Open access • Proceedings Article • DOI:10.1109/IROS.2006.281656

Simultaneous Motion Compression for Multi-User Extended Range Telepresence — Source link

Patrick Robler, Uwe D. Hanebeck

Institutions: Karlsruhe Institute of Technology

Published on: 01 Oct 2006 - Intelligent Robots and Systems

Topics: Mobile robot navigation, Mobile robot, Robot control, Telerobotics and Social robot

Simultaneous Motion Compression for Multi-User Extended Range Telepresence

Patrick Rößler and Uwe D. Hanebeck
Intelligent Sensor-Actuator-Systems Laboratory
Institute of Computer Science and Engineering
Universität Karlsruhe (TH)
Karlsruhe, Germany
{*patrick.roessler, uwe.hanebeck*}*@ieee.org*

Abstract— **Extended range telepresence allows a human user to intuitively teleoperate a mobile robot through arbitrarily large remote environments by natural walking. In order to give the user the possibility to navigate the robot through an arbitrarily large remote environments, while his own environment is of limited size, Motion Compression is used. The Motion Compression framework provides a nonlinear transformation between the user's path and the robot's path, which preserves path length and turning angles. There is, however, a difference in path curvature, which is minimized in order to guarantee a high degree of immersion.**

A major drawback of the current system is its inability to deal with non-convex time-variant environments or environments shared by multiple users. This paper presents a systematic approach to extending Motion Compression to non-convex environments. This solution will then be used to cover the multi-user case.

I. Introduction

Telepresence systems aim at giving a human user the impression of actual presence in a remote target environment. To achieve this goal, a robot that represents the user is placed in the target environment. This robot, the *teleoperator*, records live camera images and other sensor information, e. g. acoustics, and transfers them to the user through an internet connection in real-time. The sensor information is presented to the user on an immersive display, e. g. a head-mounted display.

In order to give the user the possibility to move in the target environment, his head motion is tracked and transferred to the teleoperator, which replicates this motion. As the user only perceives the target environment and his motion matches the visual cues, he immerses into the target environment.

In order to allow exploration of large target environments, mobile robots, typically wheel-based platforms equipped with a pan-tilt camera unit, are used as teleoperators. In conventional telepresence systems the wide-area motion of the platform is controlled with additional input devices like joysticks [1], foot pedals [2], or steering wheels [3], while only the rotational motion of the user's head is tracked and replicated by the pan-tilt unit.

Those conventional systems do not provide proprioceptive feedback, i. e., the sense of self motion. However, there is evidence, that proprioception is important for human navigation [4]–[6]. As a consequence, users should navigate the teleoperator through the remote environment by *natural walking* and, thus, receive proprioceptive feedback.

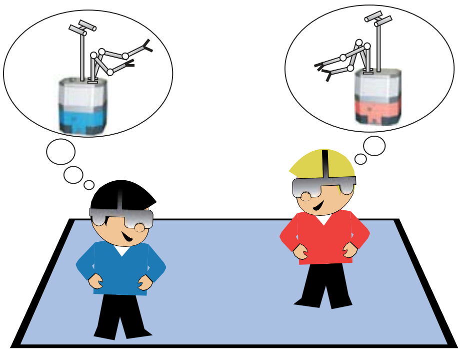

Fig. 1. Two users sharing a multi-user environment for telepresence. As they are both immersed into their respective target environments, they are not aware of each other.

Typically the local environment of the user, the *user environment*, is limited for example by the range of the tracking system or just by available space, but the target environment is of arbitrary size and shape. In order to give the user the possibility to *not* only explore target environments of the user environment's size and shape, current research aims at interfaces that allow infinite walking in limited spaces.

There are various mechanical approaches that try to provide the impression of walking on an infinite floor. Examples for actively controlled mechanical systems are the torus treadmill [7] or the CirculaFloor system [8], where the user is walking on moving floor tiles, which move him closer to the center of the user environment. In other work the user is passively guided back to the center of the user environment, e. g. by walking on a disc that is equipped with ball bearings [9]. Those systems, however, require complex mechanical setups.

