Feedback Controlled Motion Compression for Extended Range Telepresence

Patrick Rößler and Uwe D. Hanebeck
Institute of Computer Design and Fault Tolerance
Universität Karlsruhe (TH)
Karlsruhe, Germany
Email: roessler@ira.uka.de, uwe.hanebeck@ieee.org

Norbert Nitzsche
Institute of Automatic Control Engineering
Technische Universität München
München, Germany
Email: norbert.nitzsche@ei.tum.de

***Abstract*— Telepresence aims at giving a human user the impression of being present in a remote environment. This is achieved by having a robot, that is actually present in the remote environment, gather visual data. This data is presented to the user, who is wearing a head mounted display. In order to extend telepresence to an intuitive user interface for robot teleoperation, the user's motion is tracked and replicated by the robot. The user can now interact more naturally with the remote environment simply by walking around. Without further processing of the motion data, the size of the remote environment is limited to the size of the user environment. Using the motion compression algorithm allows users to be telepresent in large target environments while the size of the user environment is limited. However, as a consequence of the influence of standard motion compression on the user's natural navigation, he tends to leave the desired path. This can even lead to users leaving the user environment. We address this problem by introducing controlled motion compression which adds a feedback controller to motion compression. This controller modifies the user's perception of the target environment in such a way, that he is controlled on the path.**

I. INTRODUCTION

In future, robots will be send to places where humans do not want to go themselves, be it for safety reasons or convenience. Those scenarios include exploration of foreign planets in space missions, nuclear cleanup after a reactor accident, or exception handling for household robots. In all these tasks the robot needs supervisory control from a human operator as the robot's understanding of its environment is by far not as elaborate as the human's.

Teleoperating robots with joysticks or PDA-based remote controls [1] is not very intuitive. Telepresence provides a much more intuitive way to control a robot.

The locomotion of a human operator (*user*) is tracked and transferred to a robot (*teleoperator*) that replicates the user's motion. In addition, the robot constantly collects visual and acoustic sensory data from its environment (*target environment*). This data is transferred back and presented to the user. As the user is actually walking around, he has the impression of walking about the target environment and thus gathers proprioceptive feedback that is consistent with the visual impression of the target environment. As stated in [2], this proprioceptive feedback, i. e., vestibular and kinestethic feedback, is important for the user's spatial perception, which is a prerequisite for human way finding and navigation.

For this reason, large-scale telepresence is also an appropriate user interface for virtual reality scenarios, like simulation, training, and gaming. In virtual reality, however, the user is not represented by a teleoperator but by an *avatar* in the target environment. In the following we use the term *proxy* for both teleoperators and avatars.

In all scenarios given above, the target environment may be very large, but the size of the user environment is typically limited. Such space restrictions either result directly from the size of the available user space or from the range limit of the tracking mechanism being used. In [2], the motion compression algorithm is presented, which allows walking about arbitrarily large target environments from relatively small user environments without scaling [3] or walking-in-place like metaphors [4].

The motion compression algorithm consists of three major modules:

1) Path prediction: In path prediction, the user's desired path in the target environment is estimated, resulting in the *target path*.
2) Path transformation: In path transformation, the target path is mapped onto a user path of equal length, which features the same turning angles if the user changes direction. This path is called the *user path*. There is, however, a difference in path curvature allowing the path to be fit into the user environment.
3) User guidance: In user guidance, finally, the user is guided on the user path by exploiting the user's navigational capabilities.

In this paper we propose a modification to the user guidance module.

The remainder of this paper is structured as follows. In section II we explain the reason for a noticeable deviation from the user path as well as from the target path when using standard motion compression. Section III discusses different solutions for taking influence on the user in order to compensate for this deviation. We show that only one of these possible solutions is adequate for solving the problem. Section IV presents controlled motion compression, which controls the user according to this solution. The experiments conducted to verify controlled motion compression are presented in section V.

II. PROBLEM FORMULATION

It can be observed that when using the standard motion compression (SMC) algorithm, a user tends to leave the desired path. The user shows a behavior that resembles a human walking on an inclined plane. This originates from a combination of characteristics of human navigation and SMC's user guidance. In short, SMC introduces deviations in the proxy's orientation, which cannot be compensated by the user.

