Leveraging Configuration Spaces and Navigation Functions for Redirected Walking

Jerald Thomas*

USC Institute for Creative Technologies

ABSTRACT

Redirected walking has been shown to be an effective technique for allowing natural locomotion in a virtual environment that is larger than the physical environment. In this position paper, I identify two large limitations of redirected walking and provide brief descriptions of solutions. I continue to introduce the conceptual design for an algorithm, inspired by techniques in the field of coordinated multi-robotics, that improves upon the current state of redirected walking by addressing these limitations. I then explain how it will become the foundation for my thesis and provide future research vectors.

Index Terms: Human-centered computing—Human computer interaction (HCI)—Interaction paradigms—Virtual reality; Human-centered computing—Human computer interaction (HCI)—HCI theory, concepts and models

1 INTRODUCTION

Redirected walking (RDW) is a method of navigating in immersive virtual environments that allows the user to use natural locomotion which has been shown to improve navigation [8] and the sense of immersion [10]. The method, introduced in 2001, employs three main techniques, known as gains [6]. These gains utilize the fact that our vision dominates our vestibular senses to manipulate the user into walking a virtual path that is decoupled from their physical path. Each of the three gains has a range of possible values that can be applied to the average user without them perceiving the manipulation [9]. It is assumed that we cannot always find a set of gains that maintain perceptual thresholds while keeping the user within the boundaries of the physical environment. When this happens, we perform a reset, essentially pausing the virtual experience and reorienting the user in a favorable way, usually towards the center of the physical environment. Ideally these resets are to be minimized.

For my thesis, I plan to address two limitations of the current state of RDW algorithms: the requirement that the physical environment is ideal and the lack of algorithm design specifically for multi-user experiences. In this paper, I introduce the conceptual design of Continuous Optimized COnfiguration Alignment (COCOA), a new algorithm specifically designed to address these limitations.

Limitation I. Currently all RDW algorithms rely on the three assumptions that the physical environment is clear of any obstacles, rectangular, and static. In nearly every environment outside of a controlled lab, these assumptions are restrictive and unrealistic for practical use. To solve this, RDW algorithms should be contextually aware of the physical environment and able to generate a solution that avoids obstacles while maximizing use of non-convex physical boundaries. The algorithm should also be continuous in order to accommodate for the dynamic nature of non-ideal environments, constantly updating its model of the physical environment instead of on a timed interval.

Limitation II. All RDW algorithms were designed with single user experiences in mind. While some work has been done to determine the feasibility of using single user algorithms for multi-user experiences, algorithms designed specifically for multi-user experiences would be more ideal [3, 4]. For instance, this would allow for deliberate user collision avoidance instead of simply detecting an impending collision and resolving it. Creating a contextually aware and continuous system is the first step towards designing RDW algorithms for multi-user experiences.

2 COCOA

COCOA uses two techniques from the field of robotics, namely configuration spaces and navigation functions, to calculate the RDW gains that will move the user in an optimal direction. The result is an that algorithm is computationally non-intensive, continuous, and can be contextually aware.

Configuration Spaces. The configuration space of a robot is a higher dimensional space where each dimension represents one degree of freedom of the robot [5]. For example, a circular robot such as the iRobot Roomba would have a three dimensional configuration space (an x dimension, a y dimension, and a theta dimension). Any possible configuration of the robot is represented in the configuration space as a single point, referred to as a configuration, and any obstacles can be represented as regions within the configuration space. This is desirable because given a starting configuration and a goal configuration any multitude of path planning algorithms can be used to generate a viable, collision free path through the configuration space.

---

*e-mail: jeraldlt@usc.edu

For RDW algorithms, a user's transform can be represented with three degrees of freedom (x, y, and θ). COCOA uses a configuration space for each user that is six dimensional, comprised of both the user's physical and virtual transforms. A configuration can also be constructed that represents a goal and consists of a physical and virtual target location.

Navigation Functions. A navigation function is formulated in a way such that if you always follow the negative gradient, you are guaranteed to reach a predefined goal. The formal definition is as follows: given a n-dimensional polytope P, a function $\phi : P \to [0, 1]$ is a navigation function if it is has a value of 1 at the facets of P, a value of 0 at a unique point in P, is twice differentiable on P, and is Morse. Navigation functions are used in the field of robotics to replace the path planning component of robot control [7]. When coupled with a gradient descent execution scheme, navigation functions present some very favorable qualities. They guarantee convergence at some target location, are resilient to external disturbances, and are computationally nonintensive. Furthermore, Ayanian et al. showed that a non-convex space can be broken into a set of convex polytopes, each with its own navigation function, and that a navigation function can by synthesized to bridge two adjacent polytopes [1].