More promising work aims at algorithmic solutions to the infinite floor problem. Those algorithms map the user's path in the target environment into the user environment. In order to guarantee a high degree of immersion, as many properties of the path as possible should be preserved. These are especially path length and turning angles. Scaling of the path is obviously no adequate solution. There are basically two approaches to mask path changes.

Redirected Walking [10] masks path changes in rotations. However, walking along straight paths for a longer time cannot be mapped with this technique.

Motion Compression [11] preserves both path length and turning angles. Only path curvature is changed. Motion Compression has been formulated as a dynamic optimization problem to minimize the curvature difference between the path in the target environment and the path in the user environment.

However, Motion Compression so far has high spatial requirements, as the user environment has to be an empty convex area. In order to achieve a better exploitation of the available space, Motion Compression has to be modified to support non-convex user environments.

An additional gain can be attained, if multiple users share one user environment. Such a scenario is depicted in Fig. 1. As the users are completely immersed into their respective target environments, they are not aware of each other. Thus, Motion Compression has to prevent collisions by simultaneous compression of both paths. So far, there is only a highly heuristic approach to this problem [12], yielding very suboptimal solutions. In this paper, we modify the original optimization problem systematically in order to provide good results for simultaneous Motion Compression.

The remainder of this paper is structured as follows. A short introduction into the algorithmic framework of Motion Compression is given in section II. Section III-A reviews the path transformation problem and formulates necessary extensions for the non-convex case. A practical solution to the path transformation problem is given in section IV. This solution is modified to meet the requirements of non-convex user environments. In section V these modifications are applied to solve the multi-user case. The validity of the algorithms is shown with simulations in section VI. Finally, conclusions are drawn.

II. Motion Compression

Motion Compression is an algorithmic framework, that provides a nonlinear transformation between the user's path in the user environment and the teleoperator's path in the target environment. It consists of three functional modules, as shown in Fig. 2. For each of the three modules specialized instances can be loaded, in order to fit the given application.

The first step in Motion Compression is *path prediction*. Path prediction tries to predict the desired path of the user in the target environment based on tracking data and, if available, additional information on the target environment. In many experiments Motion Compression has proven to be robust against errors in path prediction.

This predicted path is transformed in *path transformation* in such a way that it fits into the user environment. Path transformation is formulated as a dynamic optimization problem, which will be discussed in detail in section III-A. The resulting transformed path is called *user path*.

Finally, *user guidance* guides the user on the user path while he has the impression of walking on the original path in the target environment. This is achieved by introducing small deviations in the teleoperators orientation, which are compensated by the user.

The result of these three steps is at any time and position a linear transformation between user environment and target environment [11]. This transformation can now be used to transform the user's position into the desired position of the mobile teleoperator in the target environment. The suitability of this concept was proven in [13].

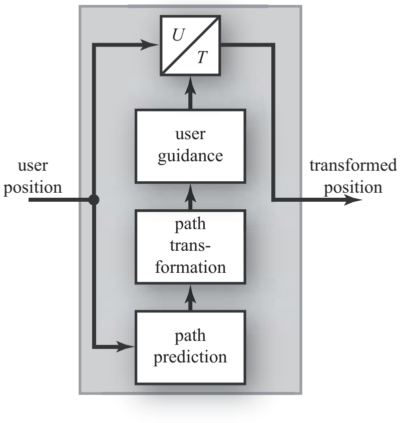

Fig. 2. Overview of the Motion Compression Framework.

III. The Path Transformation Problem

A. The General Optimization Problem

In [11] path transformation is formulated as a dynamic optimization problem. The user's state in cartesian space is given as the user's position and orientation $\underline{x} = [x, y, \phi]^{\mathrm{T}}$, with the initial state $\underline{x}_0 = [x_0, y_0, \phi_0]^{\mathrm{T}}$. While the user's state is given in cartesian space, the predicted path in the target environment is described as a curvature $\kappa_T = \kappa_T(s)$ depending on a continuous path variable $s = 0 \ldots s_E$.

