ACMGVR: Architecturally Consistent Mazes for Games in Virtual Reality

Puru Ojha
International Institute of Information Technology
India
puru.ojha@research.iiit.ac.in

Y. Raghu Reddy
International Institute of Information Technology
India
raghu.reddy@iiit.ac.in

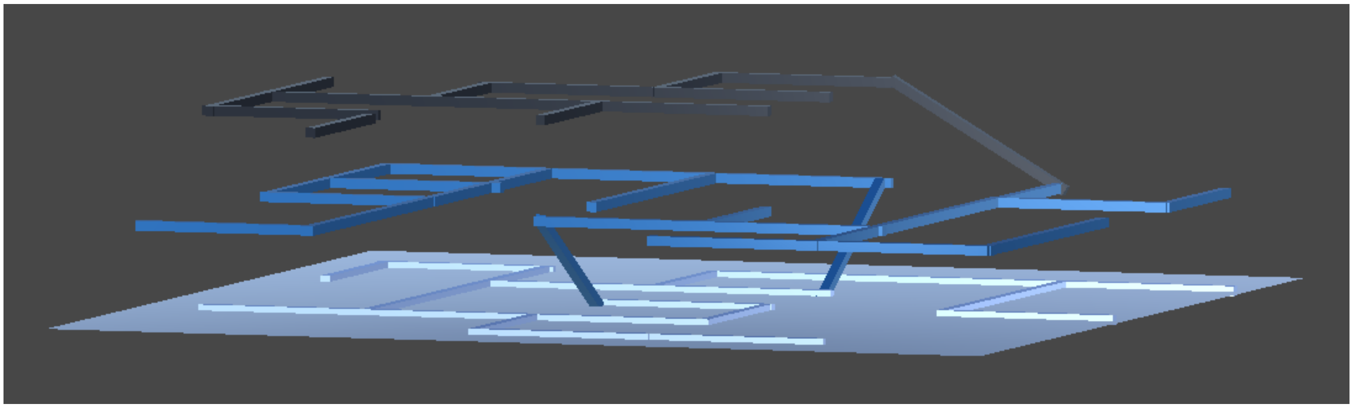


Figure 1: A Visualisation capturing birds eye view of 3D representation of maze.

Abstract

Walking is one of the most natural ways to explore an environment. While Virtual Reality (VR) facilities simulation of infinite spaces, its limited by the physical space available. To address this limitation, several techniques have been proposed, such as redirected walking, walking in place, and teleportation. While some of these techniques, such as teleportation and walking in place, disrupt the natural immersion, others like redirected walking limit the walkable area within the virtual space. The concept of overlapping architecture tackles this challenge by compressing large virtual environments into smaller spaces. This is achieved by generating portions of an environment that might be physically impossible as a whole. In this paper, we propose a system for creating maze-based virtual environments with multi-path capabilities. This system allows for infinite natural walking experiences while maintaining architectural consistency. Specifically, our system can generate distinct, customizable, and scalable virtual mazes for physical areas exceeding 4x4 meters. The combination of multi-path capabilities and architectural consistency can enhance the overall maze traversal experience for end users.

CHI PLAY Companion '24, October 14–17, 2024, Tampere, Finland

ACM ISBN 979-8-4007-0692-9/24/10
https://doi.org/10.1145/3665463.3678818

CCS Concepts

• **Human-centered computing** → **Virtual reality**; **User models**.

Keywords

Virtual Reality, Games, Mazes, Architecture

ACM Reference Format:
Puru Ojha and Y. Raghu Reddy. 2024. ACMGVR: Architecturally Consistent Mazes for Games in Virtual Reality. In *Companion Proceedings of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY Companion '24), October 14–17, 2024, Tampere, Finland.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3665463.3678818