A. User Model

A human walking in a goal oriented way constantly updates his orientation toward the goal based mainly on visual cues [5]. This update is delayed and limited by the user's dynamics. To model the behavior of the user we introduce the user's state as

$$\underline{x} = \begin{bmatrix} x \\ y \\ \phi \end{bmatrix}, \qquad (1)$$

where x and y describe the user's position and ϕ is his orientation. A simple model of a human walking toward a goal at position $\underline{x}_G = \begin{bmatrix} x_G & y_G & \phi_G \end{bmatrix}^{\mathrm{T}}$ is given as

$$\underline{\dot{x}} = \begin{bmatrix} V\overline{d}\cos(\phi) \\ V\overline{d}\sin(\phi) \\ 0 \end{bmatrix} + k \begin{bmatrix} 0 \\ 0 \\ \theta \end{bmatrix}, \qquad (2)$$

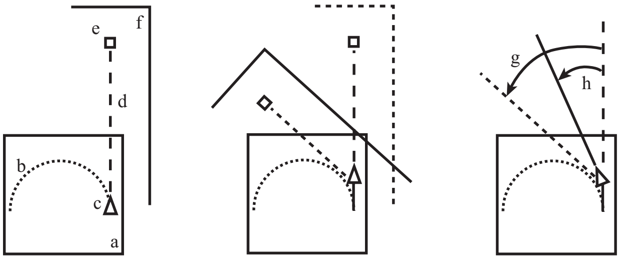
where $\theta = \mathrm{atan2}(x_G - x, y_G - y) - \phi$ is the perceived angle to the goal and $\overline{d} = \min(\sqrt{(x_G - x)^2 + (y_G - y)^2}, 1)$ is the bounded distance to the goal. The user's walking speed V is assumed to be constant and thus $V \cdot \overline{d}$ is constant and equal to V as long as the user is far away from the goal. As he approaches the goal he decelerates and finally comes to stop at the goal position. The constant k models the joint effects of dynamics and delay. This value, however, is user dependent.

B. User Guidance in Standard Motion Compression

In order to clarify the influence of motion compression on the user, we will explain user guidance in SMC by means of an example. A more detailed explanation can be found in [6]. In our scenario the target path is predicted as a straight line as depicted in Fig. 2(a). In this special case, the path transformation module transforms the target path into a circular path in the user environment. The user is then guided on this user path.

As the user is wearing a head mounted display and is presented the visual perception of the proxy, i. e., he sees through the proxy's eyes, user and proxy are coincident. As the user moves a short distance straight ahead in the direction where he saw the target, he leaves the user path, resulting in the proxy leaving the target path. Since the proxy's new position and orientation in the target environment is transformed in such a way, that it represents a circular motion, the user now sees the goal to his left as shown in Fig. 2(b). The user tries to compensate for the deviation by turning toward the goal, but according to the user model he might not fully succeed. Figure 2(c) shows the resulting deviation in orientation.

As SMC does not respond to this deviation, it results in user and proxy having an orientation pointing away from the desired path in both environments. As the orientation is integrated into the position, this deviation accumulates with time. As a result, the user leaves the desired path and eventually also the user environment. In the following, we propose a modification of the motion compression algorithm, called controlled motion compression, that controls user and proxy on the user path and the target path, respectively.



(a) (b) (c)

a... user environment
b... user path
c...user/proxy
d... target path
e... goal object
f... target environment
g... angle towards goal
h... real turning angle

Fig. 2. User guidance in standard motion compression.

III. POSSIBLE CONTROL APPROACHES

The goal of this section is to find an adequate modification of MC's user guidance module in such a way, that it compensates for the deviation described above. The only possibility to influence the user's behavior is by modifying his visual perception of the target environment. In order to make the user turn further toward the goal, the goal has to be perceived further off the path than it really is. Two potential solutions will be discussed, of which only the second one is suited to solve the problem.

