Limitless Dynamic Path Generation For Continuous Locomotion in Virtual Reality

Thesis submitted in partial fulfillment
of the requirements for the degree of

*Master of Science in **Computer Science and Engineering** by Research*

by

RAGHAV MITTAL
2018701023
raghav.mittal@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
February 2023

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled “Limitless Dynamic Path Generation For Continuous Locomotion in Virtual Reality” by Raghav Mittal, has been carried out under my supervision and is not submitted elsewhere for a degree.

Date	Adviser: Prof. Y. Raghu Reddy

To my teachers, parents, siblings, friends and everyone who inspired me.

Acknowledgments

First and foremost, I thank my thesis supervisor, Dr Raghu Reddy, for guiding me. His belief and trust in me motivated me to strive for the best. Because he trusted in my abilities, I discovered an interest in topics I had never found interesting, such as linear algebra and algorithms. He always gave me much freedom to think and discuss new ideas, and funding was never a concern. I firmly believe that due to such a culture, I have achieved more than my expectations as a master's student. Also, he inspired me to plan and follow a daily routine at work and outside.

I thank Mr Sai Aniruddha Karre and Mr Y. Pawan Kumar, with whom I co-authored multiple conference papers. Mr Karre's inputs and efforts played a crucial role in shaping this thesis. I am thankful to all my labmates for being supportive and open to discussions. And for being there to celebrate all minor and significant achievements.

This thesis was never possible without the constant support of my elder brother Abhinav Mittal. My deepest love and thanks to him for being the central pillar of my academic life. He is the one who introduced me to web development in my pre-teenage years, later to C programming and then to Python. He always took time out of his busy schedule, keeping all his stress aside to check on my progress, motivate me, and help me drive through difficulties.

I am thankful to Dr Venkatesh Cheppella. During my bachelor's years, I did multiple summer internships and my bachelor's thesis under his guidance, which changed my perspective on software engineering. During those internships, I decided to pursue a master's at IIIT-H.

I was a research student in Dr Priyanka Shrivastava's Perception & Cognition (PAC) group for more than a year at Cognitive Sciences Lab in IIIT-H. I enjoyed my time working with her. I am thankful to her for being open and supportive of me. Whether it was about academics or personal life, I felt comfortable opening up to her. Her empathetic approach toward her students influenced me profoundly.

My most special thanks to my support system and friends Vrushali Arute, Minaxi Goel and Kunal Wadhwani for sailing with me in the ups and downs of student life. I feel lucky and blessed to have them in my life. I also thank my Yuktahar "family", with whom I used to have my meals, long discussions, and night walks. I learned so much from them.

Abstract

Walking in a virtual environment is a bounded activity. It is challenging to create and navigate in a large virtual environment given a limited physical space. Various techniques like walking on a treadmill, gesture-based walking and redirected walking address this problem in different ways. These existing solutions have challenges like dependency on additional hardware, inability to adopt different room sizes, compromised rendering quality etc. Such trade-offs make the VR setup immobile and expensive for VR applications which require endless locomotion of the participating subjects.

We developed a new redirected walking method to allow the subject to walk continuously within a predefined play area without needing additional hardware support. It is a software-oriented system which dynamically generates an endless path for the participant to walk forward continuously. The system is adaptable to varied play area sizes and is highly mobile because no additional hardware setup is required. Redirected walking consists of methods which actively manipulate the virtual environment to fit it in a smaller physical space. Such techniques allow the user to explore a virtual environment larger than the physical space. Covert redirected walking and overt redirected walking are two sub-classes of the redirected walking techniques. Covert techniques manipulate the environment without being perceivable to the user, and overt redirection techniques do such manipulations while being noticed by the user. Our work is classified as an overt redirected walking technique.

The system employs a path generation technique which is inspired by the path generation mechanics of mobile-based endless running games. Generally, the path is composed of successive chunks called “Tiles” in the endless running games. The path initially consists of a fixed number of tiles. When the player begins to move forward, a new tile is added towards the end and a tile is removed from the beginning of the path. This behaviour continues until the user terminates the game. Such an approach creates a perception of an infinitely long path and minimizes memory requirements. We designed an endless path generation system which dynamically generates and updates a path within the bounds of the play area.

