}_b$  is based on  $n/2 = 12$  left-curved and  $n/2 = 12$  right-curved user paths with the corresponding absolute curvature.

Note that  $\hat{p}$  does not reach the expected limit of 0.5 even for very small curvatures. This may be due to several reasons. First, the estimated standard deviation for  $\hat{p}$  is about 0.1. Second, there is still a small chance that the user correctly detects the direction of curvature for  $\kappa \geq 0.005 \text{ m}^{-1}$ , and  $p = p(\kappa)$  drops to 0.5 at some



**Figure 15.** Experimental result. Relative frequency,  $h$ , of correctly detected direction of curvature as a function of curvature  $\kappa$ .

value  $0.0 < \kappa < 0.005 \text{ m}^{-1}$ . From the data shown in Figure 15, it is obvious that, for curvatures of  $\kappa < 0.1$ , the user is prone to fail in assessing the direction of curvature. We therefore “define”  $\kappa = 0.1 \text{ m}^{-1}$  as the perceptual threshold of curvature deviation.

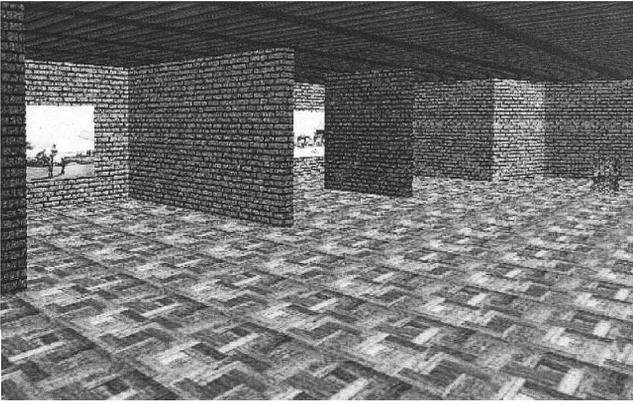
Of course, this value can only be a rough estimate and is indented as a rule of thumb because it was acquired from one subject only and for target path curvature  $\kappa_T = 0.0$ .

### 6.3 Experiment II: Visiting a Virtual Museum

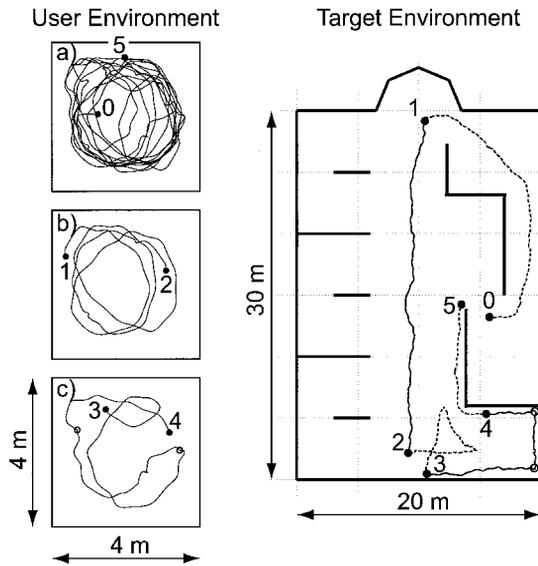
The MC algorithm is applied to visiting a large hall in a virtual museum. The objective of this experiment is to demonstrate the functionality of MC in a typical application scenario. The psychological interpretation of the results is restricted to subjective ratings of test participants and some generalized observations made by the authors.

Figure 2 has the layout of the virtual museum with a floor space of  $30 \times 20 \text{ m}$ . Figure 16 shows a screenshot. Although the available user environment is  $3 \times 3 \text{ m}$  only and thus user path curvature is usually greater than  $0.3 \text{ m}^{-1}$ , most test subjects found strolling through the virtual museum to be very comfortable and intuitive.