A. Position Transformation

Figure 3(a) displays the situation after the user moved one step as in Fig. 2(b). As stated above, the user does not fully turn toward the desired orientation, but only partially. To ensure that the user turns to the desired orientation, the target environment is rotated even further with respect to the user environment as shown in Fig. 3(b). This can be achieved by the proxy turning away from the goal. The user compensates for the additional rotation by turning toward the desired angle and walks in that direction. This results in the user following the user path as shown in Fig. 3(c), but Fig. 3(d) reveals, that in the target environment the proxy's deviation to the target path still accumulates with time. Obviously, position transformation is not the adequate solution for the problem, as it does not lead to the desired behavior.

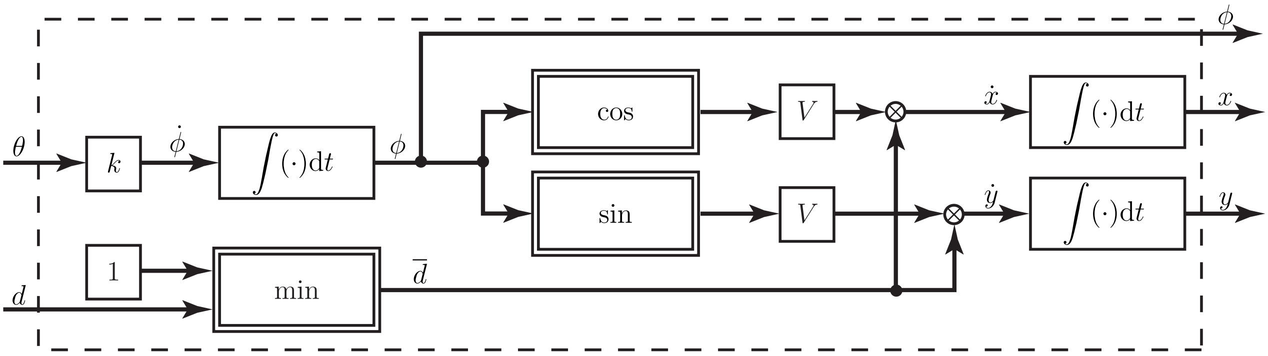
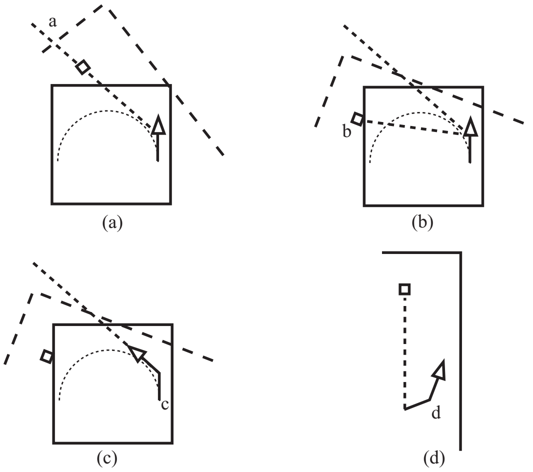


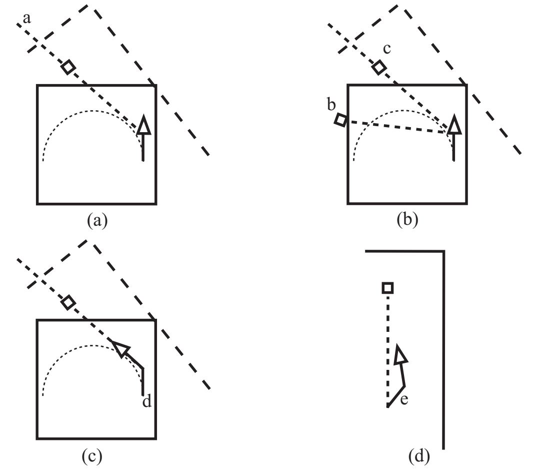
Fig. 1. The user model used for the simulation and controller design.



a... desired user orientation
c... path in user environment

b... goal after additional rotation
d... path in target environment

Fig. 3. The effect of position transformation.



a... desired user orientation
c... real goal position
e... path in target environment

b... virtual goal position
d... path in user environment

Fig. 4. The effect of view transformation.

