

Where are you going? Using human locomotion models for target estimation

Journal Article**Author(s):**

Zank, Markus; Kunz, Andreas 

Publication date:

2016-10

Permanent link:

<https://doi.org/10.3929/ethz-a-010634858>

Rights / license:

[In Copyright - Non-Commercial Use Permitted](#)

Originally published in:

The Visual Computer 32(10), <https://doi.org/10.1007/s00371-016-1229-9>

Funding acknowledgement:

153243 - Development and Validation of a Virtual Reality Simulator including Haptics and Walking for Studies in Consumer Behavior Research (SNF)

Where are you going? Using human locomotion models for target estimation

Markus Zank · Andreas Kunz

Received: date / Accepted: date

Abstract To explore virtual environments that are larger than the available physical tracking space by real walking, it is necessary to use so-called redirected walking. Redirection techniques allow the user to explore an unlimited virtual environment in a limited tracking space by introducing a small mismatch between a user's real and virtual movement, thus preventing the user from colliding with the physical walls of the tracking space. Steering algorithms are used to select the most suitable redirection technique at any given time, depending on the geometry of the real and virtual environment. Together with prediction of a user's future walking path, these algorithms select the best redirection strategy by an optimal control scheme.

In this paper, a new approach for the prediction of a person's locomotion target is presented. We use various models of human locomotion together with a set of possible targets to create a set of expected paths. These paths are then compared to the real path the user already traveled in order to calculate the probability of a certain target being the one the user is heading for. A new approach for comparing paths with each other is introduced and is compared to three others. For describing the human's path to a given target, four different models are used and compared. In order to gather data for the comparison of the models against the real path,

a user study was conducted. Based on the results of the user study, the paper concludes with a discussion on the prediction performance of the different approaches.

Keywords virtual reality · redirection · human locomotion · prediction

1 Introduction

Virtual worlds are of great importance in engineering, architecture or medicine, but also cognitive sciences use virtual worlds as controlled environments for user studies. So far, the potential of virtual environments could not be fully utilized since people who are sitting in front of a monitor and are navigating with mouse and keyboard perform differently from those who explore the environment by real walking. It was shown previously that real walking in virtual environments improves the cognitive map, eases learning tasks and allows for a better distance estimation [8,15]. Moreover, it was shown that real walking is preferred by users over other interaction methods for navigating in virtual environments, such as walking-in-place or pointing [20]. Nabiyouni et al. compared real walking to using a gamepad or the Virtusphere locomotion device¹ and found that real walking was not only preferred by the users, but also performed better in terms of accuracy [10]. However, exploring an ideally unlimited virtual environment in a physically constrained space requires a compression of the virtual environment. To avoid reduced immersion of the user and to be able to use it as a replacement for real world experiments, this compression must not be noticeable or alter the user's behavior.

Markus Zank
Innovation Center Virtual Reality, IWF
ETH Zürich
Switzerland
E-mail: zank@iwf.mavt.ethz.ch

Andreas Kunz
Innovation Center Virtual Reality, IWF
ETH Zürich
Switzerland
E-mail: kunz@iwf.mavt.ethz.ch

¹ www.virtusphere.com

An approach to solve this issue was first introduced by Razzaque et al. under the name Redirected Walking [14]. This concept introduces a subtle mismatch between the user's movement in the real and in the virtual space, by which large virtual environments can be compressed into a limited physical tracking space. Many redirection techniques were proposed in the recent years that differ in the way they redirect the user and in the break in immersion caused by the mismatch of real and virtual trajectories [7, 13, 14, 17]. They include scaling the user's forward or rotational speed, bending straight path segments into curves, and adding so-called resets, which is a forced reorientation of the user towards the center of the tracking space. However, these techniques were mostly applied on their own and without any planner. To combine these techniques and use them interchangeably in the same setup, there is the need to plan future redirection actions depending on the current position of the user in the real and in the virtual environment.

Zmuda et al. first introduced a probabilistic planner to plan future redirection techniques [23]. The planning was then later formulated as an optimal control problem by Nescher et al. [11] who use a cost function assigned to different redirection techniques to select the optimal redirection technique at any given time. This leads to improved performance compared to simpler approaches just using one single redirection technique. However, for planning it is crucial to have a good prediction of a user's future behavior. The more different the possible future paths are, the more important the prediction becomes. Consider for example the situation depicted in Figure 1. The red frame represents the outline of the available physical tracking space, while the virtual walls are drawn in black. A person entering the hall through corridor 0 has two options. If he exits through corridor 1, he should be redirected to the left. However, if he chooses corridor 2 instead, he should be redirected to the right. In both cases, using the wrong redirection technique would lead to a collision with a wall and a reset or reorientation of the user becomes necessary, causing a major break in immersion. Since many redirection techniques can only be applied while the user is moving, his future walking trajectory is of particular interest. The earlier a decision on the most likely trajectory can be made, the more subtle the redirection can be. This shows the importance of an early estimation of a person's intended walking path. This paper thus presents novel methods that use reference paths generated by human locomotion models to estimate a person's future locomotion target.

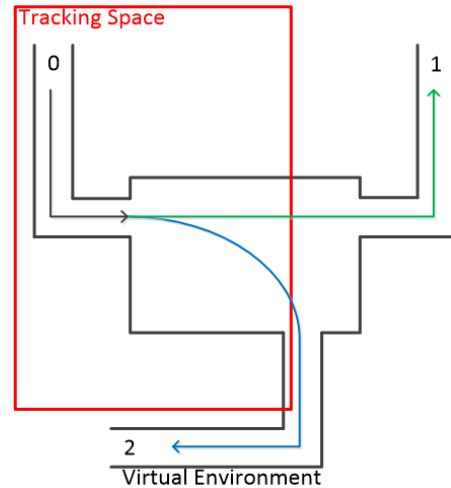


Fig. 1 Example of two possible decisions requiring redirection in opposite directions

2 Related Work

2.1 Path Prediction

To address the need for motion prediction in redirected walking, various approaches were proposed. Most of these approaches are based on the user's facing direction [13], which is in some cases combined with his direction of movement [18]. Since the direction of movement cannot be used when the user stands still, Interrante et al. [7] employed a weighted combination of gaze and movement direction depending on the user's speed. In case the user is standing still or is only moving slowly, mainly the gaze direction is used for prediction, while in case of walking the movement direction is more important. Another approach was introduced by Su et al. [19] who extrapolated the user's traveled path for the prediction.

Systems for real walking in virtual environments track the head-mounted display (HMD) worn by the user to provide the correct view in the virtual environment. It thus makes sense to use this tracking data for the prediction. However, due to the nature of human gait, there are oscillations superimposed on the head position and thus also on the movement direction, which will result in large prediction errors. To avoid this oscillation problem, the signals have to be smoothed, e.g. for 2 seconds in Interrante et al. [7]. Smoothing on the other hand introduces latencies and thus the prediction will not be able to follow fast changes in the user's movement direction anymore. To face this problem, Nescher et al. [12] used a double-exponential smoothing, whose parameters were adjusted in such a way that it was capable to follow a step function to 80% of its amplitude within 1 second.

