Redirected Walking in Irregularly Shaped Physical Environments with Dynamic Obstacles

Haiwei Chen* Samantha Chen† Evan Suma Rosenberg‡

USC Institute for Creative Technologies, Los Angeles, CA

ABSTRACT

Redirected walking (RDW) is a virtual reality (VR) locomotion technique that enables the exploration of a large virtual environment (VE) within a small physical space via real walking. Thus far, the physical environment has generally been assumed to be rectangular, static, and free of obstacles. However, it is unlikely that real-world locations that may be used for VR fulfill these constraints. In addition, accounting for dynamic obstacles such as people helps increase user safety when the view of the physical world is occluded by a head-mounted display. In this work, we present the design and initial implementation of a RDW planning algorithm that can redirect the user in an irregularly shaped physical environment with dynamically moving obstacles. This represents an important step towards the use of RDW in more dynamic, real-world environments.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality

1 INTRODUCTION

Redirected walking (RDW) is a real-walking locomotion technique used in virtual reality (VR) that enables the users to explore a virtual environment (VE) that is larger than the available physical environment [3]. As the user moves, imperceptible adjustments such as rotations and translations are made to the virtual world in order to manipulate the user's path in the physical environment. The goal of RDW is to utilize these rotation and translation gains to guide the users to collision-free paths in the physical environment so that they may enjoy the benefits of real walking when exploring a larger VE.

Past research has proposed a number of different RDW strategies but has mostly focused on physical environments that are rectangular, static, and free of obstacles. For example, generalized steering algorithms, such as Steer-To-Center, redirect the user towards the center of the physical environment to keep the user away from its boundaries [3]. However, these techniques cannot be applied to irregularly shaped environments that have no obvious centers. There also exist planning-based approaches, such as MPCRed [2]. These algorithms utilize predictions of the user's virtual path and may redirect the users near physical boundaries if doing so will lead to desirable future states. While previous planning algorithms can theoretically be adapted to work in irregularly shaped physical environments with obstacles, such environments were not a focus in these works and were not present in their evaluations.

As VR tracking hardware becomes more accessible to the average user, real-world locations, such as living rooms, may be commonly used as the walkable space. These physical environments are unlikely to resemble the open, rectangular tracking space typically used in research settings. Additionally, they may contain dynamic obstacles such as nearby people. Rather than trying to fit the largest rectangular, obstacle-free space within such an environment, relaxing these constraints can potentially provide redirection algorithms with a larger working area and more opportunities for redirection. Redirection that takes moving obstacles into account will also increase the safety of not only the user but also other people.

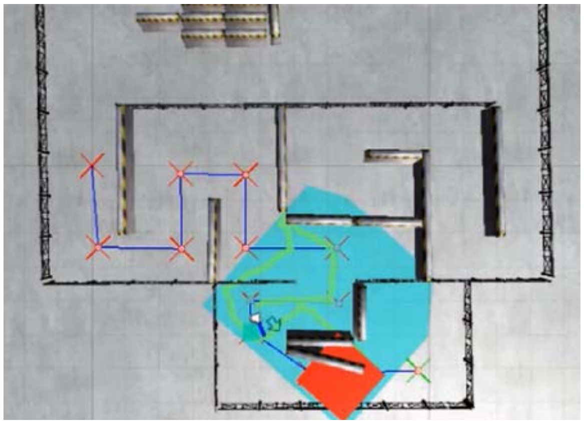

Figure 1: A visualization of the user's virtual path (blue) and physical (yellow) path in the walkable area (cyan) containing an obstacle (red).

In this work, we present a novel planning algorithm that can redirect the user to avoid collisions in an irregularly shaped physical environment with dynamically moving obstacles. The algorithm makes no assumptions about the physical environment and handles environment updates by replanning redirection gains online. This allows RDW to be used in a variety of real-world spaces.

2 ALGORITHM

Planning: Algorithm 1 shows a RDW planner based on the concept of optimal control. In a depth-first-search manner, it recursively applies redirection techniques (RETs) to a singular virtual path, consisting of walk and turn-in-place segments, to find an optimized sequence of RETs. This sequence transforms the virtual path such that it collides least with the physical boundaries (including both obstacles and walls). The types of RETs are selected based on the segment type: rotational gains are applied to turn-in-places, and translational and curvature gains are applied to walks. Reset is a special type of RET that requires the users to turn in place if other RETs cannot fully redirect them to avoid collisions with the physical boundaries. Resets can be applied implicitly in the ApplyGain function at line 5 in Algorithm 1. If the transformed paths intersect the physical boundaries, the remaining path is rotated using the selected reset angle. Resets, although effective, generally reduce immersion as the users are interrupted. Therefore, the algorithm minimizes the number of resets as the cost function.