B. View Transformation

View transformation tries to keep the key feature of position transformation (user stays on user path) by ruling out its main shortcoming (proxy leaves target path). Again Fig. 4(a) displays the situation after the user moved one step as in Fig. 2(b). To ensure that the user turns to the desired orientation, only the user's perception of the target environment is rotated resulting in a situation as shown in Fig. 4(b). This can be achieved for example by turning the proxy's head away from the goal but staying oriented toward the goal. The user now perceives a larger deviation toward the desired orientation than there really is. By trying to turn toward the virtual goal, he in fact turns toward the true goal's direction and walks toward it. This is depicted in Fig. 4(c). As above, the user follows the user path, but as shown in Fig. 4(d) the proxy also follows the target path, which is the desired behavior. Hence, view transformation is suited to solve the problem, as it keeps user as well as proxy on the desired path.

IV. CONTROLLED MOTION COMPRESSION

A. Calculation of View Transformation

The ideal additional rotation Γ can be computed if the complete user state, including speed V and the constant value k, is known, and the user is moving on the path. The distance to the goal is assumed to be large, i. e., $\overline{d} = 1$. In this case the user's turning speed must be equal to the angular speed ω of the target path with respect to the user path, i. e., $\dot{\phi} \overset{!}{=} \omega$. The relative angular speed of the two environments can be calculated from the user's speed and the radius of the target path $r = r(t)$ as $\omega = \frac{V}{r}$. The condition

$$\omega \overset{!}{=} k(\theta + \Gamma) \tag{3}$$

results in the optimal additional rotation

$$\Gamma = \frac{V}{kr} - \theta\,. \tag{4}$$

For a scenario with the user starting on the path heading toward the goal, i. e., $\theta = 0$, and the user path having a constant radius $r = 2\,\mathrm{m}$, the angle Γ is given for typical values $V = 1\,\frac{\mathrm{m}}{\mathrm{s}}$ and $k = 4.5\,\frac{1}{\mathrm{s}}$ by

$$\Gamma \approx 0.11 \approx 6.4^\circ . \tag{5}$$

B. Controller Design

In typical applications, however, the system is not stationary as above, especially $r(t)$ is not constant, and some of the parameters, e.g. k, are only approximately known. In addition, deviations from the path occur, making direct calculation of the angle Γ impossible. Hence, we derive a feedback controller, that modifies the user's perception of the target environment depending on the user's state and its deviation from the desired path. This is shown in Fig. 5. Note, that from the user's point of view, motion compression has the effect of a perturbation.

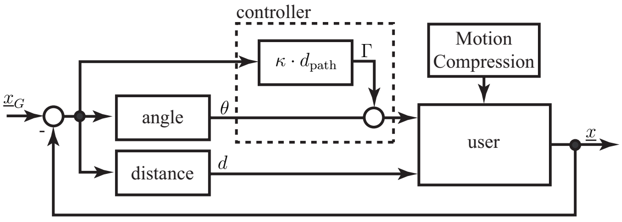


Fig. 5. Model of the controlled user.

As a first approach, the additional rotation Γ was chosen to be proportional to the signed distance from the desired path d_{path}. This can be expressed as $\Gamma = \kappa \cdot d_{\text{path}}$ with a positive value κ. Hence, the model of the controlled user can be expressed as

$$\dot{\underline{x}} = \begin{bmatrix} V\overline{d}\cos(\phi) \\ V\overline{d}\sin(\phi) \\ 0 \end{bmatrix} + k \begin{bmatrix} 0 \\ 0 \\ \theta + \Gamma \end{bmatrix} . \tag{6}$$

C. Proof of stability

In order to obtain a stable controller, we must ensure, that the generalized squared distance from the goal position

$$G(\underline{x}) = \frac{1}{2}\left(a \cdot (x_G - x)^2 + b \cdot (y_G - y)^2 + c \cdot (\phi_G - \phi)^2\right) \tag{7}$$

is decreasing, where a, b, c are positive constants.