All path prediction approaches have in common that they are derived from real-time tracking data and have no underlying model. Thus, they only give reliable prediction for a few seconds. For longer time horizons, human locomotion models are required.

2.2 Human Locomotion Models

For predicting the locomotion target, the whole path that a user traveled can be considered instead of the facing or moving direction only. This approach is based on the fact that human locomotion is known to be stereotypical [6]. Thus, the estimation of the intended target could be done by comparing the path observed from a user to other reference trajectories that were observed before in the same environment. However, this would require that every virtual scenario was tested before with other users in order to gather reference trajectories. Since this is an impractical approach, we propose to compare a user's traveled path to other reference paths that were created by human locomotion models.

There exists a large variety of human locomotion models that also differ in their complexity depending on the application. Simple Brownian approaches for example are used to handle occlusions for surveillance cameras [3], while more advanced models are used in robotics for obstacle avoidance [21] or for planning a robot's path in an environment.

More realistic human locomotion models are usually derived from a set of measured data from traveled paths. This data is then used to tune the parameters of an artificial locomotion model in order to minimize the deviation between model and observation. Once a certain start and target is given, these locomotion models can generate an expected ideal walking path.

In this paper, we propose to use these models to create reference paths to a number of given targets and to continuously compare a user's observed trajectory to them as he walk to his target. Based on the similarity of the observed path and the modeled paths for different targets, an estimation can be made of which is the most likely target. For the estimation, we use four different path models and four different ways to compare these path models with the observed user trajectory. The path models will be briefly explained in the next section.

2.2.1 Cirio et al

Cirio et al. [4] observed that the angle α between the target's orientation and the line connecting the target and the user decreases linearly with a user's distance to a target $\|I\|$. Moreover, they observed that a users

walking velocity is inversely proportional to his turning speed. In their model, they thus propose equation (1).

$$\frac{\|I\|}{\dot{\|I\|}} = \frac{\alpha}{\dot{\alpha}} \quad (1)$$

Using Euler integration, they update α , and together with limits on the human walking and turning speed, they obtain an updated position. The position is updated in this way until the target is reached.

2.2.2 Arechavaleta et al

The second model was introduced by Arechavaleta et al. [2]. It is based on the fact that human paths like many other movements adhere to certain optimality criteria. They use equation (2) for the system's dynamics and cost function (3) that has to be minimized by the path to a target.

$$\begin{pmatrix} \dot{x}_T \\ \dot{y}_T \\ \dot{\varphi} \\ \dot{\kappa} \end{pmatrix} = \begin{pmatrix} \cos \varphi \\ \sin \varphi \\ \kappa_T \\ 0 \end{pmatrix} u_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} u_2 \quad (2)$$

$$J = \frac{1}{2} \int_0^T < u(\tau), u(\tau) > d\tau \quad (3)$$

In this equation, x and y describe the position of the torso's middle point between the left and the right shoulder, while φ is the torso's orientation. Since they also take the path's curvature into account, u_1 is the linear velocity and u_2 is the control of the time derivative of the curvature.

2.2.3 Fink et al

Fink et al. [5] use a steering model based on (4), where φ is the user's heading, ψ_g the goal's orientation, d_g the distance to the goal, and $b = 3.25$, $k_g = 7.50$, $c_1 = 0.40$ and $c_2 = 0.04$.

$$\ddot{\varphi} = -b\dot{\varphi} - k_g(\varphi - \psi_g)(e^{-c_1 d_g} + c_2) \quad (4)$$

Like in the model by Cirio et al., the position and the orientation is continuously updated until the target is reached.

2.2.4 Graph

The planning of redirection is time-critical and can already take a long time, if the number of available actions is high [11]. The generation of a model path will cost additional time, especially if a prediction over multiple time steps is involved. A very simple though not realistic model was included for comparison, in which the walking trajectory is represented by linear path segments. Although this model does not realistically represent human locomotion, it is the simplest possible approach. If such a simple model is already sufficient for the prediction of a users intended walking target, there is no need for more complicated models. This segmented linear representation of path can be found in robotics literature for more complex environments. However, our test environment is simple and free of obstacles. Thus we simply connect start and target point with a single straight line.

The path models mentioned above allow creating a path given a start and a target location, and while the start position is the user's starting position, the location for the targets is often not that clear. Anything a person could want to walk to can potentially be a target. This includes objects that the user can interact with, images and locations that offer a nice view, but also key locations of the environment's spatial configuration such as doors or hallways. In this paper, we assume that the locations of the targets are known. In a real application, this has to be either tagged by hand or considered when building the environment, or recognized automatically in the environment for example based on a floor plan.

3 Comparing Paths

The path models presented in the previous chapter allow generating a possible path of a human from a given start position to a certain target. Assuming an accurate model, the path the user actually travels to the chosen target should be similar to the one the model generated initially. If a suitable measure for the similarity would exist that can be used while the user is still walking, it would be possible to decide to which model the observed path is most likely belonging to and thus which is the most likely target. In the following chapter, three such similarity measures and a comparison method will be presented.

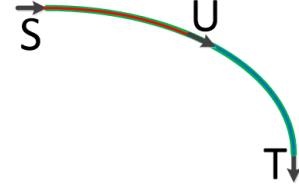


Fig. 2 Selecting a position u on the optimal path from S to T (green) and calculating the solutions for S to U (red) and U to T (blue) results in the same overall path because of Bellman's Principle of Optimality

3.1 Cost Function

Previous research has shown that many human movements follow certain optimality criteria [1, 9]. They use a cost function J that is minimized by the trajectory a human moves on. Based on this, a novel comparison method is proposed here that uses such a cost function to compare paths based on their cost by estimating the amount of movement wasted by deviating from the optimal path.

The exact definition of J varies between authors. In the case of Mombaur et al. for example the function (5) is used, where a is the acceleration, ψ is the angle between the current facing direction and the direction to the target, and T the overall time. But they all optimize towards a short path while at the same time minimizing the changes in forward acceleration and curvature. This corresponds to a smooth path without quick changes in direction and no unnecessary changes in velocity.

$$J(\mathcal{P}) = T + 1.2 \int_0^T \dot{a}_{forw}^2 dt + 1.7 \int_0^T \dot{a}_{rot}^2 dt + 5.2 \int_0^T \psi^2 dt \quad (5)$$

Bellman's Principle of Optimality is a necessary condition for optimality. It states that any part of a solution of an optimization problem itself is the optimal solution of the associated sub-problem. In the context of human paths, this means that any optimal path can be split and the solution for the two parts is the same as for the combined case (Figure 2). In this case, equation (6) follows in which $\mathcal{P}_M$ is a path generated by a model, J is a cost function associated with human path planning, s and t are the path's start and end points, and $u \in \mathcal{P}_M(s, t)$ is any point on the original path.