1 INTRODUCTION

Locomotion, the act of moving from one location to another, is considered one of the most fundamental and universal activities performed during an interaction in Virtual Reality (VR) [8]. Traversing Virtual Environment (VE) larger than the physical space that the user is present in brings many challenges. Various techniques have been developed to address this issue. Locomotion techniques in VR broadly fall into two categories, natural locomotion and supernatural locomotion. In supernatural locomotion, techniques like controller-based locomotion, teleportation, walking in place [27], and world in miniature [24] are used to enable user movement within a VE. While natural locomotion feels more immersive, supernatural locomotion provides an opportunity to travel faster and further. Despite its gentle learning curve, supernatural locomotion is significantly affected by cybersickness. According to sensory conflict theory [14], this cybersickness arises from conflicting signals received by the body's proprioceptive, vestibular, and visual systems. Conversely, the primary limitation of natural locomotion is that it confines the extent of exploration to the boundaries of the available physical space, posing a significant constraint for practical applications.

Many natural locomotion techniques have emerged like redirected walking [11] which enhances the accessible area by using imperceptible gains, but these are still restrained by the limits of imperceptibility to change an individual's human perception. Some techniques change the VEs to accommodate traversal in large virtual areas. Bending the virtual environment whenever the user is near the physical boundary is one such way [22]. Designers have created virtual spaces that are physically impossible by exploiting the fact that only a specific portion of the Virtual Environment (VE) is visible to the user. However, this approach limits the use case to indoor virtual environments. Given their inherent indoor structure, mazes emerge as ideal candidates for leveraging this technique. In [12], the authors incorporated portals to seamlessly transport users between different pre-designed sections of the maze.

To the best of our knowledge, none of the works done till now provide a strategy that allows multi-directional traversal to a user, something that usually exists in a real maze. They provided only a single direction that the user walks on while generating the next segments. Previous works on mazes have also not focused on allowing a player backward traversal through the environment. In this work, we use a depth-first search through the available space to generate an appropriate maze that offers multi-directional options while maintaining architectural consistency throughout. It allows a more convincing perception of a real maze where users can switch directions, trace back their path, and walk uninterrupted. Here, we have developed a consistent maze that renders the same path that the user traveled through to allow them to choose a different option and explore a previously unexplored area of the maze.

In sum, the major contributions of our work are the following:

(1) a novel algorithm to design dynamic and infinite multi-path mazes for games in virtual reality.
(2) a mechanism to allow natural locomotion in both forward and backward directions in the maze.
(3) an approach to modify the mazes for different size of physical area.

The demonstrated imperceptibility to change of a human when engaged with a task [26] inspired our hypothesis that when traversing through the maze the consistency in local orientation within the maze architecture will have precedence over the overall architectural impossibility.

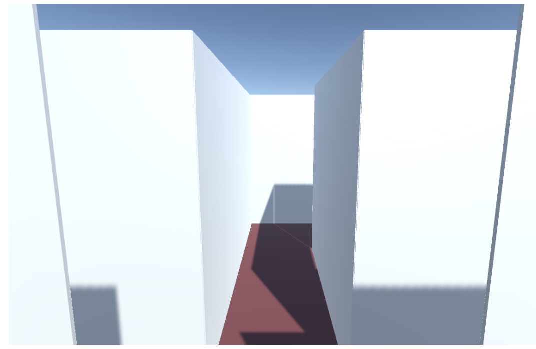
2 RELATED WORK

Many methodologies have been used to enable a user to experience virtual environments larger than the available physical space. Locomotion techniques similar to natural walking have been more effective in protecting the immersion of the user and reducing cybersickness [19].