For simplification, we assume without loss of generality, that $\underline{x}_G = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$ and that the user is situated in the fourth quadrant, i. e., $x, y \le 0$, with the negative x-Axis being the desired path. In this case b is the weight for the distance to the desired path $d_{\text{path}} = -y$, which should be minimized by the controller. Thus, we let $a = c = 1$ and choose b to be a positive value. The squared generalized distance to goal position simplifies to

$$G(\underline{x}) = \frac{1}{2}\left((-x)^2 + b \cdot (-y)^2 + (-\phi)^2\right) . \tag{8}$$

The user model only captures the case of a user walking toward a goal, i. e., he is roughly oriented toward the goal, and thus we can assume $\theta \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. The model for a user wandering freely about the target environment would be more complicated and exceed our needs. As the deviations from the path are relatively small, we can also assume, that the user's deviation from the path is smaller than the distance to the goal on the path and thus $|y| < |x|$, resulting in $\alpha = \operatorname{atan2}(-x, -y) \in \left[0, \frac{\pi}{4}\right]$. As the user typically stops before actually reaching the goal, it is safe to assume $d > \sqrt{2}$ and thus $x, y \le -1$ and $\overline{d} = 1$. Figure 6 shows the region considered here.

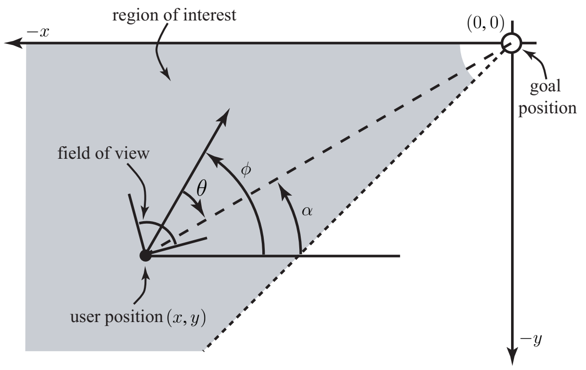


Fig. 6. Relationship of values used for the proof of stability.

With $G(\underline{x}) > 0$ for all $\underline{x} \neq \underline{0}$ and $G(\underline{0}) = 0$, G is a Lyapunov's function. The controller is stable, if $\dot{G}(\underline{x}) < 0$ for all $\underline{x}$ in the region described above. The derivative of $G(\underline{x})$ with respect to time is given as

$$\dot{G}(\underline{x}) = \frac{\partial G}{\partial \underline{x}^{\mathrm{T}}}\dot{\underline{x}}(\underline{x}) = \begin{bmatrix} x & b \cdot y & \phi \end{bmatrix} \dot{\underline{x}} \tag{9}$$

resulting in

$$\dot{G}(\underline{x}) = Vx\cos(\phi) + bVy\sin(\phi) + k\phi\theta - \kappa k\phi y \overset{!}{<} 0 . \tag{10}$$

It is easy to see, that $\dot{G}(\underline{x}) < 0$ if $\phi = 0$. For the remaining cases we now derive a value κ that provides stability depending on the various constants in the user model.

a) 1. Case: θ is negative, i. e., $\theta \in \left[-\frac{\pi}{4}, 0\right)$, resulting in $\phi \in \left[0, \frac{\pi}{2}\right]$.

$$\begin{aligned} 0 &> Vx\cos(\phi) + bVy\sin(\phi) + k\phi\theta - \kappa k\phi y \\ &= \underbrace{Vx\cos(\phi)}_{\le 0} + \underbrace{y}_{<0} \cdot (bV\sin(\phi) - \kappa k\phi) + \underbrace{k\phi\theta}_{\le 0} . \end{aligned} \tag{11}$$