To evaluate the path generation system, we developed a virtual art gallery. The art gallery is an endless, dynamically generated corridor with paintings/art items hanging on the walls. The path generation system generates and updates the gallery when the user moves forward in the gallery. We conducted three user studies to validate our work and gather feedback to check whether the generated environment induces sickness in the subjects and whether it can adjust to different room sizes without affecting the user’s experience of presence and ease of locomotion. In the last and final user study, we employed a simulator sickness questionnaire (SSQ), a presence questionnaire (PQ) and a custom questionnaire (CQ). No significant differences were found in the scores of all three questionnaires between the two different room sizes. We found an overall low score in the experienced realism sub-scale of the PQ, indicating that the subjects didn't perceive the generated environment as very real. CQ results suggest that the participants walked normally.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In 1965, Southerland presented his vision of an "ultimate display" - a technology to mimic all of the human senses virtually [46]. However, due to the immense amount of computing resources and space requirements, such a technology was not considered reasonable.

In 1989, Jaron Lanier coined the term "Virtual Reality" (VR) [35]. It generated a lot of buzz, which resulted in contributions from the research community that helped Southerland's vision come true. This enthusiasm led to the rise of CAVE systems in 1992, made for academic purposes [27], [38]. However, despite advancements, VR failed to find widespread interest. The computing hardware was not advanced enough to generate real-time graphics to create convincing virtual environments.

VR technology picked up again in 2012 with the launch of commercial VR head-mounted displays (HMDs or headsets). In 2012, a Kickstarter campaign envisioned by Palmer Luckey gave birth to Oculus Rift to provide an affordable, high-quality HMD. The success of Oculus Rift, has seen a massive rise in the use of HMDs [5]. All major companies like Google, Samsung, Apple, Sony, and Valve compete to dominate the billion-dollar VR market [33].

According to a report published in 2021 by Omdia, the consumer VR market will be worth $16bn by 2026, a 148% increase from 2021. The VR headset penetration will grow from 2.4 headsets per 100 households to 6.3 in 2026, highlighting a long road for mass VR adaption. Nevertheless, active VR headset users are set to outnumber Xbox users by 2026 [21]. By 2024, the cumulative installed base of VR headsets is expected to surpass the 34 million mark. Continued improvements to VR hardware, such as introducing smaller and more fashionable devices, are likely to support the increased adoption of VR in the consumer and industrial settings.[4]

Healthcare, education, workforce development, and manufacturing are expected to be among the sectors most disrupted by VR technology. In 2021, Emirates launched a VR app to allow customers to explore the cabin from the comfort of their homes. Facebook also launched a VR remote work app, Horizon Workrooms, a VR experience specifically for people to work together using the company's Oculus headsets [4]. In the automotive industry, the virtual training module offers workstation training for new joiners to avoid accidents; as per Ford's report of 2015, employee injuries and accidents were reduced by 70% with the help of virtual training sessions. The system provides hands-on training for