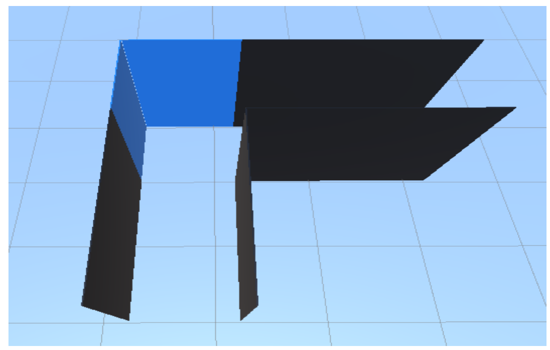
2.1 LOCOMOTION IN VR

Many methods have been designed to facilitate travel inside a VE. Teleportation is a common technique currently being used in many VR applications. Using teleportation users can travel through large distances by specifying their destination. After selecting the destination, the user's view is transferred to the target position with either a sudden change or a smooth viewpoint transition [13]. Gestures can also be used to emulate natural locomotion. Suma *et al.* [27] demonstrated the usability of walking-in-place technique on Microsoft Kinect. They used the depth sensors of Kinect to detect human actions and binded them to virtual mouse and keyboard commands. Movements like bobbing your head or Walking in Place can be detected and then used to generate equivalent motion in the Virtual Environment. With [20] Nilsson showed that less tiring gestures like swinging arms can also be used as an input. Works like [6] showed that Omnidirectional treadmills can be also used to keep the user stationary at a place while producing motion which is then transferred to the VE. However further research like Bowman *et al.* [8] showed that users are prone to losing their balance when side-stepping or turning on Omnidirectional treadmills, furthermore, the associated cost and the lack of portability is another reason behind the infrequent use of these treadmills.

Reorientation techniques are used to reset the user's direction when they are at risk of walking out of the physical space. Three reorientation techniques freeze-backup, freeze-turn, and 2:1 turn were introduced by Willimas *et al.* [32]. Peck *et al.* [21] compared reorientation techniques that had turns with/without audio/visual instructions. Through empirical experimentation, they discovered



(a)



(b)

Figure 2: (a) Player's perspective of the maze (b) Building blocks of the maze

that reorientation using visual distractors is more effective at increasing the user's sense of presence. Neth *et al.* [18] showed that event-based distractions are also effective in increasing the imperceptibility to change of the user. Reorientation during saccades[4] and eye movements [5] have been proposed manipulating the viewpoint between blinks and rapid eye movements.

Redirected walking has been used to manipulate the players movement and applying different gains between virtual and real movement to facilitate larger virtual areas. In [1] Azmandian showed that minimum 6X6 m area is required for these redirection techniques to work.

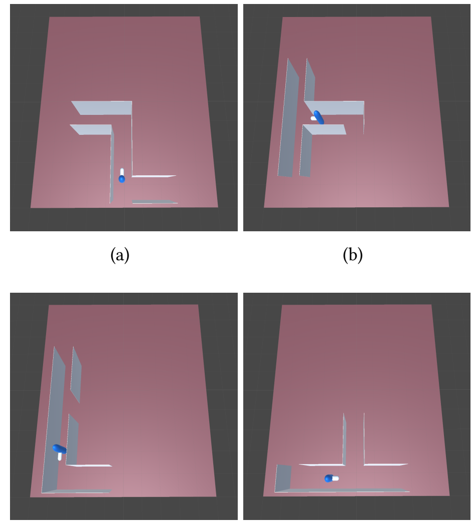
In another approach, virtual environments are designed in a way that they can accommodate natural movement while keeping the player inside the physical boundary. Many interesting techniques have been employed to manipulate the VR scenes. One way of doing this is to compress large Virtual Environments into smaller physical spaces by means of self-overlapping architecture [28]. In Flexible Spaces [10] they have done procedural generation of an overlapping architecture but only for a fixed area of size 9x9m.

2.2 REDIRECTED WALKING

With [11] Razzaque *et al.* introduced redirected walking locomotion techniques for virtual locomotion. They proposed applying imperceptible gains between user's real and virtual motion. Different types of gains like translational [31], rotational, curvature and bending gains are explored to allow viewpoint manipulation of the user. The limits of various gains such that they remain imperceptible have been analysed by empirical evaluation in the study done by Grechkin *et al.* [9]. Nilsson *et al.* in [19] divided redirected walking in to two major categories, (1) Manipulating the mapping between user's real and virtual movement, and (2) Manipulating the architectural properties of VEs to enable natural movement of the user.

Although unrestricted walking is desired in a VE but overt manipulation may also become necessary when the user is dangerously close to the boundary area. Various techniques like freezing the user and making them turn by Williams *et al.* [21], and using visual distractors to rotate the user by Peck *et al.* [21] , are proposed.