The right side of equation 11 is negative if

$$bV\sin(\phi) - \kappa k\phi > 0 . \tag{12}$$

This is true for

$$\kappa < \frac{bV}{k}\frac{\sin(\frac{\pi}{2})}{\frac{\pi}{2}} \le \frac{bV}{k}\frac{\sin(\phi)}{\phi} . \tag{13}$$

b) 2. Case: $\theta \in \left[0, \frac{\pi}{4}\right]$ and $\phi \in \left(0, \frac{\pi}{4}\right]$. In order to show that

$$0 \overset{!}{>} \underbrace{Vx\cos(\phi)}_{\le 0} + y \cdot (bV\sin(\phi) - \kappa k\phi) + k\phi\theta , \tag{14}$$

it is sufficient to find a κ in such a way that

$$0 > y \cdot (bV\sin(\phi) - \kappa k\phi) + k\phi\theta . \tag{15}$$

This is true for

$$\kappa < \frac{bV}{k}\frac{\sin(\phi)}{\phi} + \frac{\theta}{y}\,. \tag{16}$$

Because of $y < -1$ we find κ as

$$\kappa < \frac{Vb}{k}\frac{\sin(\frac{\pi}{4})}{\frac{\pi}{4}} - \frac{\pi}{4} \leq \frac{Vb}{k}\frac{\sin(\phi)}{\phi} - \theta < \frac{Vb}{k}\frac{\sin(\phi)}{\phi} + \frac{\theta}{y}\,. \tag{17}$$

c) 3. Case: For the case of $\theta \in \left(0, \frac{\pi}{4}\right]$ and $\phi \in \left[-\frac{\pi}{4}, 0\right)$, we have $\dot{G} < 0$ if

$$\kappa > \frac{Vx\cos(\phi)}{ky\phi} + \frac{Vb}{k}\frac{\sin(\phi)}{\phi} + \frac{\theta}{y}\quad. \tag{18}$$

For typical values $b = 10$, $k = 4.5\,\frac{1}{\mathrm{s}}$, and $V = 1\,\frac{\mathrm{m}}{\mathrm{s}}$, a solution, that was found numerically is $\kappa > 0$.

Given the three cases above, we conclude

$$0 < \kappa < 1.22 \tag{19}$$

with constants chosen to be the same as in the third case. For the controller $\kappa = 1$ was selected.

V. EXPERIMENTAL VERIFICATION

In order to verify our controller and to identify the parameters in the user model we conducted several experiments with human users as well as the user model.

A. Experimental Setup

The user environment is of size $4\text{x}4\,\mathrm{m}^2$, in which the user can move freely. This space is bordered by a $0.5\,\mathrm{m}$ wide security margin, which can be used by the user, too. For good immersion, the user wears a high quality head mounted display with a resolution of $1280\text{x}1024$ Pixels per eye that displays a field of view of approximately $60°$. The user position is tracked by an acoustic tracking system, that provides approximately 20 position and orientation updates per second.

For the experiments, we use a virtual target environment modeled with the GNU/Maverik toolkit [7]. In order to minimize the effects of the environment on the user the target environment was modeled as a large empty plane with a single goal object in a distance of $50\,\mathrm{m}$ from the proxy.

The experimental setup provides a CORBA-Interface that allows to put a simulation in the place of user, tracker and virtual environment and thus allows testing the proposed algorithms with the user model.

B. Parameter Identification

In order to identify the parameter k, the same experiment was conducted multiple times with one user and standard motion compression. From the data generated by the experiment for each run, the largest deviation of the user was computed. k was found by comparing the largest deviation of the simulation with the mean of the largest deviation from the user's runs. For $k = 4.5\,\frac{1}{\mathrm{s}}$ the user model has a largest deviation of $-2.20\,\mathrm{m}$, the mean largest deviation of five runs by the same human user was $-2.32\,\mathrm{m}$. Figure 7 shows a comparison of the human user's path during his five runs and the user model's path. It is clearly visible that the model gives a good approximation of the user's overall behavior. The differences in detail, however, are due to the simple model and variations in the parameter k and the speed V.

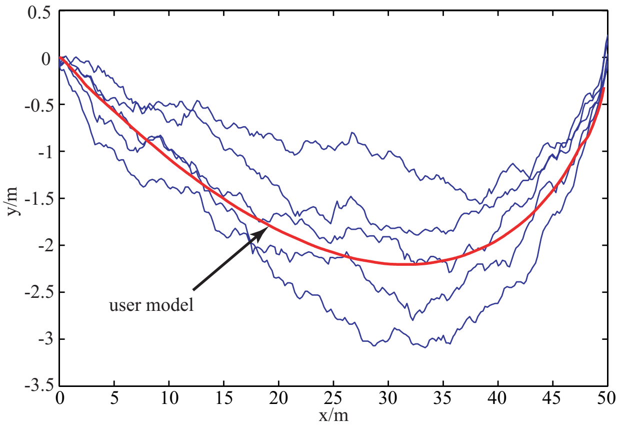


Fig. 7. Five runs of a human user compared to the simulated user.