The optimal user path is then given as the path $\kappa = \kappa(s)$, that minimizes the overall curvature difference. This corresponds to the optimization criterion

$$J = \int_0^{s_E} (\kappa_T - \kappa)^2 \, ds \quad . \tag{1}$$

J has to be minimized regarding the functional relationship between s and $\underline{x}$ and a set of spatial constraints to the user path.

The functional relations are given as

$$\begin{aligned} \frac{dx}{ds} &= \dot{x} = \cos\phi \ , \\ \frac{dy}{ds} &= \dot{y} = \sin\phi \ , \text{ and} \\ \frac{d\phi}{ds} &= \dot{\phi} = \kappa \ . \end{aligned} \tag{2}$$

Spatial constraints can be modeled as a set of inequality constraints in the form of

$$\underline{g}(x, y) \geq 0 \quad . \tag{3}$$

Each of the components $g_i(x, y)$ describes the relationship between the user's current position and one obstacle, e. g. a boundary of the user environment.

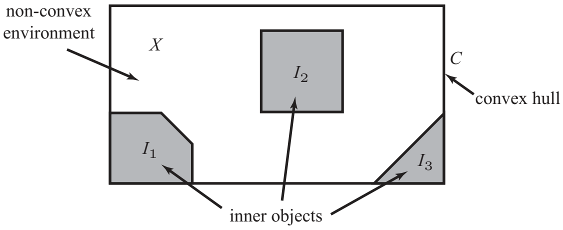

Fig. 3. Example of a non-convex environment X described as by means of its convex hull C and some inner objects $I_1 \dots I_3$.

B. Describing Arbitrary Environments

In order to be able to extend the optimization problem to non-convex user environments, an efficient description of such environments has to be used. It is a well known fact, that arbitrary non-convex user environment X can be described by a set of convex areas as

$$X = C - I_1 - \dots - I_n \ , \tag{4}$$

where C is the convex hull of X and I_i are convex descriptions of the inner objects. This is shown in Fig. 3. If X is a convex environment the equation $X = C$ is valid.

The description in equation (4) can be used to specify several types of inequality constraints for non-convex user environments.

The first set of constraints

$$\underline{g}_C(x, y) \geq 0 \tag{5}$$

describes, that the path is inside the convex hull of the user environment X.

In order to guarantee, that the user path does not intersect the boundaries of the inner objects, an additional set of constraints

$$\underline{g}_{I_i}(x, y) \geq 0 \tag{6}$$

is added for every inner object I_i.

The complete set of constraints is then given as

$$\underline{g}(x, y) = \begin{bmatrix} \underline{g}_C(x, y) \\ \underline{g}_{I_1}(x, y) \\ \vdots \\ \underline{g}_{I_n}(x, y) \end{bmatrix} \geq 0 \ . \tag{7}$$

Unfortunately, there is no analytical solution known for the general path transformation problem.

IV. The γ-Arc Algorithm

A. Convex Environments

Although the general problem has no analytic solution, it is possible to find a heuristic, that solves a simplified problem. The γ-Arc algorithm, which is a generalization of the semicircle algorithm presented in [11], gives a suboptimal, yet satisfactory solution for path transformation.

Given the assumptions, that

- the user environment X is convex, i. e., $X = C$ and, thus, $\underline{g}(x, y) = \underline{g}_C(x, y)$,
- the predicted path in the target environment is a straight line, i. e., $\kappa_T = 0$ starting at $\underline{x}_0$, and
- the path in the target environment has no fixed endpoint, or the endpoint is far away from the user,

the optimal user path is given as follows.