Steering algorithms have also been designed which apply various gains based on user's position and steer them away from boundary. Razzaque [11] prposed Steer to center, steer to orbit, and steer to multiple waypoint. Recently algorithms which predict user's future path to apply appropriate redirection gains have also been developed. Zmuda *et al.* proposed FORCE [33] which use subtle continuous redirection using key decision points. Nescher *et al.* proposed MPCRed [17] which predicts future travel path based on information of physical and virtual environment. In [25] strauss *et al.* used reinforcement learning to calculate appropriate gains for a virtual environment. Thomas [29] used artificial potential functions to constantly guide the user towards empty area of the physical environment. In [17] Thomas *et al.* designed an algorithm which uses a map of the VE to determine optimal redirection technique. Although most techniques have focused on redirection with respect to a single user, works like Azmandian [2] and bachman [3] have worked on multi user redirection techniques. Dong [7] used artificial potential function to enable multi user redirected walking.



(a) (b) (c) (d)

Figure 3: Procedurally Generated Overlapping Mazes

2.3 OVERLAPPING ARCHITECTURE

In [28] Suma *et al.* presented 'Impossible Spaces', a design mechanic to compress large virtual environments into small physical areas by overlapping parts of virtual environments such that the overlap is imperceptible. They found that detection threshold for a small virtual environment is an overlap of up to 56% and for large virtual environment is an overlap of up to 31%. Han et al. [10] proposed giving Horizontal, Vertical, and Accordian folds to the Virtual Environment for extended travel

Vasylevska and Kaufmann [30] developed an algorithm for procedural generation of a spatial layout when the content of the environment is more important than the overall architecture. In [26] 'Change Blindness' was used to change the position of the door between entering and exiting a room to reuse the available physical space. Space Bender [22] was designed by Simeone *et al.* which bends the virtual environment when the user is near the physical boundary. Despite producing great results, the overlapping architectural techniques can only be applied to interior spaces.

Many specialised VE that allows for infinite walking are also proposed. Environments like corridors and mazes suit this because of their interior structure. Mittal *et al.* [15] designed limitless path consisting of corridors. In [12] Koltai *et al.* developed a method that generates tile based overlapping mazes which use a portal to teleport user from one part of the maze to another seamlessly. They showed that the transition from one part to another remain unnoticeable if done properly [16].

With [23] Simons showed that when focusing on a main task the user ignores other unexpected changes in the environment.

3 METHODOLOGY

We wanted to create a more convincing strategy to generate virtual mazes that while keeping the core aspects of a maze intact, allows a user to traverse infinitely large mazes. The mazes generated have

i) Only one path joining any two positions ii) Multiple paths for the user to choose from iii) Unrestricted travel in both forward and backward directions to enable practical uses in various environments. The works done before while producing mazes in VR provided only two out of the above three qualities.

ACMGVR achieves this by first doing Depth First Search from the starting position to check for available physical space. Taking into account the typical configurations of generic mazes, we have made a deliberate design decision to ensure that every wall is positioned at a right angle to its adjacent walls. This approach guarantees a consistent and structured layout throughout the maze. The construction of the maze is based on two fundamental building blocks: section walls and connector walls. The section walls are responsible for forming the corridors within the maze, while the connector walls serve the purpose of linking these corridors together. Specifically, section walls consist of two parallel walls, each with a length denoted by l and they are separated by a width denoted by w. On the other hand, connector walls are characterized by their length w, and are utilized to connect various section walls. The strategic combination of section walls and connector walls results in the formation of the desired maze structure. Figure 2 illustrates the configuration where two section walls, depicted in black, are connected by two connector walls, depicted in blue. This visual representation highlights the structural relationship between the section walls and the connector walls, demonstrating how they integrate to form a cohesive part of the maze. Both variables, namely l and w, can be adjusted by the developers according to their specific design requirements. This flexibility allows for customization and optimization of the maze structure to meet various design objectives and constraints.