C. Controlled Motion Compression

In order to verify controlled motion compression the same experiment was conducted with controlled motion compression. As the current implementation of path prediction interferes with the controller, the path was assumed to be a straight line from the user's starting point to the goal object.

In a first experiment, controlled motion compression was tested with the user model. Again k was chosen as $4.5\,\frac{1}{\mathrm{s}}$ and κ was set to 1. Figure 8 gives a comparison of the model's path in the target environment using controlled motion compression with its path using SMC. The largest deviation from the desired path with controlled motion compression is $-0.11\,\mathrm{m}$.

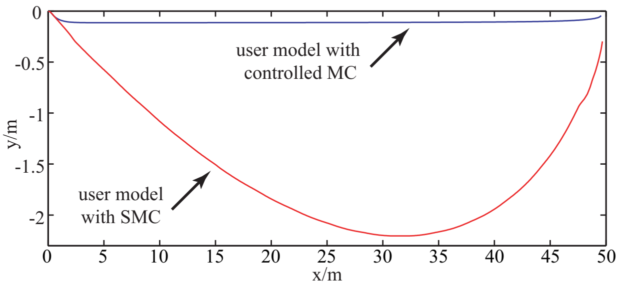
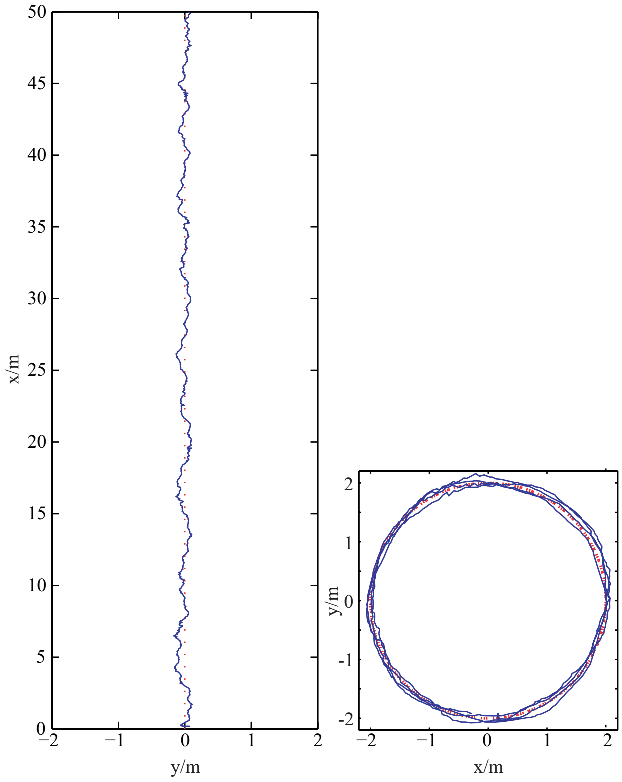


Fig. 8. Simulated target path with SMC and with controlled motion compression.

The same experiment was conducted several times with a human user. Figure 9 shows the user's behavior, when using controlled motion compression. It is clearly visible, that user and proxy closely follow the user path and the target path, respectively. Especially, the user does not leave the user environment. This proves the soundness of our approach of controlling a human user by an additional view transformation. In five runs the largest deviation of the user from the desired path was $-0.2\,\mathrm{m}$ which is significantly smaller than the deviation using SMC as discussed in Section V-B. Mean and variance of the deviation from these five runs are given as $-0.04\,\mathrm{m}$ and $0.005\,\mathrm{m}^2$, respectively.



(a) (b)

Fig. 9. Path of human user when using controlled motion compression in target environment (a) and user environment (b).