The user path starts at the current position and orientation $\underline{x}_0$ of the user. The path is the *largest circle segment* fitting into the user environment, where the parameter $\gamma \in (0, 2\pi]$ determines the segment length. This circle segment can be easily found geometrically. Fig. 4 compares the circle segments found for different choices of γ. By successively re-calculating the largest γ-Arc, arbitrary paths can be generated as a series of momentary circles.

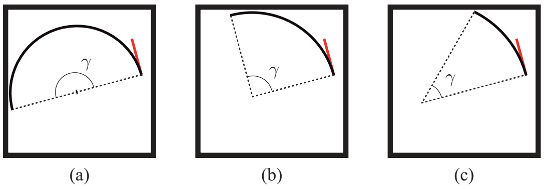

Fig. 4. Largest circle segments fitting into a quadratic user environment for different values of γ. (a) $\gamma = \pi$, (b) $\gamma = \frac{\pi}{2}$, (c) $\gamma = \frac{\pi}{4}$.

B. Extension to Non-Convex Environments

In order to be able to apply the γ-Arc algorithm to non-convex user environments, it has to be modified in such a way, that the resulting path also meets the constraints given by $\underline{g}_{I_i}$.

In a first step, the γ-Arc algorithm as described above is used to find the largest circle segment, that fits into the convex hull C of the user environment. The resulting arc is called *maxarc*. If *marcarc* does not intersect the boundaries of any of the inner objects I_i, it is considered to be the best user path. However, if there is an intersection another arc has to be calculated.

For every bounding wall of the inner objects, the *largest* circle segment, that either touches, but not intersects, the wall or leads through one of the endpoints of the wall is calculated. The *smallest* of these segments, called *maxinnner*, is the largest γ-Arc compatible with all constraints $\underline{g}(x, y) \geq 0$. A more formal description is given in Algorithm 1. Fig. 5 shows

Algorithm 1 γ-Arc algorithm for non-convex environments.

```
$maxarc \leftarrow$ GAMMA-ARC($C$)
if !INTERSECTS($maxarc, I_i$) then
    return $maxarc$
end if
for all $I_i$ do
    $inner \leftarrow$ GAMMA-ARC($I_i$)
    if ISSMALLER($inner, maxinner$) then
        $maxinner \leftarrow inner$
    end if
end for
return $maxinner$
```

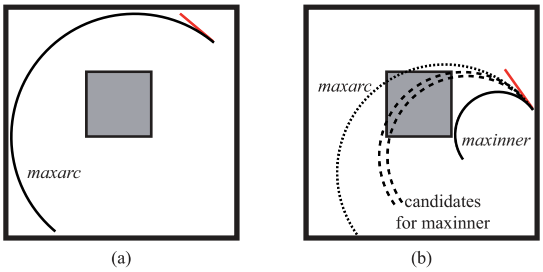

Fig. 5. Results of the γ-arc algorithm for non-convex environments with $\gamma = \pi$. (a) Case where *maxarc* is the adequate solution. (b) Case where *maxinner* has to be calculated.

the relation between *maxarc* and *maxinner*. As the number of calculations grows linear with the number of inner objects, the modified solution is still very easy to compute.

C. Choosing the Parameter γ

In this algorithm, the parameter γ can be regarded as tuning parameter that describes the trade-off between greediness and look-ahead. Typically a value of $\gamma > \pi$ is not useful, because this would lead to walls situated behind the user effecting the resulting path. A value of $\gamma \to 0$ results in a user path consisting of almost straight line segments connected by sharp turns. $\gamma = \frac{\pi}{2}$ seems to be a good choice, however, theoretical validation is still missing.

V. Multi-User Motion Compression

If Motion Compression is extended to multi-user telepresence, several users share one user environment. As all users are immersed into their respective target environments, they are not aware of each other. Thus, path transformation has to transform the paths in such a way, that no collisions between the users occur.

In this section, we show how the problem of multi-user Motion Compression can be reduced to the non-convex case.