For checking the available directions for spawning the maze, we project rays of length ($l + 3/2 * w$) from the current player position in three directions: current direction of the player, 90 degree clockwise, and 90 degree anticlockwise from the current direction. Given that each wall is positioned at a right angle to the adjacent walls, these three directions are sufficient for accurately measuring the available space. For rays that do not intersect with the boundary, three additional rays are projected from their endpoints. This procedure is repeated three times to ensure that there is sufficient space for the generation of the maze. This iterative approach guarantees that the maze structure is adequately spacious and well-defined. This depth first search algorithm yields one, two, or three possible directions. The system chooses one or more directions randomly from the available options. The chosen directions are then used to create single or multi directional pathways for the player. To create the pathways, section walls are instantiated in all the selected directions, and required connector walls are utilized to stitch the connector walls together. This ensures a coherent and well-structured maze layout. Since every sections is either in the same direction as the current one or 90 degrees rotated from it, we do not need to store the exact coordinates of each section to store the architecture of the maze. By utilising this feature, we can store all the corresponding sections associated with the maze in the form of a graph, this helps us in keeping the storage requirements to minimum. Each node in the graph represents a section and each section is connected to children nodes which represent the following sections associated with the current section. Each node also has an index value stored in them. The index value is either +1, -1, or 0. The index value of a node tells us the rotation that the current node has with respect to its parent in multiples of 90 degrees.

When the player starts moving, the system generates the maze in the forward direction while removing the older portions from the scene. For every section, two sections in forward direction are rendered and one previous section connected to the current section is kept. This procedural generation and destruction help us in utilising the same physical space again. We also keep track of the direction that the player moves in and we save that in the form of index data for posterity. We also randomly close some pathways with horizontal connector walls to enhance the similarity to real mazes. One key aspect in which our system is different from the previous works is that it allows the player to go back to their previous positions and explore older sections or spawn completely new territories branching out from them. In the scenario when the user turns around, we use the saved information to generate an exact section of the maze that they walked through enabling them to explore older, previously unexplored sections of the maze. We do this by traversing through the graph that stores the information about the relative placement of the sections. If the user lands on a node with no children, the system does the same depth first search to measure the available space and appends new nodes to the current node. As the system is using the same physical area to render different parts of the maze, it effectively produces an overlapping architecture.

We can visualize the same by imagining the maze as a multi storey structure of mazes. A visualisation of one of the mazes is shown in Figure 1. The image provides a bird's-eye view of a maze, where older sections are retained, and overlapping areas are depicted as parts of the maze on different floors. Sections aligned vertically in the image represent different parts of the maze that occupy the same physical space. The connecting corridors between different floors illustrate instances when a section wall begins to overlap with older portions of the maze. Only three storeys are drawn here for the sake of brevity. The player's perspective of the maze is shown in Figure2 (a). In this instance, the maze has a height of 3 meters, while the player stands at 170 cm. Since the height of the maze does not affect our algorithm, it can be easily adjusted according to the developer's requirements. Figure 3 depicts a model player navigating through a procedurally generated maze. The player is represented as a blue cylinder, with a smaller white cylinder indicating the direction they are currently facing. Sub-figures (a) through (d) showcase various snapshots of the maze captured at different timestamps. The player starts from (a) and takes 4 left after crossing each section. On closer inspection of sub-figures (a) and (d) it can be seen that the user is walking within the same physical area of the room. However, from their perspective, they appear to be moving deeper into the maze.

We developed the prototype using Unity Engine and Oculus Quest 3, and designed it for a 6x6 meter area. However, in theory, the system should function with any Head-Mounted Display (HMD) and in spaces larger than a minimum of 4x4 meters. Both section walls and corridor walls are designed as prefabs in Unity. These prefabs can then be instantiated as per requirement to draw the whole maze. The core algorithm remains unaffected even if we adjust the length and width of section walls or connector walls to fit the available physical space. Each section wall is equipped with collision triggers that detect the player's movement within it. Since each trigger is associated with a section wall, we can also find the index of the corresponding section wall whenever player triggers a collision. We save the indexes of the triggers that the player hits in an integer queue. This helps us in navigating through the graph that stores the architecture of the maze while also keeping track of the player's position inside it. The triggers are strategically placed at the center of the wall sections, aligning the normal vector of the trigger with the direction of the section walls. When the player collides with a trigger, a vector is generated to store the direction of the collision. By calculating the angle between the collision vector and the normal vector of the trigger, we can ascertain the player's direction. If the angle between the trigger's normal vector and the collision vector is obtuse, it indicates that the player has turned around. This change in direction alters the traversal path within the graph. Instead of moving from the parent node to the child node, as in forward motion, the traversal moves from the child node to the parent node. In such scenarios, two sections of the maze are maintained in both forward and backward directions.