Figure 10 shows the value of the control input Γ over the path length. For the user model, Γ quickly reaches a value of 0.11 which corresponds to the value determined theoretically in Section IV-A. For the human user, Γ typically is smaller than 0.2.

VI. CONCLUSION

In this paper, we presented controlled motion compression, an extension to the standard motion compression algorithm. In standard motion compression, users tend to leave the desired user path and eventually the user environment. This problem originates from a combination of human navigation and SMC's user guidance module.

Controlled motion compression adds a feedback controller to the user guidance module in order to compensate for the deviations introduced by motion compression. The controller follows the paradigm of view transformation.

The experiments show, that controlled motion compression is superior to SMC, if the goal and the desired path to the goal are known. In order to gain flexibility the path prediction module will have to be modified to satisfy the requirements of controlled motion compression.

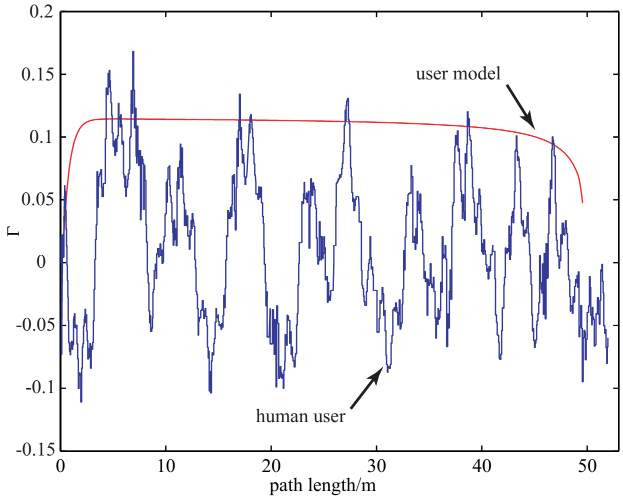


Fig. 10. Value of the control input Γ over the path length.

Controlled motion compression is especially important as future plans include multi-user environments, where several users share one user environment. For collision avoidance and path planning, knowing the user paths and ensuring that they are followed exactly is a main prerequisite.

Controlling a human user by view transformation is not only limited to telepresence systems using motion compression. It can also be used to guide users on arbitrary paths in telepresence applications using other kinds of input or even in augmented reality scenarios.

ACKNOWLEDGMENTS

This work was supported in part by the German Research Foundation (DFG) within the Collaborative Research Centers SFB 588 on “Humanoid robots – learning and cooperating multimodal robots” and SFB 453 on “High-fidelity telepresence and teleaction”.

REFERENCES

[1] T. Fong and C. Thorpe, “Vehicle Teleoperation Interfaces,” *Autonomous Robots*, vol. 11, no. 1, pp. 9–18, 2001.

[2] N. Nitzsche, U. D. Hanebeck, and G. Schmidt, “Motion Compression for Telepresent Walking in Large-Scale Remote Environments,” in *Proceedings of SPIE Vol. 5079, AeroSense Symposium*, Orlando, Fl, 2003, pp. 265–276.

[3] J. J. LaViola Jr., D. A. Feliz, D. F. Keefe, and R. C. Zeleznik, “Hands-free Multi-Scale Navigation in Virtual Environments,” in *Symposium on Interactive 3D Graphics*, Chapel Hill, NC, 2001, pp. 9–15.

[4] M. Slater, M. Usoh, and M. Steed, “Steps and Ladders in Virtual Reality,” in *Proceedings of the ACM Conference on Virtual Reality Software and Technology*, Singapore, 1994, pp. 45–54.

[5] S. K. Rushton, J. M. Harris, M. R. Lloyd, and J. P. Wann, “Guidance of Locomotion on Foot Uses Perceived Target Location Rather Than Optic Flow,” *Current Biology*, vol. 8, no. 21, pp. 1191–1194, 1998.

[6] N. Nitzsche, U. D. Hanebeck, and G. Schmidt, “Motion Compression for Telepresent Walking in Large Target Environments,” *Presence*, vol. 13, no. 1, pp. 44–60, 2004.

[7] “http://aig.cs.man.ac.uk/maverik/.”