Navigation functions are used in COCOA specifically for their robustness to perturbations. It is likely that the user will either not be able to move in the ideal direction through the configuration space (due to constraints on gains) or will simply choose not to. Unlike many robotic control techniques, navigation functions will handle this well and will simply continue to "nudge" the user in the most favorable direction. At a high level COCOA takes a current user configuration and a goal configuration and attempts to align them by calculating the optimal set of gains that will move the user towards the goal. Each frame, COCOA generates the configuration space by taking into account boundaries, obstacles, and other users. Then a navigation function is generated using a goal configuration. The algorithm determines the ideal direction (Ψ) to move in the configuration space using gradient descent on the navigation function. Given the change in the user's physical transform from the last frame the algorithm calculates the gains that will move the user through the configuration space in a direction as close as possible to Ψ.

3 THESIS FOCUS

For my thesis I plan to first focus on the development and optimization of COCOA. Once complete I plan to use COCOA to conduct research on some interesting questions that cannot be answered with current RDW algorithms. I have identified a small handful of research vectors in which a continuous and contextually aware capable algorithm, such as COCOA, could be applied.

General Algorithm. A general algorithm is a RDW algorithm that has no planning component. I plan to first compare COCOA to the common general algorithm, Steer to Center (S2C). To accomplish this, a goal configuration where the physical target is the center of the physical environment will be used. I then plan to explore some more sophisticated heuristics for picking the target physical position.

Planning Algorithm. COPPER, a pre-planned algorithm soon to be released as part of the Redirected Walking Toolkit [2], generates a list of optimal corresponding physical and virtual targets prior to the start of an experience. It then uses an online counter deviation algorithm to make sure that the user reaches the next physical and virtual target at the same time. I plan to replace COPPER's counter-deviation method with COCOA.

Dynamic Obstacle Avoidance. As physical obstacles and other users can be represented in the configuration space it is possible to use COCOA to actively redirect users around or away from obstacles or the other users.

The Handshake Problem. There will be times in a multi-user experience when the users will want to share the same spatial reference frame. As users progress through the experience the mappings between their physical and virtual transforms will diverge over time. Determining how we realign those mappings so that two users can be in close proximity to each other both physically and virtually is the "handshake problem" and cannot be solved elegantly with single user algorithms. It is possible to pick goal configurations and use COCOA to re-align multiple user's spatial reference frames.

REFERENCES

[1] N. Ayanian, V. Kallem, and V. Kumar. Synthesis of feedback controllers for multiple aerial robots with geometric constraints. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3126–3131. IEEE, 2011.

[2] M. Azmandian, T. Grechkin, M. Bolas, and E. Suma. The Redirected Walking Toolkit: A unified development and deployment platform for exploring large virtual environments. In *IEEE VR Workshop on Everyday VR*, 2016.

[3] M. Azmandian, T. Grechkin, and E. Suma Rosenberg. An evaluation of strategies for two user redirected walking in shared physical spaces. In *2017 IEEE VR*. IEEE, 2017.

[4] J. E. Holm. *Collision Prediction and Prevention in a Simultaneous Multi-User Immersive Virtual Environment*. PhD thesis, Miami University, 2012.

[5] T. Lozano-Perez. Spatial planning: A configuration space approach. *IEEE trans. on computers*, 100(2):108–120, 1983.

[6] S. Razzaque, Z. Kohn, and M. C. Whitton. Redirected walking. In *Proceedings of EUROGRAPHICS*, volume 9, pages 105–106. Citeseer, 2001.

[7] E. Rimon and D. E. Koditschek. Exact robot navigation using artificial potential functions. *IEEE Transactions on robotics and automation*, 8(5):501–518, 1992.

[8] R. A. Ruddle. The effect of translational and rotational body-based information on navigation. In *Human walking in virtual environments*, pages 99–112. Springer, 2013.

[9] F. Steinicke, G. Bruder, J. Jerald, H. Frenz, and M. Lappe. Estimation of detection thresholds for redirected walking techniques. *IEEE Transactions on Visualization and Computer Graphics*, 16(1):17–27, 2010.

[10] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks Jr. Walking¿ walking-in-place¿ flying, in virtual environments. In *Proceedings of the 26th annual ACM SIGGRAPH*, pages 359–364. ACM Press/Addison-Wesley Publishing Co., 1999.