Task completion is intrinsic to any game, and task engagement directly impacts the cognitive abilities of the user. Based on this observation, we hypothesize that when users are occupied with a task, the relative local directions they choose become a more critical indicator of their cognitive map than the overall architectural feasibility. We designed this algorithm specifically to test the above hypothesis. The overlapping structure of the maze tackles the constraints of a limited physical space, while hiding in itself the impossible architecture of the overall structure. This impossible architecture of the maze can be used as a detection factor in the experiment to support the hypothesis. The information of the architecture of the maze is saved as a graph, this helps us in keeping computational complexity to a minimum while optimizing performance of our algorithm. It also enables the backtracking feature of our system, which can be considered necessary for task associated experimentation of our hypothesis. Moreover, the simplicity and modifiability of our system provides a perfect platform to generate distinct, customizable, and scalable virtual mazes for limited physical areas.

4 FUTURE WORK

As the system remains untested at this stage, experimentation is needed to assess the impact of overlapping architecture of mazes on player's cognitive map of the structure. We may enhance their cognitive mapping abilities and facilitate smoother traversal by altering the color gradient of the maze as the player progresses. Furthermore, adjusting the lengths of various maze sections randomly within gameplay can contribute to a more convincing and less predictable experience for players. These modifications will allow us to create diverse challenges within the maze while maintaining engagement. Comparing the performance of our algorithm with other baselines methods of maze generation is required for qualitative evaluation. Lastly, introducing a treasure hunt element provides an excellent opportunity to evaluate the playability of the maze game. By observing how players interact with hidden objectives or rewards, we can refine the design and enhance overall gameplay.

References

[1] Mahdi Azmandian, Timofey Grechkin, Mark T. Bolas, and Evan A. Suma. 2015. Physical Space Requirements for Redirected Walking: How Size and Shape Affect Performance. In *ICAT-EGVE*. https://api.semanticscholar.org/CorpusID:34314888

[2] Mahdi Azmandian, Timofey Grechkin, and Evan Suma Rosenberg. 2017. An evaluation of strategies for two-user redirected walking in shared physical spaces. In *2017 IEEE Virtual Reality (VR)*. 91–98. https://doi.org/10.1109/VR.2017.7892235

[3] Eric R. Bachmann, Jeanette Holm, Michael A. Zmuda, and Eric Hodgson. 2013. Collision prediction and prevention in a simultaneous two-user immersive virtual environment. In *2013 IEEE Virtual Reality (VR)*. 89–90. https://doi.org/10.1109/VR.2013.6549377

[4] Benjamin Bolte and Markus Lappe. 2015. Subliminal Reorientation and Repositioning in Immersive Virtual Environments using Saccadic Suppression. *IEEE Transactions on Visualization and Computer Graphics* 21, 4 (2015), 545–552. https://doi.org/10.1109/TVCG.2015.2391851

[5] Benjamin Bolte and Markus Lappe. 2015. Subliminal Reorientation and Repositioning in Immersive Virtual Environments using Saccadic Suppression. *IEEE Transactions on Visualization and Computer Graphics* 21 (04 2015), 1–1. https://doi.org/10.1109/TVCG.2015.2391851

[6] Rudolph Darken, William Cockayne, and David Carmein. 1997. The Omni-Directional Treadmill: A Locomotion Device for Virtual Worlds. *UIST (User Interface Software and Technology): Proceedings of the ACM Symposium*, 213–221. https://doi.org/10.1145/263407.263550

[7] Tianyang Dong, Xianwei Chen, Yifan Song, Wenyuan Ying, and Jing Fan. 2020. Dynamic Artificial Potential Fields for Multi-User Redirected Walking. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 146–154. https://doi.org/10.1109/VR46266.2020.00033

[8] Joseph J LaViola Doug A Bowman, Ernst Kruijff and Ivan Poupyrev. 2004. *3D User Interfaces: Theory and Practice* (2nd. ed.). Addison-Wesley Professional, New York, NY.