In order to prevent other users to approach too closely, every user is enclosed by a security area, which is regarded as inner object for all other users. This security area can have any convex shape, e. g. triangular or quadratic. An optimal shape is yet to be determined. However, asymmetric shapes seem to yield the best results. As a result, at any given time, for every user the environment is a non-convex environment and the path transformation problem can be solved with the modified γ-Arc algorithm as presented in section IV-B. However, this environment changes as the other users move. Fig. 6 shows an example for two users with their security areas and their respective *maxarcs*.

As a result, the users are guided around each other, while moving along in the user environment.

VI. Simulations

The Motion Compression framework has already proven practical for mobile robot teleoperation [13] as well as for avatar control in computer games [14]. For this reason it is sufficient in this context, to show the feasibility of the new path transformation methods by means of simulations.

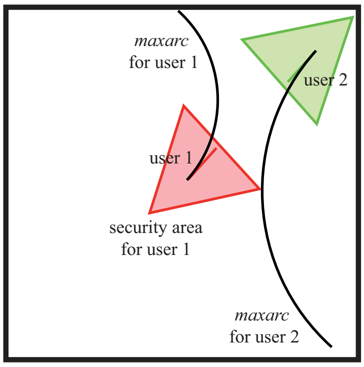

Fig. 6. Two users with their security areas (in this example in triangular shape) and their respective *maxarcs* for $\gamma = \frac{\pi}{2}$.

As real users, of course, deviate from the ideal transformed path, the inner objects should be padded with an extra security margin in real applications.

A. Non-Convex Environments

We will first show the feasibility of our approach for the non-convex case. In all simulations the user was moving in a $4 \times 4\,\mathrm{m}^2$ user environment, with $0.1\,\mathrm{m}$ step length and $\gamma = \frac{\pi}{2}$.

In such a setting without additional obstacles, a typical curvature of approximately $0.5\,\mathrm{m}^{-1}$ is expected. Obviously, inner object will raise this value.

In a first simulation a user moves straight along for 15 m in the target environment. Fig. 7(a) shows his user path, which was transformed with respect to two inner objects placed in the upper right corner and the lower left corner of the user environment. It is clearly visible, that the algorithm leads the user as close as possible to the inner objects, in order to minimize path curvature. The curvature plot shown in Fig. 7(b) shows the absolute value of the path curvature. The curvature typically stays well below a value of $1\,\mathrm{m}^{-1}$, which is a good result given the expected value for the convex case.

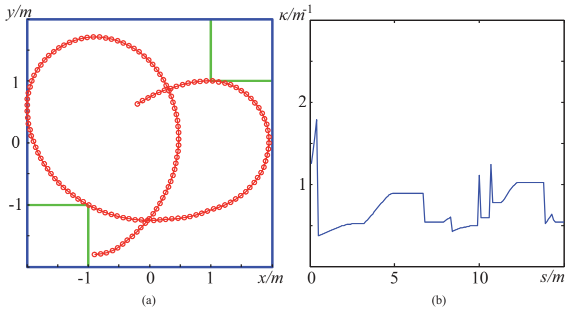

Fig. 7. (a) User path of a user walking straight along for 15 m with two inner objects in the corners of the user environment. (b) Curvature of the user path.

In order to show that more complex paths are also possible, the user performs a 90° turn in a second simulation after walking straight along for 2.5 m in the target environment. After turning he follows a straight path for another 7.5 m. The user environment features two quadratic inner objects inside the user environment. Fig. 8(a) shows, that the user path is transformed in such a way that the user is guided between the inner objects. Again, this behavior guarantees best usage of available space and minimizes path curvature. The curvature plot in Fig. 8(b) shows, that the curvature is relatively low apart from some curvature peaks. Additional curvature peaks may arise, when the user turns toward a bounding wall of the user environment. However, this is not a result of the new algorithm, but a yet unsolved problem of Motion Compression.