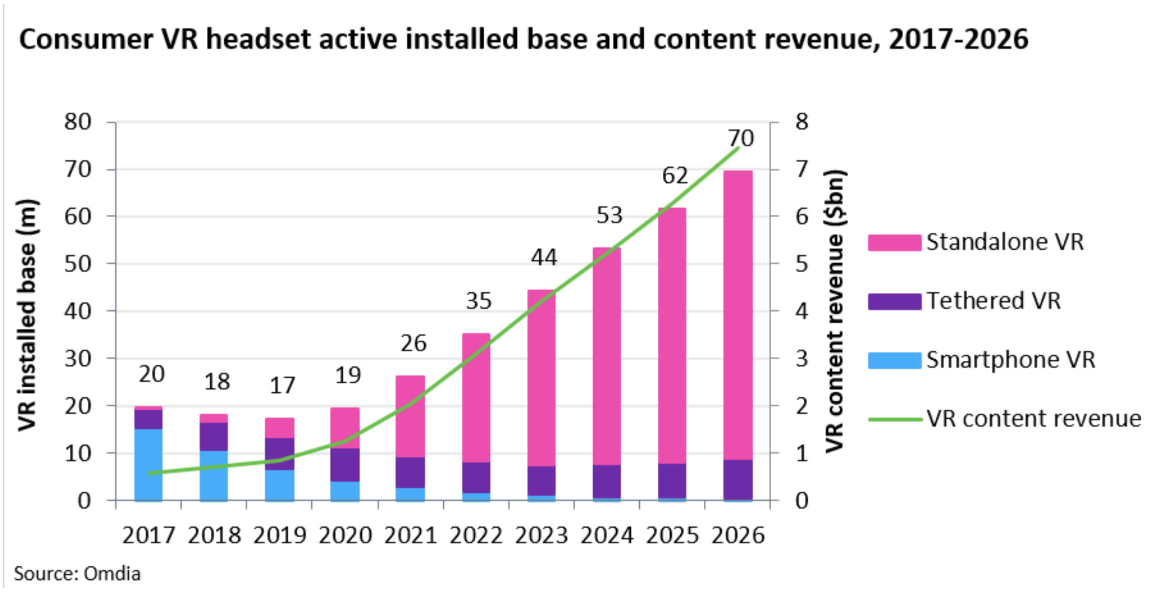

Figure 1.1 VR Forecast 2021

newcomers using headsets, simulators, glasses, and more. In addition, the retail industry has now started implementing VR technologies to maintain customer relationships. For instance, in July 2019, Walmart accomplished 15 minutes training modules instead of its earlier 8 hours modules with the help of virtual training tools [3]. Similarly, the healthcare industry is utilising virtual training solutions across its services. Training helps healthcare providers in improving operating accuracy with zero error. Again, in gaming and entertainment, the NFL and other American Football teams are using the technology for training to reduce the injury risk [3]. The automobile manufacturers Audi, Porsche and Ford collaborated with a VR startup, Holoride, to provide an in-car VR experience. The Holoride's technology turns a car into a gaming controller by calibrating graphics with car speed and orientation in real-time [1].

In traditional VR, the user is limited to viewing the environment from a single point. However, room-scale VR allows the user to move freely in a virtual environment just as they would in real life. Room-scale wireless VR headsets like Oculus Quest are becoming very popular [8][10]. These headsets do not necessarily require external (separate) hardware (trackers and workstations). They are also called "standalone" VR headsets.

The integrated inside-out positional tracking system for device localisation, and on-device computing, makes it possible to run VR independently of workstations. The tracking sensors in the Inside-out positional tracking system are attached to the participant's head, as opposed to the Outside-in positional tracking system, in which the sensors are attached to the physical space in which the participant is present (refer to figure 1.2) [13]. It is a significant advantage over traditional wired headsets like Oculus Rift, HTC Vive etc., which require an external hardware setup making the devices less mobile and bulky to carry. Standalone VR market share is rapidly increasing. In 2021, 83% of the consumer VR headsets sales were standalone headsets. It was largely driven by the success of Meta (Facebook) Quest/Quest

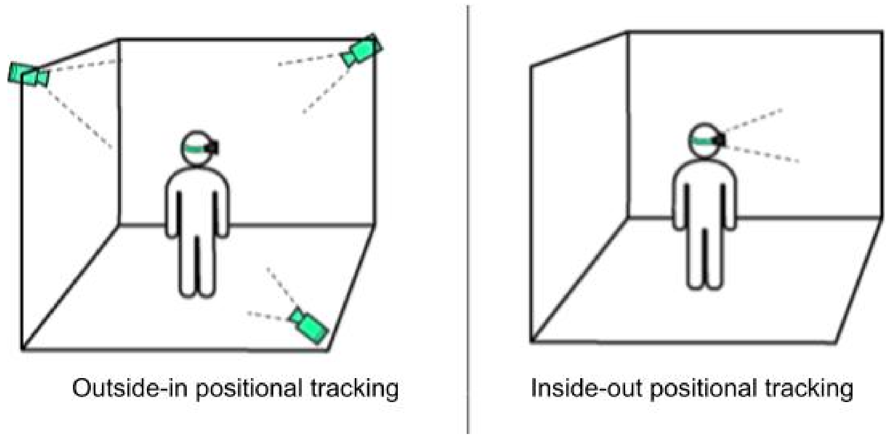

Figure 1.2 Outside-in and Inside-out positional tracking in VR. [13]

2 headsets which provide an immersive, untethered VR experience at a low price point [21]. Following Oculus Quest and Quest 2.0, HTC, Pico and Lenovo also launched multiple standalone VR devices between 2020 to 2022.

Real-world environments are bounded by physical obstacles, for example, walls. Walking in a virtual environment larger than the available physical space may harm the user's life. Also, VR devices can only track a limited physical area. These limitations acts as a virtual obstacle because walking out of the tracked area may disturb the overall user experience. Interestingly, such restrictions combined with the recent advancements in VR (room-scale tracking and standalone VR headsets) open up opportunities to develop various techniques to make the locomotion in VR easy and efficient.