[9] Timofey Grechkin, Jerald Thomas, Mahdi Azmandian, Mark Bolas, and Evan Suma. 2016. Revisiting detection thresholds for redirected walking: combining translation and curvature gains. 113–120. https://doi.org/10.1145/2931002.2931018

[10] Jihae Han, Andrew Vande Moere, and Adalberto L. Simeone. 2022. Foldable Spaces: An Overt Redirection Approach for Natural Walking in Virtual Reality. In *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 167–175. https://doi.org/10.1109/VR51125.2022.00035

[11] Eurographics Jonathan, C. Roberts, Short Presentation, Sharif Razzaque, Zachariah Kohn, and Mary Whitton. 2001. Redirected Walking. *Proceedings of Eurographics* (08 2001).

[12] Balázs Gyula Koltai, Jakob Elkjær Husted, Ronny Vangsted, Thomas Noes Mikkelsen, and Martin Kraus. 2020. Procedurally Generated Self Overlapping Mazes in Virtual Reality. In *Interactivity, Game Creation, Design, Learning, and Innovation*, Anthony Brooks and Eva Irene Brooks (Eds.). Springer International Publishing, Cham, 229–243.

[13] Eike Langbehn, Paul Lubos, and Frank Steinicke. 2018. Evaluation of Locomotion Techniques for Room-Scale VR: Joystick, Teleportation, and Redirected Walking. https://doi.org/10.1145/3234253.3234291

[14] Jr LaViola. 2000. A discussion of cybersickness in virtual environments. *ACM SIGCHI Bulletin* 32 (01 2000), 47–56. https://doi.org/10.1145/333329.333344

[15] Raghav Mittal, Sai Anirudh Karre, and Y. Raghu Reddy. 2021. Designing Limitless Path in Virtual Reality Environment. In *Virtual, Augmented and Mixed Reality*, Jessie Y. C. Chen and Gino Fragomeni (Eds.). Springer International Publishing, Cham, 80–95.

[16] Jannik A. I. H. Neerdal, Thomas B. Hansen, Nicolai B. Hansen, Kresta Louise F. Bonita, and Martin Kraus. 2020. Navigating Procedurally Generated Overt Self-overlapping Environments in VR. In *Interactivity, Game Creation, Design, Learning, and Innovation*, Anthony Brooks and Eva Irene Brooks (Eds.). Springer International Publishing, Cham, 244–260.

[17] Thomas Nescher, Ying-Yin Huang, and Andreas Kunz. 2014. Planning redirection techniques for optimal free walking experience using model predictive control. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*. 111–118. https://doi.org/10.1109/3DUI.2014.6798851

[18] Christian T. Neth, Jan L. Souman, David Engel, Uwe Kloos, Heinrich H. Bulthoff, and Betty J. Mohler. 2012. Velocity-Dependent Dynamic Curvature Gain for Redirected Walking. *IEEE Transactions on Visualization and Computer Graphics* 18, 7 (2012), 1041–1052. https://doi.org/10.1109/TVCG.2011.275

[19] Niels Christian Nilsson, Tabitha Peck, Gerd Bruder, Eri Hodgson, Stefania Serafin, Mary Whitton, Frank Steinicke, and Evan Suma Rosenberg. 2018. 15 Years of Research on Redirected Walking in Immersive Virtual Environments. *IEEE Computer Graphics and Applications* 38, 2 (2018), 44–56. https://doi.org/10.1109/MCG.2018.111125628

[20] Niels C. Nilsson, Stefania Serafin, Morten H. Laursen, Kasper S. Pedersen, Erik Sikström, and Rolf Nordahl. 2013. Tapping-In-Place: Increasing the naturalness of immersive walking-in-place locomotion through novel gestural input. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. 31–38. https://doi.org/10.1109/3DUI.2013.6550193