As long as the user decides to view one piece of art and walks there straight ahead, no peaks of curvature deviation occur and subjects quickly become accustomed to the inconsistency of visual and proprioceptive



**Figure 16.** Screen shot of the virtual museum used in the experiment.



**Figure 17.** Corresponding traces of a test subject in the user and the target environment. Note that the user environment plots are enlarged by a factor of 3 with respect to the target environment. (a) Entire path from 0 to 5. (b) More or less straight motion from 1 to 2. (c) Path from 3 to 4 with two  $90^\circ$  turns indicated by the empty circles.

perception. Subjects are observed to reach normal walking speed of  $0.5\text{--}0.8\text{ m/s}$  after a few cautious initial steps. However, peak curvature and changes of direction of curvature are likely to occur when the user decides to

choose a new goal of locomotion at some unfavorable position close to the boundaries of the user environment. As a result, the user experiences a spinning motion of the target environment. This spin never exceeds an angle of  $90^\circ$  due to the nature of the semicircle algorithm. In addition, user path curvature was limited to  $10\text{ m}^{-1}$ . Therefore, users do not lose the sense of where they are in the target environment during the spin; they only have to reorient themselves.

Limiting the user path curvature does increase the fidelity of MC because it avoids the very quick spins encountered when the user is at the boundaries. However, the transformed user path is no longer guaranteed to be within the boundaries. Therefore, the boundaries of the user environment must have some clearance to the physical boundaries of the available user space. In addition, the user environment boundaries are blended into the rendered image of the target environment as a red ribbon when the user is about to leave the boundaries.

## 7 Conclusion

A novel general and systematic method for compressing large-scale voluntary locomotion into much smaller available space has been presented and termed *motion compression* (MC). MC does not rely on scaling, rather, it is based on the modification of path curvature. Thus, turning angles and distances are transformed with a ratio of 1:1. However, MC can be combined with scaling to magnify or reduce target environments that otherwise would not be accessible by human walking. Consider, for instance, walking through a biological cell or through a solar system.

The advantage over conventional interfaces for locomotion in virtual or remote real environments (target environments) is that MC provides the sensation of motion by approximating proprioceptive cues. The visual sensation is consistent with motion in the target environment; the proprioceptive sensation, however, is consistent with the physical motion of the user. Due to the modification of path curvature, this proprioceptive sensation is but an approximation of the proprioception

that is consistent with motion in the target environment.

Small inconsistencies are tolerated by the user. An experiment revealed that path curvatures of less than  $0.1 \text{ m}^{-1}$ , corresponding to a radius of 10 m, can be made unnoticeable to the user by providing visual cues convincing the user that he or she were walking straight ahead. Even stronger curvature deviation is tolerable if the target environment provides sufficient distraction. In the current setup, the user can visit a virtual museum of  $30 \times 20 \text{ m}$  floor space from a tracking area of  $3 \times 3 \text{ m}$ . MC exploits the effect of one sensation (vision) masking another (proprioception).

MC is introduced for general unstructured target environments as well as target environments satisfying certain prerequisites. In any case, MC comprises three main components: estimation/recognition of intention for predicting future locomotion of the user, transformation of the predicted path into the limited user environment, and tracking the user's physical motion and mapping it to the target environment. Mapping user motion to the target environment is performed in such a way that the resulting visual cues affect the user's path tracking control, thus guiding the user on the transformed path.

MC can be applied to telepresent locomotion in virtual as well as real target environments. In virtual environments, the user controls the motion of a virtual representation of himself, thus getting the impression of being immersed in the virtual environment. In real target environments, a mobile teleoperator is controlled according to the user's physical motion.

Virtual environment applications include large virtual meeting places, virtual museums, and visiting virtual prototypes of architecture. A real target environment could also be a museum, offering mobile teleoperators that people from other parts of the world can log into.

In our laboratory, an HMD and magnetic tracking are employed as the human-system interface. However, the method can readily be applied to other visual displays and tracking systems, such as a CAVE (Cruz-Neira, Sandin, & DeFanti, 1993).

In the future, MC will be extended to 3D environments including stairs, ladders, and slopes. Another ex-

ension is to share the user environment with its infrastructure (floorspace, tracking system and so on) among several users. Path transformation in this multiuser scenario not only has to consider the boundaries and static obstacles of the room but must also provide reliable user-to-user collision avoidance measures.

With respect to the path prediction and transformation components, stochastic processing promises a substantial increase of performance. The current implementation relies on just one predicted path. If that path proves wrong, degradation of realism is very likely to occur due to strong curvature or even curvature peaks.

## Acknowledgments

This work was supported in part by the German Research Foundation (DFG) within the Collaborative Research Centre SFB 453 on "High-Fidelity Telepresence and Teleaction."