1.1 Background

1.1.1 Virtual Reality, Immersion and Presence

Over the years, several attempts have been made to define the term **"virtual reality"**. Here are some examples - "real-time interactive graphics with 3D models, combined with a display technology that gives the user the immersion in the model world and direct manipulation" [7]; "immersive, interactive, multi-sensory, viewer-centred, 3D computer-generated environments and the combination of technologies required for building environments" [10].

There are three fundamental features of VR systems: immersion, presence in the virtual environment and interaction with that environment [9]. One of the pioneers of the VR field, Slater, described immersion as follows – "**immersion** is a description of a technology, and describes the extent to which the computer displays are capable of delivering an inclusive, extensive, surrounding and vivid illusion of reality to the senses of a human participant." He has differentiated immersion from presence in the following line "Immersion can be an objective and quantifiable description of what any particular system does provide. **Presence** is a state of consciousness, the (psychological) sense of being in the virtual environment.” [40]

1.1.2 Locomotion, Locomotion Techniques and Redirected Walking

Locomotion is an essential aspect of VR. It is the ability to move in space. It is one of the integral components of VR because it allows the users to perform tasks like **exploration, search and manoeuvring** in virtual environments [28]. Perhaps these are the tasks most frequently performed by humans in the real world.

A **locomotion technique (LT)** is a combination of different movements and tasks through which a VR user navigates in a virtual environment. Di Luca et al. [11] noticed that there isn’t a perfect LT that satisfies all the requirements of all the users. Each LT is designed to cater for a specific set of tasks. Real-world factors like physical space and energy consumption determine the best LT for each use case. Often there are discrepancies between the real world and virtual world situations. Such differences create space for the creators to develop new LTs different from actual real-world locomotion.

Redirected walking is a class of locomotion techniques. As the term “redirected” suggests, the system redirects the user from time to time to keep them within the physical space while walking in the virtual environment. The redirection system manipulates the scene that is displayed to the user, so the user unknowingly compensates for scene motion and can thus explore a large virtual environment in a limited space [14].

1.1.3 Degrees of Freedom (DoF)

The movements of a user in a 3D space can be classified into two types – **Rotational movements** and **Translational movements** (refer to figure 1.3).

Rotational movements consist of rolling, pitching and yawing. The ability to tilt from left to right and vice versa while facing forward is defined as rolling. Pitching is the ability to move your head toward and away from the chin to look up and down along a vertical axis. Yawing is the twisting of the head from left to right and vice versa.

Translational movements consist of strafing, surging, and elevating. The ability to move forward and backward is referred to as surging. It enables the user to move closer and further away from objects in the environment. The movement from side to side is known as strafing. Elevating is the movement up and down. It allows the user to crouch or jump inside the virtual world.

3-DoF VR headsets limit movement to the rotational axis, allowing the user to explore the environment by rotating their head only. The **6-DoF VR headset** can track both rotational and translational movements, allowing the user to walk around the environment without the use of external controllers

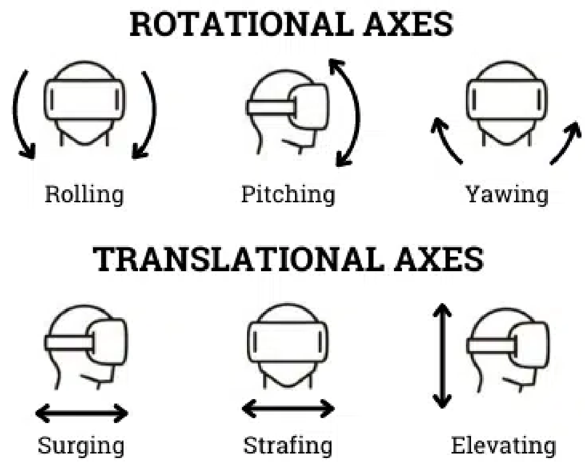

Figure 1.3 6-DoF headsets support head movement in all six of these axes, whereas 3-DoF headsets only support three. [51]

such as a joystick or treadmill. The user can then interact with, look around, and move within the virtual world. [1] [2]

1.1.4 Simulator Sickness

Simulator sickness is a symptom akin to motion sickness that can occur as a side effect of exposure to various virtual reality situations. Initially, the phrase "simulator sickness" was associated with effects caused by simulators that consisted of a platform, typically mobile, and visual stimuli created by a computer, but no head-tracking. The introduction of HMDs resulted in the creation of a new word, "cybersickness", because such devices generate additional issues that may potentially result in unpleasant symptoms, such as the delay between actual head motions and the generated image. However, researchers are now using both words to describe the unpleasant feelings elicited by virtual reality technology. [12] [22]

1.2 Scope of our work

Natural walking-based techniques in VR are known to increase immersion, lessen simulator sickness and improve user experience [28]. It might be because walking is a natural way of moving inside a virtual environment [50]. Several LTs allow a user to mimic actual walking behaviour.