---

*e-mail: chenh@ict.usc.edu
†e-mail: samchen@ict.usc.edu
‡e-mail: suma@ict.usc.edu

2018 IEEE Conference on Virtual Reality and 3D User Interfaces
18-22 March, Reutlingen, Germany
978-1-5386-3365-6/18/$31.00 ©2018 IEEE

```
Data: $X$: Current Configuration; $s$: Current Segment; $depth$:
      Current Depth; $E$: Physical Boundaries
Result: $bestcost$, $bestactions$
1  $Plan(X, s, depth)$
2      $bestcost = \infty$, $nextactions = [\,]$
3      for $gain$ in $Gains$ do
           /* Transform segment by gain                */
4          $cost, X_f = ApplyGain(s, gain, E)$
5          if $cost >= bestcost$ then
6              break
7          end
8          if $s.next != null$ or $depth < 2$ then
9              $nextactions, nextcost =$
                 $Plan(X_f, s.next, depth + 1)$
10             $cost = cost + nextcost$
11         end
12         if $cost < bestcost$ then
13             $bestcost = cost$
14             $bestactions = nextactions.push(gain)$
15         end
16     end
```

Algorithm 1: Plan redirection gains for a set of segments

Handling Dynamic Environments: The planning algorithm can be adapted to work in a dynamic environment, by testing each path against nearby physical boundaries at line 5 in Algorithm 1. To achieve online performance, the algorithm runs in a background thread and plans RETs for every three segments of the virtual path. The planning event is triggered under three conditions: 1) a certain distance before the user reaches every 3rd waypoint, 2) if RETs fail to perform a reset when the users are too close to a physical boundary, and 3) if changes are detected in the physical environment. To ensure safety, an external collision detector runs at every frame to monitor whether the users are too close to a physical boundary. If so, reset actions taken from the planner are triggered to stop the users and prompt them to turn in place.

Handling Misalignment: Planning takes $O(EN^3)$ time, where N is the number of possible gain values and E is the number of physical boundaries. During planning, RETs cannot be injected because the new plan has not yet been generated. Although planning mostly takes less than 1 second in our implementation, failure to inject RETs that should have been injected according to the plan leads to misalignment between the user's predicted and actual physical paths. To handle this, we design three heuristics: 1) if the misalignment causes the users to approach a physical boundary earlier than predicted, the external collision detector preemptively triggers a reset using the next planned reset direction; 2) the extra, unplanned user head rotation that arises from natural human movement is injected with rotational gains to make up for the previously unapplied gains; 3) because RETs applied during turns-in-place are generally more effective in redirecting the user than RETs applied during walk segments, our system avoids triggering planning during turns-in-place. These designs allow us to align predicted paths with actual paths in the physical space and ensure the safety of the user.

3 IMPLEMENTATION

The presented algorithm can be implemented on VR systems that support at least room-sized tracking. Locations of physical boundaries can be obtained by mapping, tracking, or a combination.

We implemented our algorithm using the open-source Redirected Walking Toolkit [1] and Acer Mixed Reality Headset[1]. Our implementation also uses reasonable gain values with respect to cybersickness [4]. For preliminary testing, we created a VE with a predefined singular path and prepared an irregularly shaped physical space of approximately 18x16 sq. ft. A movable cart of 1.5x2.25 sq. ft. was used as the dynamic obstacle (see Figure 2). In our initial implementation, the cart was not tracked; its position was updated manually. Figure 1 shows the path visualization of a user walk-through. The cart was moved to a different location when the user reached the 4th waypoint. Only a single reset was required, and the algorithm successfully prevented user collisions with the physical boundaries. In the future, we plan to conduct formal experiments.

Figure 2: The physical space (outlined with wood and white tape).

4 DISCUSSION AND CONCLUSION

We present an algorithm that can redirect the user in an irregularly shaped physical environment with dynamic obstacles, if they follow a singular virtual path. The algorithm is suitable for linear virtual experiences such as guided virtual tours. Our work can be extended to VEs with branching paths or widely open VEs, where path prediction is less feasible. Coupling the presented algorithm with a real-time environment mapping pipeline such as Simultaneous Localization and Mapping will also allow VR systems to support real walking in a more flexible way. While the view occlusion of VR headsets generally limit the walkable area to carefully prepared, obstacle-free environments, this limitation can be lifted if the user can be safely redirected away from physical objects mapped by the system.

ACKNOWLEDGMENTS

The work depicted here is sponsored by the U.S. Army Research Laboratory (ARL) under contract number W911NF-14-D-0005, the USC Annenberg Graduate Research Fellowship, and the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1418060. Statements and opinions expressed and content included do not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

REFERENCES

[1] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma. The redirected walking toolkit: a unified development platform for exploring large virtual environments. In *Everyday Virtual Reality (WEVR), 2016 IEEE 2nd Workshop on*, pp. 9–14. IEEE, 2016.

[2] T. Nescher, Y.-Y. Huang, and A. Kunz. Planning redirection techniques for optimal free walking experience using model predictive control. In *3D User Interfaces (3DUI), 2014 IEEE Symposium on*, pp. 111–118. IEEE, 2014.

[3] S. Razzaque. *Redirected walking*. University of North Carolina at Chapel Hill, 2005.

[4] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe. Analyses of human sensitivity to redirected walking. In *Proceedings of the 2008 ACM symposium on Virtual reality software and technology*, pp. 149–156. ACM, 2008.

---

[1] https://www.acer.com/ac/en/US/content/series/wmr