$$J(\mathcal{P}_M(s, t)) = J(\mathcal{P}_M(s, u)) + J(\mathcal{P}_M(u, t)) \quad (6)$$

Since $\mathcal{P}_M(s, t)$ is optimal, (7) will hold for $\forall p \in \mathbb{R}^2$. This means that every path diverging from the optimal

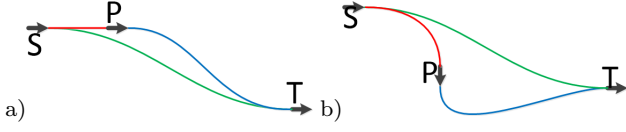


Fig. 3 Model path $\mathcal{P}_M(S, T)$ from start S to target T (green), path $\mathcal{P}_{real}(S, P)$ traveled by user P (red) and new model path $\mathcal{P}_M(P, T)$ from P to T (blue). In case a), the deviations are small and J_{loss} will also be small. In case b), the observed path deviates far from the model and J_{loss} will be large.

model will have higher cost, even if the user follows the optimal path from now on.

$$J(\mathcal{P}_M(s, t)) \leq J(\mathcal{P}_M(s, p)) + J(\mathcal{P}_M(p, t)) \quad (7)$$

We use this to define J_{loss} as in (8), where $\mathcal{P}_{real}$ is the user's recorded path and p is his or her current position. J_{loss} is zero, if and only if $\mathcal{P}_M$ is a perfect model, since J has to be nonnegative to be a valid cost function. This means that J_{loss} is a measure for the user's deviation from the optimal path and that the increase in path cost is measured through the cost function J . Since J is a cost function validated with experimental data, the increase in cost is correctly weighted between changes in length and curvature. In a more informal way, J_{loss} can be thought of as a measure for "wasted" movement on a path towards a certain target. It will for example increase only a little if the user deviates from the path (Figure 3a), however it would increase rapidly if the user turns away from or moves past the target (Figure 3b).

$$J_{loss} = J(\mathcal{P}_{real}(s, p)) + J(\mathcal{P}_M(p, t)) - J(\mathcal{P}_M(s, t)) \quad (8)$$

3.2 Dynamic Time Warping

Dynamic Time Warping is a method for comparing sequences of temporal data originally published by Sakoe and Chiba [16]. It is widely used in classifying time varying patterns such as speech, video or movement data.

It uses a distance measure $d(s, t)$ between the symbols s and t and then, given two sequences S and T of length M resp. N , finds a sequence of indices i, j of length L such that $\sum_{k=1}^L d(S(i(k)), T(j(k)))$ is minimal. This is usually done using a dynamic programming algorithm and there can be restrictions on $i(k), j(k)$ with respect to $i(k-1), j(k-1)$ depending on the application.

Dynamic Time Warping has been used successfully to deal with varying talking speeds and gesture recognition and thus it should also be well suited for dealing with speed differences between walkers and models.

To adapt this to human locomotion paths, the sequences S and T are sequences of points in $\mathbb{R}^2$. Therefore, d is chosen to be the euclidean L2 norm. Furthermore, all points in the sequences have to be connected and only forward steps are allowed, therefore $0 \leq i(k) - i(k-1) \leq 1$ and $0 \leq j(k) - j(k-1) \leq 1$. The overall distance from the comparison is then divided by the length of the path to compensate for varying path lengths.

Since the model covers the whole path from the starting location to the end, and the recorded path is only partially complete, they cannot be compared directly. Instead, only the beginning of the model path is used, such that it has the same length as the recorded path.

3.3 Minimal Distance

The Minimal Distance predictor finds the closest point in the reference path T for each point of the test path S and sums up all distances. The sum is divided by the path length to prevent $MD(S, T)$ from being dependent on the length of the path.

$$MD(S, T) = \sum_{i=1}^M \min_{t \in T} (|S(i), t|) / M \quad (9)$$

3.4 Double Exponential Smoothed Direction

Nescher et al. [12] originally proposed this approach to estimate an intended target. It is based on the idea to use a user's facing or movement direction to infer the intended target. However, due to gait-induced oscillations this direction varies over the course of the step which disturbs the prediction. To solve this, Nescher et al. proposed to use a double exponential smoother to reduce these oscillations while keeping the latency lower than when using a moving average filter of comparable performance. Equations (10) and (11) from [12] describe the smoothing of the movement direction $\vec{\omega}$ with $\alpha = 0.004$, $\beta = 0.004$ and the output $\vec{s}$. Using a path model, s can be compared to an expected angle at this position along the path. This allows to use this method in situations where one target is straight ahead in the beginning while another is towards the side.

$$\vec{s}_t = \alpha \vec{\omega}_t + (1 - \alpha)(\vec{s}_{t-1} + \vec{b}_{t-1}) \quad (10)$$

$$\vec{b}_t = \beta(\vec{s}_t - \vec{s}_{t-1}) + (1 - \beta) \vec{b}_{t-1} \quad (11)$$

Any direction-based approach can use either direction of movement or facing direction, since sensors for gaze direction or torso orientation are typically not available. Although gaze has been used as an indicator for a person's intention in other fields, a user can also look at something without the intention to walk there. While the movement direction might have advantages, its robustness depends on the walking speed and it becomes useless once the user stands still. In future work, a speed dependent blending between walking and facing direction might be interesting, but this is outside of the scope of this paper.

In order to estimate the target, the angular deviation between the smoothed movement direction $\vec{s}$ and the direct connections between the user's position and the targets are compared.

4 Development Over Time

All prediction methods described in the previous chapter take the user's past path into account. While this allows to accumulate data to make a better prediction, it can also reduce the responsiveness of the prediction. Consider a user who walks towards one target, and halfway he or she decides to go to another target instead. In this case, the accumulated distance from the first half will bias the prediction towards the original target, even though a new prediction started at this point would clearly predict the new target.

There are two ways to approach this problem. The first approach is to detect the user's change of mind, for example from a sudden turn or when he stops and starts walking again. However, this approach is based on a defined rule set and if none of these rules are triggered, the prediction is not reset. The second approach is to simply add a decay factor to the distance. This weights the distance accumulated on the last meter more than the distances from the rest of the path. There are two parameters that can be tuned for an exponential decay: the decay rate and a final cutoff, after which no samples are included. This also allows to discard samples with a very high distance values and also limits the number of points to be compared.

This decay can either be weighted with time or distance, and both designs have certain advantages. For walking behavior they are linked by the walking speed, but if the user stops walking, the behavior changes. If one uses the time for the decay, the prediction will include an increasingly shorter path until only the current

standing position is included (superimposed with the user's head movements). This will cause the prediction to go towards a uniform distribution and when the user starts walking again, the predictor is essentially reinitialized. If the path length is used, the prediction will stay as it was when the user stopped and only once he continues will it go towards the new target.