Fig. 9 shows a comparison between the curvature from this simulation and from the same path in a convex environment without the inner objects. It is easy to see, that the curvature peaks are due to the inner objects appearing in the look-ahead of the algorithm. A less greedy algorithm, i. e., using a larger value for γ would reduce these peaks, but lead to a higher overall curvature.

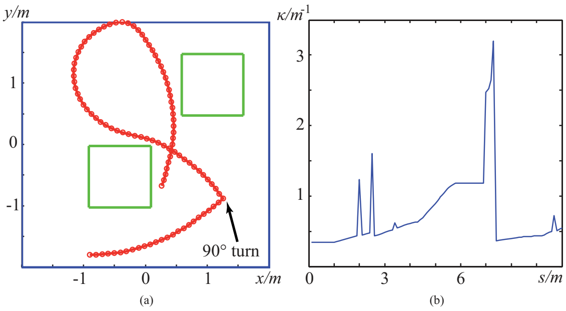

Fig. 8. (a) User path of a user walking straight along for 10 m with a 90° turn after 2.5 m. There are two inner objects inside the user environment. (b) Curvature of the user path.

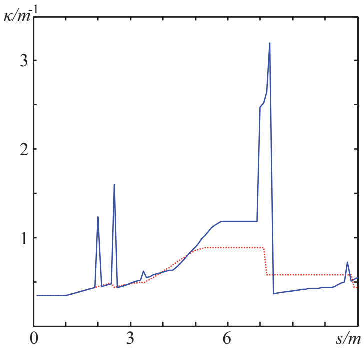

Fig. 9. Comparison between the curvatures in the non-convex case (solid blue line) from Fig. 8 and the same path in a convex user environment (dotted red line) without the inner objects.

The simulation in Fig. 10 shows a rather extreme case of a non-convex environment. The available space is a 0.5 m wide corridor around a large quadratic inner object. Fig. 10(a) shows, that the algorithm still finds a valid user path and that even steep turns are still possible. Of course, path curvature is relatively high (Fig. 10(b)).

In this case, using a smaller value for γ might be the better choice. This would lead to fewer but higher curvature peaks. The effects on immersion are still to be investigated.

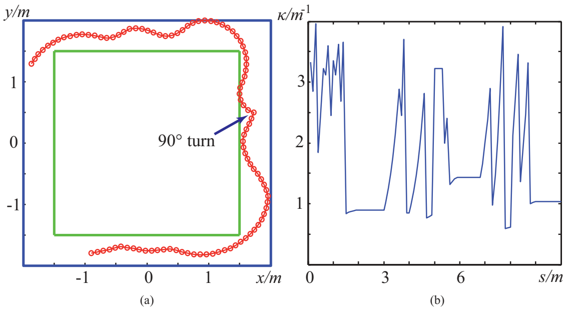

Fig. 10. (a) User path of a user walking straight along for 10 m with a 90° turn after 5 m. The user environment is massively limited by a large inner object. (b) Curvature of the user path.

B. Multi-User Environments

Additional simulations were performed for the multi-user case, in order to show that the proposed approach meets the requirements.

Fig. 11 shows the results of a simulation where two users share one user environment. The target path for both users is a straight line with a length of 10 m. Both users are surrounded by a quadratic security area of $1 \times 1\,\mathrm{m}^2$ aligned parallel to the user. The user paths for user 1 (red circles) and user 2 (green diamonds) are shown in Fig. 11(a). Figures 11(b) and (c) show the path curvature for user 1 and user 2, respectively. Curvature is typically below $1\,\mathrm{m}^{-1}$ with only very few curvature peaks.