[21] Tabitha C. Peck, Mary C. Whitton, and Henry Fuchs. 2008. Evaluation of Reorientation Techniques for Walking in Large Virtual Environments. In *2008 IEEE Virtual Reality Conference*. 121–127. https://doi.org/10.1109/VR.2008.4480761

[22] Adalberto L. Simeone, Niels Christian Nilsson, André Zenner, Marco Speicher, and Florian Daiber. 2020. The Space Bender: Supporting Natural Walking via Overt Manipulation of the Virtual Environment. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 598–606. https://doi.org/10.1109/VR46266.2020.00082

[23] Daniel Simons and Christopher Chabris. 1999. Gorillas in Our Midst: Sustained Inattentional Blindness for Dynamic Events. *Perception* 28 (02 1999), 1059–74. https://doi.org/10.1068/p2952

[24] Richard Stoakley, Matthew Conway, and Y Pausch. 1970. Virtual Reality on a WIM: Interactive Worlds in Miniature. (02 1970). https://doi.org/10.1145/223904.223938

[25] Ryan R. Strauss, Raghuram Ramanujan, Andrew Becker, and Tabitha C. Peck. 2020. A Steering Algorithm for Redirected Walking Using Reinforcement Learning. *IEEE Transactions on Visualization and Computer Graphics* 26, 5 (2020), 1955–1963. https://doi.org/10.1109/TVCG.2020.2973060

[26] Evan A. Suma, Seth Clark, David Krum, Samantha Finkelstein, Mark Bolas, and Zachary Warte. 2011. Leveraging change blindness for redirection in virtual environments. In *2011 IEEE Virtual Reality Conference*. 159–166. https://doi.org/10.1109/VR.2011.5759455

[27] Evan A. Suma, Belinda Lange, Albert Skip Rizzo, David M. Krum, and Mark Bolas. 2011. FAAST: The Flexible Action and Articulated Skeleton Toolkit. In *2011 IEEE Virtual Reality Conference*. 247–248. https://doi.org/10.1109/VR.2011.5759491

[28] Evan A. Suma, Zachary Lipps, Samantha Finkelstein, David M. Krum, and Mark Bolas. 2012. Impossible Spaces: Maximizing Natural Walking in Virtual Environments with Self-Overlapping Architecture. *IEEE Transactions on Visualization and Computer Graphics* 18, 4 (2012), 555–564. https://doi.org/10.1109/TVCG.2012.47

[29] Jerald Thomas and Evan Suma Rosenberg. 2019. A General Reactive Algorithm for Redirected Walking Using Artificial Potential Functions. In *2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. 56–62. https://doi.org/10.1109/VR.2019.8797983

[30] Khrystyna Vasylevska, Hannes Kaufmann, Mark Bolas, and Evan A. Suma. 2013. Flexible spaces: Dynamic layout generation for infinite walking in virtual environments. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*. 39–42. https://doi.org/10.1109/3DUI.2013.6550194

[31] Betsy Williams, Gayathri Narasimham, Timothy McNamara, Thomas Carr, John Rieser, and Bobby Bodenheimer. 2006. Updating orientation in large virtual environments using scaled translational gain. *Proceedings - APGV 2006: Symposium on Applied Perception in Graphics and Visualization*, 21–28. https://doi.org/10.1145/1140491.1140495

[32] Betsy Williams, Gayathri Narasimham, Björn Rump, Timothy McNamara, Thomas Carr, John Rieser, and Bobby Bodenheimer. 2007. Exploring large virtual environments with an HMD when physical space is limited. *ACM International Conference Proceeding Series* 253, 41–48. https://doi.org/10.1145/1272582.1272590

[33] Michael A. Zmuda, Joshua L. Wonser, Eric R. Bachmann, and Eric Hodgson. 2013. Optimizing Constrained-Environment Redirected Walking Instructions Using Search Techniques. *IEEE Transactions on Visualization and Computer Graphics* 19, 11 (2013), 1872–1884. https://doi.org/10.1109/TVCG.2013.88