Equation (12) shows the resulting distance for a path with N points. D is a $N \times 1$ distance vector calculated with another distance measure, $\alpha = [0, \infty)$ is the decay factor, d_i is the distance along the path between the C^{th} and N^{th} sample in either time or space and C is the index of the last sample that is still within the cutoff distance.

$$D_{decay} = \sum_{i=C}^N D(i) \cdot e^{-\alpha \cdot \Delta d_i} \quad (12)$$

5 Making Decisions

The estimator gives one scalar value d_i per target i for every position sample provided by the tracking system, independent of model and comparison method. On its own, this vector is not useful for planning; instead, either a binary decision or a probability distribution over all targets is needed. For n given targets, there are two ways of approaching this problem. In the first approach, every target can be analyzed on its own and it can be decided if it is still a possible target or not. This could be done for example based on experimental data describing how much humans typically deviate from the optimal path. The alternative approach is to view it as a classification problem. In this case, the whole n -dimensional vector is used and it is either assigned to one of the targets or the probability of each individual target is estimated.

In this paper, we limit ourselves to two targets and therefore we will present a decision scheme in two dimensions. The easiest way to do this would be to simply assume the target with the smallest distance to be the correct one. However, there are some problems with this approach. First, even a small error could lead to a wrong decision; second, slight changes in the path, for example caused by gait, can cause the estimation to alternate between the two targets which in turn would have an adverse effect on the planning algorithm's performance. However, any Model Predictive Control approach like the one used in [11] can automatically deal with different future paths with certain probabilities. Therefore, the distance vector needs to be mapped to a probability distribution. To do so, a sigmoid function (13) is used

to map the difference in distance between the two targets to the range $[0, 1]$ for target 1. For target 2, 14 is used. The sigmoid function should be 0.5 for $d_1 = d_2$, but the slope can be set dependent on the comparison method by setting c appropriately.

However, this parameter requires manual tuning and depends on the layout of the environment. Therefore, we propose to use the same models and distance functions to generate a set of paths from a random position close to the expected starting position to a position near the targets. We then apply the chosen predictor to these paths to obtain the distance values. Based on these values we can then determine c (or train for example a support vector machine or any classical machine learning approach). This allows to completely tune the decision function without any human generated data.

$$P_1 = \frac{1}{1 + e^{-c(d_2 - d_1)}} \quad (13)$$

$$P_2 = \frac{1}{1 + e^{-c(d_1 - d_2)}} = 1 - P_1 \quad (14)$$

6 Experiment

In order to test the proposed prediction method and compare the models and methods, an experiment was designed. The goal was to offer participants two targets and have them walk to one of them. They were shown an empty room with two exits and were instructed to select one of them and then walk towards it. They were deliberately not given any additional task such as a search or fetch task to avoid situations where they change their target during locomotion. Although this does not represent the most realistic use case, it provides clear and uncontaminated data for an initial evaluation. At the same time, it was decided to leave the choice of the target to the participants.

After they exited the room through one of the exits 1 - 4 (see Figure 4) and reached the end of the respective corridor, they were instructed to return to their starting position S. Due to the size of the available tracking space and the decision not to use redirection in the user study, it was not possible to connect the rooms in any other way while keeping the target positions. As soon as they returned to the start position and before they could turn around, the virtual world changed behind them to the next layout. Four different locations for the exits were tested resulting in a total of six combinations. Figure 4 shows the location of all targets in the room, the conditions are listed in table 1. The targets were deliberately not shown as simple points or gates and the

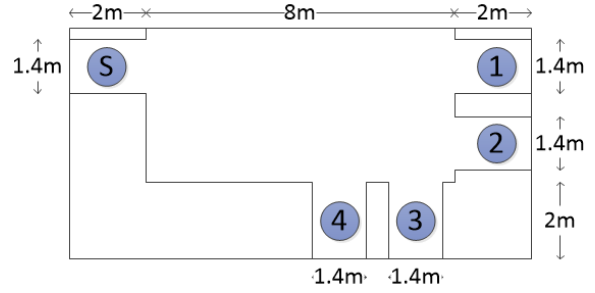


Fig. 4 Layout of the virtual environment with start location S and targets 1 through 4. Only two exits were visible at any given time.

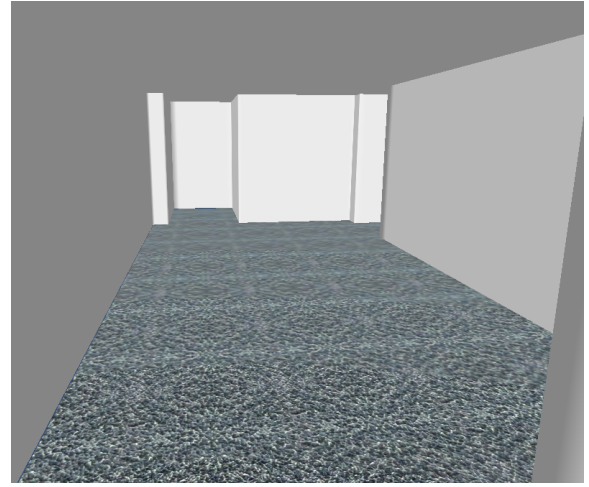


Fig. 5 The image shows a view from the starting location towards the two exits in condition 2.

Condition	Targets
1	1 & 2
2	1 & 3
3	1 & 4
4	2 & 3
5	2 & 4
6	3 & 4

Table 1 Conditions used in the experiment

corridors were long enough to encourage people to enter them as they would a real corridor. If the targets were depicted as points instead, people would not be forced to approach them in the angle we wanted them to. Some people would execute an initial turn and approach the target on a straight line in which case the prediction is much easier and can be done by extrapolating the movement direction instead.

The experiment was designed as a balanced block design with three repetitions, which gives a total of 18 paths per participant. The user study was conducted with our virtual reality setup, using an Intersense IS-1200 tracking system attached to an Oculus DK2 head-



Fig. 6 The VR setup used in the experiment with HMD, tracking system, and backpack

mounted display². Both are connected to an HP Elitebook 8560w running Windows 7. The virtual environments are created in Unity3D³. 18 people were recruited from the student body (13 male, 5 female). The average age was 23.7 years (standard deviation 2.3 years), average height was 1.77 meters (standard deviation 0.09). Nobody had prior experience with our setup.

Due to technical reasons or because some participants did not correctly complete the task, some trials had to be excluded from the later evaluation. In the user study, we found some trials in which participants changed the target half way through, but no trials were excluded for this reason. because in some of these cases it was unclear if he actually changed his mind or just did not walk according to the model, in which case he should of course not be excluded.

With these restrictions a total of 304 trajectories that were used in the evaluation. Although position and orientation information was recorded at a frequency of 180 Hz, the evaluation data was downsampled to reduce the amount of data to be analyzed. For the prediction, the first sample is used as a starting point and the target's position is used as the end point for the model path.