Special thanks must be given to Prof. Jianbo Su, Shanghai Jiao Tong University, for many inspiring discussions during his three-month visit to our laboratory.

The authors would also like to express their gratitude to the anonymous reviewers for their invaluable comments and suggestions.

## References

- Bakker, N. H., Werkhoven, P. J., & Passenier, P. O. (1999). The effect of proprioceptive and visual feedback on geographical orientation in virtual environments. *Presence: Teleoperators and Virtual Environments*, 8, 36–53.
- Becker, W., Nasios, G., Raab, S., & Jürgens, R. (2002). Fusion of vestibular and podokinesthetic information during self-turning towards instructed targets. *Exp. Brain Res.* 144(4), 458–474.
- Chance, S. S., Gaunet, F., Beall, A. C., & Loomis, J. M. (1998). Locomotion mode affects the updating of objects encountered during travel: The contribution of vestibular and proprioceptive inputs to path integration. *Presence: Teleoperators and Virtual Environments*, 7, 168–178.
- Cruz-Neira, C., Sandin, D., & DeFanti, T. A. (1993). Surround-screen projection-based virtual reality: The design

- and implementation of the CAVE. *Proceedings of the ACM SIGGRAPH*, 135–142.
- Darken, R. P. (1999). Spatial orientation and wayfinding in large-scale virtual spaces II. *Presence: Teleoperators and Virtual Environments*, 8, iii–vi.
- Darken, R. P., Cockayne, W. R., & Carmein, D. (1997). The omni-directional treadmill: A locomotion device for virtual worlds. *Proceedings of the ACM UIST*, 213–222.
- Fong, T., & Thorpe, C. (2001). Vehicle teleoperation interfaces. *Autonomous Robots*, 11, 9–18.
- Glasauer, S., Amorim, M.-A., Viaud-Delmon, I., & Berthoz, A. (2002). Differential effects of labyrinthine dysfunction on distance and direction during blindfolded walking of a triangular path. *Exp. Brain Res.*, 145, 489–497.
- Iwata, H. (1999). Walking about virtual environments on an infinite floor. *Proceedings of the Virtual Reality Conference*, 286–293.
- Kolev, O., Mergner, T., Kimming, H., & Becker, W. (1996). Detection thresholds for object motion and self-motion during vestibular and viso-motor stimulation. *Brain Res. Bull.*, 40, 451–457.
- LaViola, J. J., Jr., Feliz, D. A., Keefe, D. F., & Zeleznik, R. C. (2001). Hands-free multi-scale navigation in virtual environments. *Symposium on Interactive 3D Graphics*, 9–15.
- Nitzsche, N., Hanebeck, U. D., & Schmidt, G. (2001). Mobile haptic interaction with extended real or virtual environments. *Proceedings of the IEEE International Workshop on Robot-Human Interactive Communication*, 313–318.
- Paulos, E., & Canny, J. (1998). Designing personal tele-embodiment. *Proceedings of the IEEE International Conference on Robotics & Automation*, 3173–3178.
- Razzaque, S., Kohn, Z., & Whitton, M. C. (2001). Redirected walking. *Proceedings of EUROGRAPHICS*.
- Ruddle, R. A., & Jones, D. M. (2001). Movement in cluttered virtual environments. *Presence: Teleoperators and Virtual Environments*, 10, 511–524.
- Slater, M., Usoh, M., & Steed, A. (1994). Steps and ladders in virtual reality. *Proceedings of the ACM Conference on Virtual Reality Software and Technology*, 45–54.
- von Stryk, O. (1995). *Numerische Lösung optimaler Steuerungsprobleme: Diskretisierung, Parameteroptimierung und Berechnung der adjungierten Variablen*. Düsseldorf: VDI-Verlag.
- . (1999). *User's guide for DIRCOL: A direct collocation method for the numerical solution of optimal control problems*. Lehrstuhl für Höhere Mathematik und Numerische Mathematik, Technische Universität, München, 2.1.
- Welch, G., Bishop, G., Vicci, L., Brumback, S., Keller, K., & Colucci, D., (2001). High-performance wide-area optical tracking—The HiBall tracking system. *Presence: Teleoperators and Virtual Environments*, 10, 1–21.