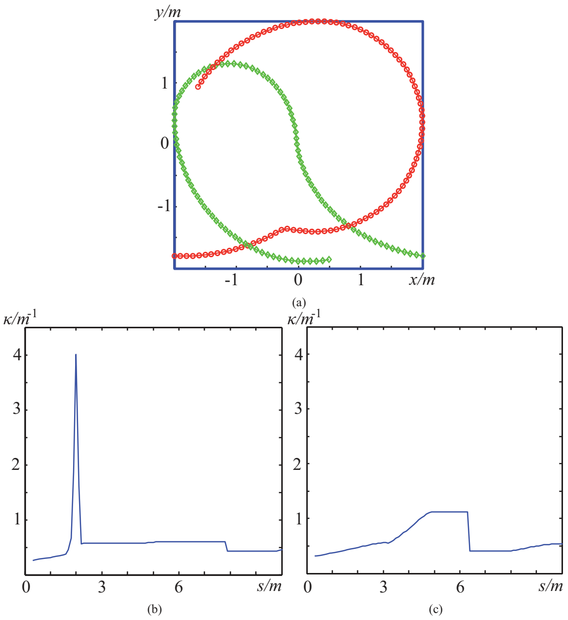

Fig. 11. (a) Simultaneously transformed paths for two users, both walking straight along for 10 m. (b) Curvature of user path for user 1. (c) Curvature of user path for user 2.

A second simulation for two users in the same environment is shown in Fig. 12. Both target paths feature a 90° turn. This is reflected in the users' transformed paths in Fig. 12(a). The curvature plots Fig. 12(b) and (c) show that both users experience a curvature only minimally higher than for a single user moving along the same path.

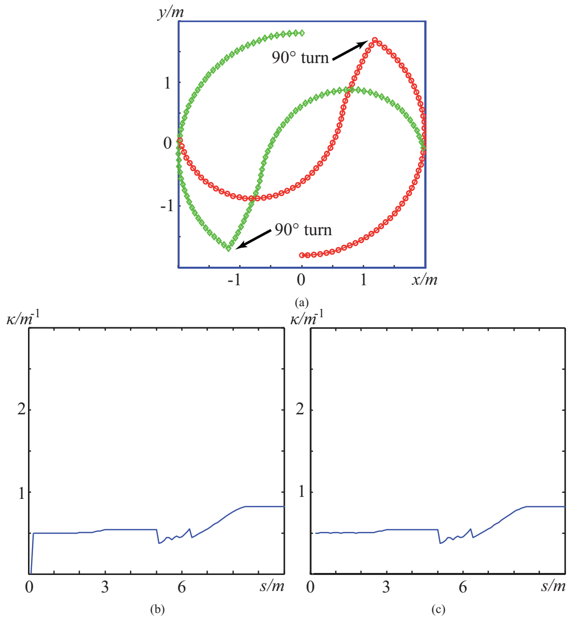

Fig. 12. (a) Simultaneously transformed paths for two users, both following a straight path for 5 m before performing a 90° turn and then moving straight along for another 5 m. (b) Curvature of user path for user 1. (c) Curvature of user path for user 2.

VII. Conclusions

This paper presents novel extensions to Motion Compression for extended range telepresence. The Motion Compression framework allows human users to intuitively control mobile robots robots through arbitrarily large target environments by natural walking in a limited user environment. This is achieved by providing a nonlinear transformation between the user's path in the user environment and the robot's path in the target environment. This nonlinear transformation preserves path length and turning angles, curvature difference is kept at a minimum level.

The framework can currently only deal with convex single-user environments. The new extensions, however, also allow non-convex user environments. This was achieved by reformulating the original optimization problem and modifying the algorithmic solution accordingly. The results for non-convex environments were also used to extend Motion Compression to multi-user environments.

In various simulations for both the non-convex case and the multi-user case, we showed the feasibility of the algorithms. Overall curvature of the paths is only minimally higher than with convex environments. In addition, there are only few curvature peaks. Thus, immersion will not be lost.

Of course, the solutions presented in this paper will be integrated into the existing telepresence system at the Intelligent Sensor-Actuator-Systems Laboratory at the Universität Karlsruhe (TH) in the near future. Experiments will be conducted to verify the simulation results.

Future work will focus on multi-user environments. There are still open questions, e. g. if and how collision avoidance can be guaranteed under all circumstances. Another topic is the systematic determination of the tuning parameter γ for a given user environment. Adaptive adjustments to this value might yield the best solutions.