7 Results

There are two main metrics to compare the performance of the different combinations of models and estimation approaches presented previously. The first one is the percentage of correct estimations. However, it is possible that the probabilities for the two targets are very close together and when assuming the more likely one to be the correct answer, even a small error might lead to a wrong decision or an alternating result. To avoid this situation, samples can also be assigned to the "undecided" category, meaning they are assigned to neither of the two targets. This introduces the second metric, the percentage of samples that are classified. We only accept a decision if $P_i \geq 0.6$ for any target i . If none of the targets is above this value, the sample is classified as undecided. Unless otherwise noted $p < 0.05$ is used as a significance level.

As a first step, the sigmoid function defined in section 5, that is used for the decision, needs to be tuned. As expected, there is a trade-off between maximum correctness ($R_{CC} \rightarrow 1$) and a high number of classified samples ($R_{CT} \rightarrow 1$). It can also be seen that certain combinations perform strictly better than others. With the inclusion of all conditions, the predictors using the model by Fink et al. perform better than the ones with the model by Arechevaleta et al. Figure 11 shows the resulting ratio of correctly classified samples to the overall number of classified samples ($= R_{CC}$) as a function of the ratio of classified samples to the total number of recorded samples ($= R_{CT}$) for all the model/comparison combinations and conditions included in the evaluation.

For all further results, the parameters for the sigmoid functions are determined from model generated paths as described in section 5 with 10 model-generated paths for each target. Figure 7 shows training paths generated for condition 1 with the Fink/DTW predictor. For easier comparison, c is set such that 85% of the example path's samples are classified with $P \geq 0.6$. Tables 2 and 3 summarize R_{CC} for all model-predictor combinations in average over all conditions and for individual conditions, table 4 shows the standard deviation between users.

The following section highlights notable properties of different model-predictor combinations. Figure 8 shows all paths recorded for condition 2 using the model by Arechavaleta et al. together with the cost based estimator. The undecided classification occurs mainly at the beginning and in the zone where the two path-groups overlap. In Figure 9 an example of an estimator using DTW and Fink's locomotion model can be seen. Unlike the cost-based estimator, almost all samples in

² www.oculus.com

³ www.unity3d.com

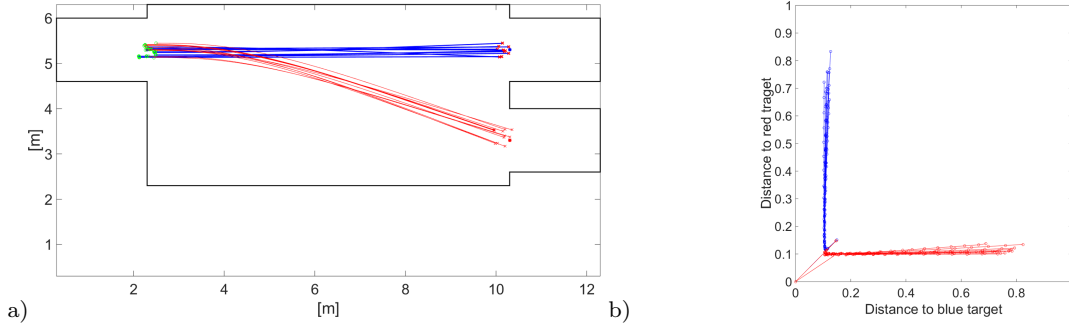


Fig. 7 a) Paths generated with the model by Fink et al. for condition 1. b) Cost of the paths using the DTW predictor.

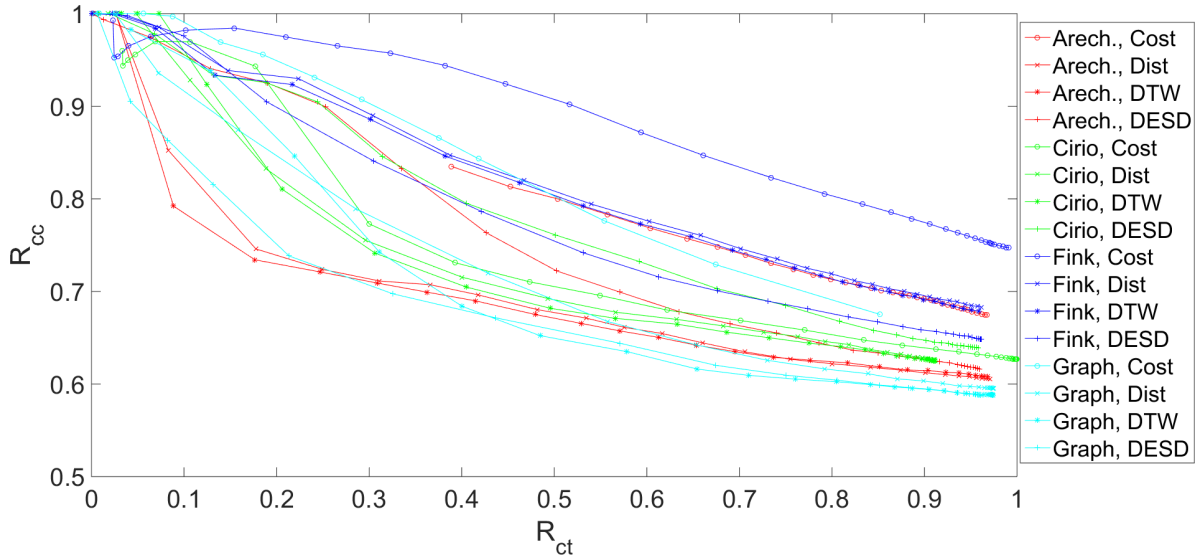


Fig. 11 The plot shows the R_{CT} and R_{CC} values for all tested model/classifier combinations using the sigmoid function defined in (13) with different values of $c = \sqrt{2}^n$, $n = [-10, 24]$.

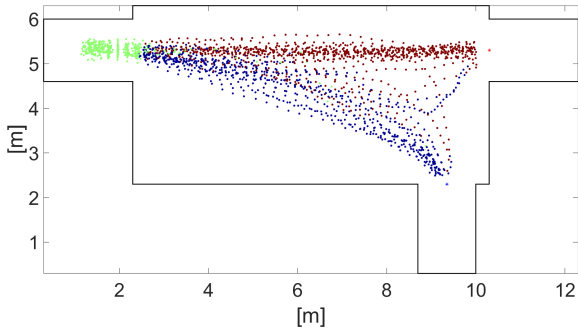


Fig. 8 Paths recorded for condition 2 for the combination Archavaleta/Cost. Red and blue points are assigned to the respective targets, green samples are undecided

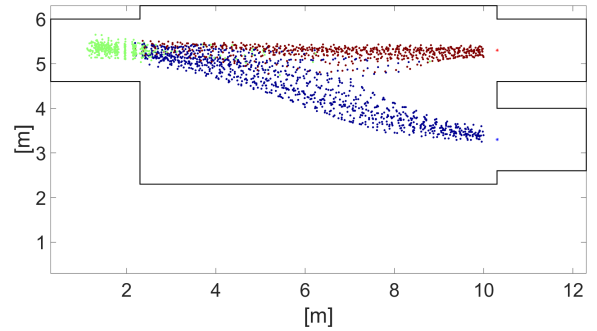


Fig. 9 Paths recorded for condition 1 for the combination Fink/DTW. Red and blue points are assigned to the respective targets, green samples are undecided

the beginning are classified as "undecided", but later all samples are assigned to a target either correctly or incorrectly. Figure 10 shows the prediction for condition 5. It can be seen that there is a long section where all the paths overlap, making a prediction impossible. However, this is caused by the fact that all predictors

were tuned for $R_{CT} = 0.85$ to facilitate an easier comparison, which forces an very early decision in this case.

Regarding the correct to classified ratio, the cost based estimators are significantly better than all other estimators.

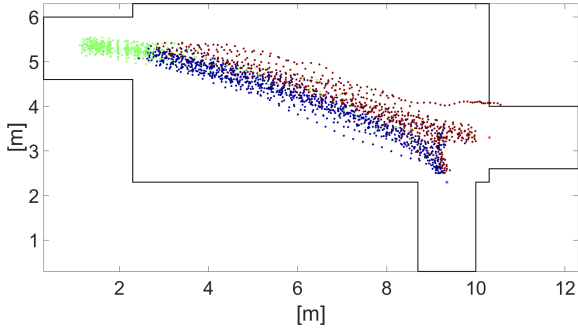


Fig. 10 Paths recorded for condition 5 for the combination Cirio/DTW. Red and blue points are assigned to the respective targets, green samples are undecided

	Arech.	Cirio	Fink	Graph
Cost	0.72	0.62	0.79	0.66
Dist	0.64	0.66	0.72	0.59
DTW	0.64	0.64	0.72	0.58
SD	0.65	0.67	0.67	0.62

Table 2 Number of correct samples over number of classified samples (R_{cc}) for all conditions

		Condition					
		1	2	3	4	5	6
Arech.	Cost	0.84	0.74	0.78	0.59	0.64	0.72
	Dist	0.84	0.62	0.82	0.42	0.57	0.54
	DTW	0.84	0.64	0.83	0.41	0.56	0.54
	SD	0.82	0.69	0.78	0.44	0.63	0.51
Cirio	Cost	0.54	0.66	0.47	0.72	0.69	0.66
	Dist	0.80	0.74	0.80	0.47	0.51	0.62
	DTW	0.81	0.74	0.83	0.46	0.50	0.51
	SD	0.83	0.74	0.76	0.47	0.62	0.57
Fink	Cost	0.93	0.83	0.85	0.66	0.75	0.69
	Dist	0.84	0.75	0.79	0.60	0.65	0.72
	DTW	0.84	0.74	0.78	0.60	0.63	0.72
	SD	0.82	0.78	0.70	0.47	0.66	0.58
Graph	Cost	0.89	0.80	0.65	0.46	0.64	0.54
	Dist	0.86	0.70	0.53	0.40	0.54	0.51
	DTW	0.85	0.66	0.52	0.40	0.54	0.51
	SD	0.82	0.74	0.65	0.47	0.61	0.42

Table 3 Number of correct samples over number of classified samples (R_{cc}) for individual conditions

	Arech.	Cirio	Fink	Graph
Cost	0.11	0.10	0.07	0.08
Dist	0.07	0.08	0.10	0.10
DTW	0.07	0.07	0.11	0.10
SD	0.05	0.07	0.09	0.07

Table 4 Standard deviation of the correct to classified ratio between users

Over all comparison methods, the model by Fink et al. performs significantly better than all other models, while the graph model is significantly worse than all others. There is no significant difference between Cirio’s and Arechavaleta’s models.

Condition	Target 1	Target 2	Target 3	Target 4
1	51.0%	49.0%	-	-
2	58.9%	-	41.1%	-
3	35.3%	-	-	64.7%
4	-	39.0%	61.0%	-
5	-	52.9%	-	47.1%
6	-	-	51.0%	49.0%

Table 5 Percentage of times targets were picked depending on the condition.

Target	Repeated in
1	19.0%
2	25.8%
3	45.7%
4	37.5%
Total	32.6%

Table 6 Percentage of times where a given target was repeated (only trials where this was possible were included)

7.1 Target statistics

The presented approach allows for the use of a prior probability if there is additional information about the typical behavior in this specific environment available. Since the participants had the choice of selecting their target, we also evaluated the frequency of choice for each target and condition. Table 5 shows the percentage of how often each target was selected relative to the total number of paths. Although it was not always possible to select the same target twice in two consecutive runs, 32.6% of times participants selected the same target again if this was possible. The results per target are shown in table 6.

7.2 Model performance

For the performance evaluation of the models, the same comparison metrics were used, but only the correct model path with the complete recorded path were compared. For DTW, distance and cost function, the model by Arechavaleta is significantly closer to the observed paths than the others. While there is no significant difference between Fink’s and Cirio’s model, the graph model is significantly worse. This shows that the additional complexity of the model actually increases the performance.

7.3 Development over time

Up to now, all measurements were presented without using the aforementioned decay factor. In the following section, we present the results for different α and cutoff parameters for the Fink & DTW predictor. This combination was chosen for its already good performance

α cutoff	1m	2m	3m	∞
0	0.71	0.72	0.72	0.74
0.23	0.72	0.72	0.73	0.74
0.35	0.72	0.72	0.73	0.74
0.69	0.72	0.72	0.73	0.74

Table 7 Average R_{cc} for selected combinations for α and the cutoff length for the Fink, DTW combination

and low runtime. The parameters are chosen in such a way that they either have a weight of 0.5 or a cutoff at 1, 2, or 3 meters or don't have any cutoff or decay. Table 7 shows the results for the respective parameters. There are no significant differences in the overall performance, but individual paths improved when the user did change his mind and switched targets. Figure 12a shows the 3rd path of user 4, figure 12b shows the prediction for all tested α and cutoff combinations. Here it can be seen that the responsiveness increases both with increasing α as well as decreasing cutoff length.

This example shows that there is no downside when limiting the number of points and adding an additional decay factor and it can therefore be used to deal with users changing their mind.

7.4 Performance over time

Since the walking task is performed over a certain time, the performance of the estimation will also change over time. However, the estimation should be available as early as possible to have more time for redirecting the user if necessary. Figure 14 shows the performance along the paths. For easy comparison, the paths were all segmented into 20 parts of equal length. The performance increases over time, both in the number of classified and the number of correct samples. The cost based estimators show a higher R_{CT} early on, but the increase is slower compared to the distance and DTW estimators, which start at zero but increase very quickly at around 10% progress.