One important direction of research aims at finding a more general solution for the path transformation problem, that minimizes curvature difference for arbitrary paths with steep direction changes or wide curves.

Given its features, extended range telepresence can be considered as the modern approach to robot teleoperation, where the users do not have to focus on the act of teleoperation, but on the task that is to be solved in the target environment.

Acknowledgment

This work was supported in part by the German Research Foundation (DFG) within the Collaborative Research Center SFB 588 on "Humanoid robots–learning and cooperating multimodal robots".

References

[1] R. Stemmer, R. Brockers, S. Drüe, and J. Thiem, "Comprehensive Data Acquisition for a Telepresence Application," in *Proceedings of Systems, Man, and Cybernetics*, The Hague, The Netherlands, 2004, pp. 5344–5349.

[2] S. M. Goza, R. O. Ambrose, M. A. Diftler, and I. M. Spain, "Telepresence Control of the NASA/DARPA Robonaut on a Mobility Platform," in *Proceedings of ACM International Conference on Human Factors in Computing Systems (CHI 2004)*, Vienna, Austria, 2004, pp. 623–629.

[3] C. Bunz, M. Deflorian, C. Hofer, F. Laquai, M. Rungger, F. Freyberger, and M. Buss, "Development of an Affordable Mobile Robot for Teleexploration," in *Proceedings of IEEE Mechatronics & Robotics (MechRob'04)*, Aachen, Germany, 2004, pp. 865–870.

[4] R. P. Darken, T. Allard, and L. B. Achille, "Spatial Orientation and Wayfinding in Large-Scale Virtual Spaces: An Introduction," *Presence*, vol. 7, no. 2, pp. 101–107, 1998.

[5] B. Peterson, M. Wells, T. A. Furness III, and E. Hunt, "The Effects of the Interface on Navigation in Virtual Environments," in *Proceedings of Human Factors and Ergonomics Society 1998 Annual Meeting*, vol. 5, 1998, pp. 1496–1505.

[6] M. J. Tarr and W. H. Warren, "Virtual Reality in Behavioral Neuroscience and Beyond," *Nature Neuroscience Supplement*, vol. 5, pp. 1089–1092, 2002.

[7] H. Iwata, "The Torus Treadmill: Realizing Locomotion in VEs," *IEEE Computer Graphics and Applications*, vol. 19, no. 6, pp. 30–35, 1999.

[8] H. Iwata, H. Yano, H. Fukushima, and H. Noma, "CirculaFloor," *IEEE Computer Graphics and Applications*, vol. 25, no. 1, pp. 64–67, 2005.

[9] J.-Y. Huang, "An Omnidirectional Stroll-Based Virtual Reality Interface and Its Application on Overhead Crane Training," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 39–51, 2003.

[10] S. Razzaque, Z. Kohn, and M. C. Whitton, "Redirected Walking," in *Proceedings of Eurographics*, Manchester, UK, 2001.

[11] 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.

[12] J. Su, "Optimal Incremental Approach to the Motion Compression for Telepresent Locomotion," in *Proceedings of the IEEE Intl. Conference on Robotics and Automation (ICRA'04)*, New Orleans, LA, USA, 2004, pp. 1664–1669.

[13] P. Rößler, F. Beutler, U. D. Hanebeck, and N. Nitzsche, "Motion Compression Applied to Guidance of a Mobile Teleoperator," in *Proceedings of the IEEE Intl. Conference on Intelligent Robots and Systems (IROS'05)*, Edmonton, AB, Canada, 2005, pp. 2495–2500.

[14] P. Rößler, F. Beutler, and U. D. Hanebeck, "A Framework for Telepresent Game-Play in Large Virtual Environments," in *2nd Intl. Conference on Informatics in Control, Automation and Robotics (ICINCO 2005)*, vol. 3, Barcelona, Spain, 2005, pp. 150–155.