7.5 User performance

An important factor of the performance is the general applicability of the model and comparison method. Figure 13 shows the performance per user for one of the model and estimation combinations. While there is a relatively large number of outliers. This is mainly caused by the condition that are hard to classify. Nevertheless, it should be noted that there are paths that are perfectly classified for every user. This means that even though there are some outliers that performed poorly,

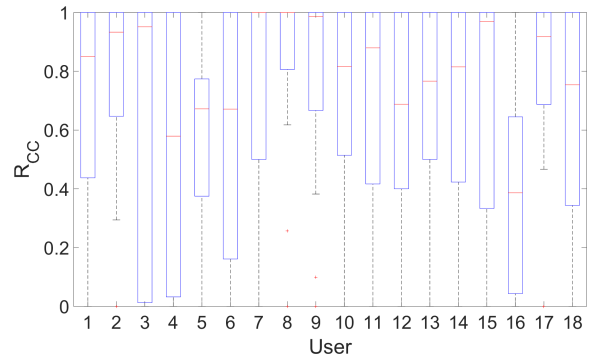


Fig. 13 Performance per user with the model by Fink et al. and the cost based estimation method for all conditions.

there were no user for which the system failed completely.

8 Discussion

In this paper, the experiments showed the feasibility of target prediction using human locomotion models. The more complex models like the one by Arechavaleta et al. performed significantly better than the simple connecting line, but they come with longer computation times. For reducing the computational effort, the models allow for different update steps size which is directly related to the run time of the model itself and also to the number of points in the final model and therefore the run time for the comparison. For the Dynamic Time Warping approach, the run time could be significantly reduced by limiting the number of points in the model and recorded path, as well as putting some limitations on the matching. Since the model by Arechavaleta et al. uses optimization, the run time is directly influenced by the abort criteria and the initial conditions.

Although the prediction worked for all the participants in our experiment, it is possible that for some people the model does not describe their behavior accurately. However, it is important to keep in mind that even a prediction that appears to be wrong considering the final target, might have been right at the time before the participant changed his mind. Because these cases can be hard to detect, these trials were not excluded from the evaluation, but for future experiments it might be worth considering to ask the participants to announce their target before they start to walk.

9 Conclusion

A novel approach for predicting a person's intended locomotion target was presented in this paper. The novel method of comparing an observed path to model paths

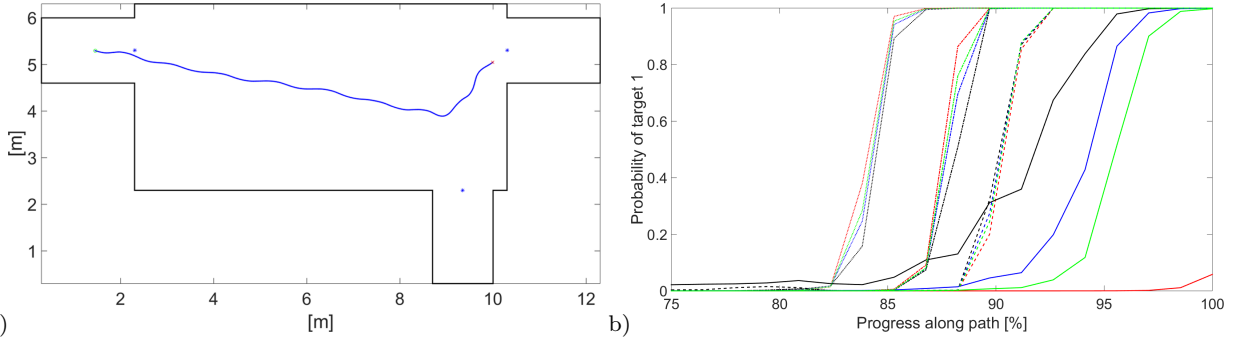


Fig. 12 a) 3rd path of user 4. b) Probability of target 1 for all parameters: Alpha = 0 (red), 0.23 (green), 0.35 (blue), 0.69 (black) and cutoff = 1 (dot), 2 (dot,dash), 3 (dash), ∞ (line)

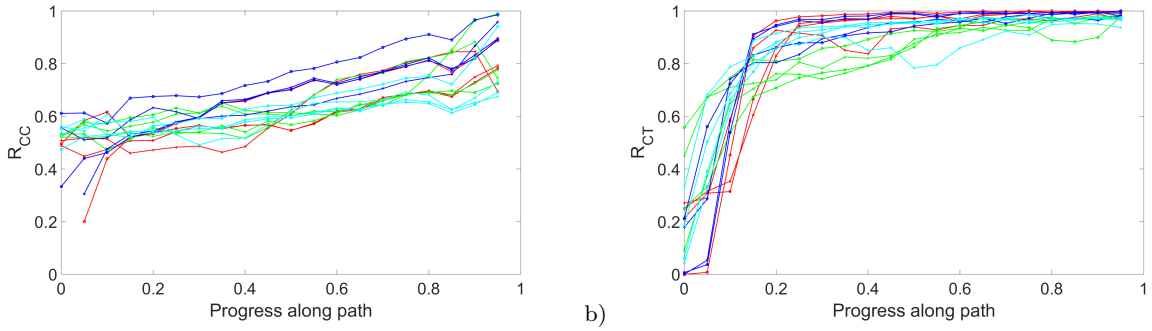


Fig. 14 a) Classified to total ratio over time. b) Correct to classified ratio over time. For the legend, please refer to Figure 11.

using a cost function has outperformed all comparison methods including both a direction based approach and the classic Dynamic Time Warping technique. At the same time, the model by Arechavaleta et al. outperformed all other models while the simple graph model was outperformed by the other models.

Figure 15 summarizes the proposed prediction strategy which consists of an offline and an online phase. In the offline phase the available targets in the environment are determined and the decision function (see chapter 5) is tuned. In the online phase the prediction is triggered by new tracking data and if the user is walking the observed path is compared to previously generated reference paths using one of the presented prediction methods (see chapter 3). Once the user arrives at a target new reference paths to the next set of targets are generated.

The proposed learning of parameters from model paths was tested and the performance did not change compared to the results previously presented in [22]. This is a further step away from using actual human path data to initialize basic parameters for the predictors. In the long run human data should only be used where absolutely necessary, for example to define prior probabilities containing human behavior. In the future,

more advanced methods to map the distance vector to the probability should be used. The concept needs to be extended to allow more than two targets and for a version used in an application, it should not be tuned purely based on R_{CT} . Instead there should be a bound on the number of misclassifications and R_{CT} should be free.

Since we observed some differences between users, another approach could be to adapt the classifier online to the current user. In this case the initial parameters would be model generated and while the user is walking, they adapt to his or her particular walking pattern.

The last thing that has to be added by hand is the location of the targets. For future applications it would be advantageous to automatically recognize target points in the environment. Furthermore, the experiment presented in this paper had a very simple layout. In reality, a user can not only walk from one starting location to a goal. Instead, it is possible that he continues to a next target once he reached a waypoint, or he stops halfway to look around for example. In more complex environments, the changing visibility will have an influence on paths. Moreover, in future work both static and mobile obstacles will also be included in the path modeling.

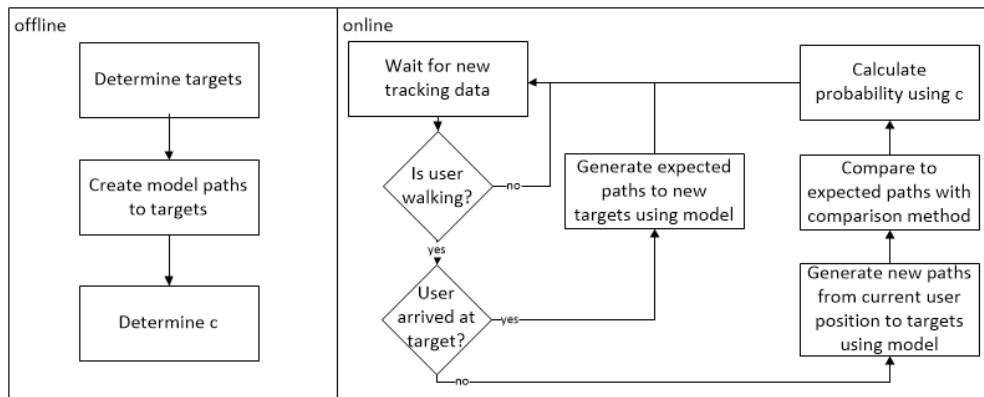


Fig. 15 Offline and online components of the proposed prediction method.

In order to allow for an earlier initial estimate of the walking target, eye tracking could be a suitable supplement to the method presented in this paper. However, to justify the additional complexity and increased cost, the performance improvement needs to be significant.

Acknowledgements The authors would like to thank the Swiss National Science Foundation (project number 205121 153243) for funding this work.

References

1. Arechavaleta, G., Laumond, J.P., Hicheur, H., Berthoz, A.: The nonholonomic nature of human locomotion: a modeling study. In: *Biomedical Robotics and Biomechanics*, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Conference on, pp. 158–163. IEEE (2006)
2. Arechavaleta, G., Laumond, J.P., Hicheur, H., Berthoz, A.: An optimality principle governing human walking. *IEEE Transactions on Robotics* **24**(1), 5–14 (2008)
3. Bruce, A., Gordon, G.: Better motion prediction for people-tracking. In: *Proc. of the Int. Conf. on Robotics & Automation (ICRA)*, Barcelona, Spain (2004)
4. Cirio, G., Olivier, A.H., Marchal, M., Pettre, J.: Kinematic evaluation of virtual walking trajectories. *IEEE Transactions on Visualization and Computer Graphics* **19**(4), 671–680 (2013)
5. Fink, P.W., Foo, P.S., Warren, W.H.: Obstacle avoidance during walking in real and virtual environments. *ACM Transactions on Applied Perception* **4**(1), 2 (2007)
6. Hicheur, H., Pham, Q.C., Arechavaleta, G., Laumond, J.P., Berthoz, A.: The formation of trajectories during goal-oriented locomotion in humans. i. a stereotyped behaviour. *European Journal of Neuroscience* **26**(8), 2376–2390 (2007)
7. Interrante, V., Ries, B., Anderson, L.: Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In: *3D User Interfaces, 2007. 3DUI'07. IEEE Symposium on*. IEEE (2007)
8. Larrue, F., Sauzeon, H., Wallet, G., Foloppe, D., Cazalès, J.R., Gross, C., N'Kaoua, B.: Influence of body-centered information on the transfer of spatial learning from a virtual to a real environment. *Journal of Cognitive Psychology* **26**(8), 906–918 (2014)
9. Mombaur, K., Truong, A., Laumond, J.P.: From human to humanoid locomotion inverse optimal control approach. *Autonomous Robots* **28**(3), 369–383 (2010)
10. Nabiyouni, M., Saktheeswaran, A., Bowman, D.A., Karanth, A.: Comparing the performance of natural, semi-natural, and non-natural locomotion techniques in virtual reality. *Symposium on 3D User Interfaces*, IEEE (2015)
11. Nescher, T., Huang, Y.Y., Kunz, A.: Planning redirection techniques for optimal free walking experience using model predictive control. *Symposium on 3D User Interfaces*, IEEE (2014)
12. Nescher, T., Kunz, A.: Analysis of short term path prediction of human locomotion for augmented and virtual reality applications. In: *Cyberworlds (CW)*, 2012 International Conference on, pp. 15–22. IEEE (2012)
13. Nitzsche, N., Hanebeck, U.D., Schmidt, G.: Motion compression for telepresent walking in large target environments. *Presence: Teleoperators and Virtual Environments* **13**(1), 44–60 (2004)
14. Razzaque, S., Kohn, Z., Whitton, M.C.: Redirected walking. In: *Proceedings of EUROGRAPHICS*, vol. 9, pp. 105–106. Citeseer (2001)
15. Ruddle, R.A., Volkova, E., Bühlhoff, H.H.: Walking improves your cognitive map in environments that are large-scale and large in extent. *TOCHI '11: ACM Transactions on Computer-Human Interaction* **18**(2), 10:1–10:20 (2011). DOI 10.1145/1970378.1970384
16. Sakoe, H., Chiba, S.: A dynamic programming approach to continuous speech recognition. In: *Proceedings of the seventh international congress on acoustics*, vol. 3, pp. 65–69 (1971)
17. Steinicke, F., Bruder, G., Jerald, J., Frenz, H., Lappe, M.: Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics* **16**(1), 17–27 (2010)
18. Steinicke, F., Bruder, G., Kohli, L., Jerald, J., Hinrichs, K.: Taxonomy and implementation of redirection techniques for ubiquitous passive haptic feedback. In: *Cyberworlds, 2008 International Conference on*, pp. 217–223. IEEE (2008)
19. Su, J.: Motion compression for telepresence locomotion. *Presence: Teleoperators and Virtual Environments* **16**(4), 385–398 (2007). DOI <http://dx.doi.org/10.1162/pres.16.4.385>
20. Ushoh, M., Arthur, K., Whitton, M.C., Bastos, R., Steed, A., Slater, M., Brooks Jr, F.P.: Walking > walking-in-place > flying, in virtual environments. In: *Proceedings of*

- the 26th annual conference on Computer graphics and interactive techniques, pp. 359–364. ACM Press/Addison-Wesley Publishing Co. (1999)
21. Yen, H.C., Huang, H.P., Chung, S.Y.: Goal-directed pedestrian model for long-term motion prediction with application to robot motion planning. In: Advanced robotics and Its Social Impacts, 2008. ARSO 2008. IEEE Workshop on, pp. 1–6. IEEE (2008)
 22. Zank, M., Kunz, A.: Using locomotion models for estimating walking targets in immersive virtual environments. In: Cyberworlds (CW), 2015 International Conference on, pp. 229–236 (2015)
 23. Zmuda, M., Wonser, J.L., Bachmann, E.R., Hodgson, E., et al.: Optimizing constrained-environment redirected walking instructions using search techniques. Visualization and Computer Graphics, IEEE Transactions on **19**(11), 1872–1884 (2013)