An extensive collection of different LTs provided by [11] on their webpage[3] displays a number of creative strategies that aim to make the user's virtual experience limitless. Each technique serves a

[1] https://developers.google.com/vr/discover/degrees-of-freedom
[2] https://www.roadtovr.com/introduction-positional-tracking-degrees-freedom-dof/
[3] https://locomotionvault.github.io/

different purpose, but it can be classified based on aspects like DoF, the hardware used, and body posture requirements (standing, sitting etc.). We found that very few techniques allow natural walking without specialised hardware.

Some techniques depend upon specialised hardware setups, for example, an omnidirectional treadmill. There are software-oriented techniques too, for instance: redirected walking. Redirected walking techniques often have a condition on physical space requirements and may require high computational resources for proper functioning. Hardware-based LTs are not very attractive, especially for standalone HMDs, because binding a VR headset with any external hardware defeats the purpose of standalone devices, that is, mobility. Moreover, external hardware like treadmills is costly and very bulky.

We were interested in exploring LTs which allow a person to walk continuously in a limited virtual space naturally. We found that it can have multiple use cases in the sports, healthcare, entertainment and gaming domains. We wanted the technique to be cheap on resources (hardware, computational) and adaptable to different play area shapes and sizes.

This dissertation presents a software system to dynamically generate an endless path in a virtual environment to walk naturally without worrying about hitting the boundaries of the physical space or going out of the tracked area. We call it *Limitless Path Generation System*. It is a software-based LT which requires lesser computing resources and can be adapted to different room sizes.

We will first review the existing software-oriented LTs and later we present the thought process behind our technique, functioning, proof-of-concept implementation and validation.

1.3 Organization of thesis

The rest of the thesis is organised as follows:

Chapter 2 provides an overview of the related work on LTs for VR. In this chapter, we describe existing work proposed by the research community about LTs closely associated with our work.

Chapter 3 provides details about the novel limitless path generation algorithm and its main components, namely boundary detection system and path generation system

Chapter 4 introduces the system we have designed. It provides details about the system components and user interface.

Chapter 5 describes the implementation of a proof-of-concept called limitless art gallery to demonstrate the potential of our work.

Chapter 6 describes the enhancements which we have suggested to improve the performance of the proposed path generation algorithm.

Chapter 7 contains the validation of the proof-of-concept of our work. We show the methodology and the results of multiple user studies we did.

Chapter 8 is the concluding chapter in which we discuss our contributions to the community, the limitations of our work and the scope of future work.

Chapter 2

Related Work

This chapter introduces state of the art locomotion techniques for continuous natural locomotion in a virtual environment. We broadly classify locomotion techniques into the following categories:

- **Unnatural locomotion techniques**: Travelling techniques which do not require real walking to explore an environment.
- **Natural locomotion techniques**: Locomotion techniques which require natural bipedal motion to explore an environment.

2.1 Unnatural Locomotion Techniques

Several locomotion techniques allow for exploration of the environment without walking. Such techniques are divided into two categories: *Walk-in-Place (WIP) techniques* that require hand and foot gestures, and *hand-controller/joystick-based techniques*. Because they do not require the user to move from their original position, these techniques are easier to implement and more common in the market. The researchers and developers do not have to worry about the user colliding with any physical obstacles [48].

2.1.1 Walk-in-place techniques

WIP techniques let the user explore an environment without leaving their original position. For example jogging (hands+feet) [25], flying (hands only) [29], stepping (feet only) [47] etc. The user has to mimic different actions through body gestures. The gestures are captured either by the inertial sensors of the headset or through motion sensors attached to the body or by using a camera to track and identify the gestures. [54]

ArmSwingVR is a walk-in-place method in which the user has to swing their arms without the need for any foot or head movement. Pai et al. (2017) have claimed that this method is very low profile in comparison to the other WIP methods because it does not require any specialised peripheral devices to run [29]. Another WIP method called VR-STEP leverages inertial sensors of the mobile phone to translate the user's motion into virtual movement. The user has to mimick walking while standing in one place. However, this technique is only available for mobile phones [47].

WIP walking techniques are beneficial when there is not enough physical space for natural walking. WIP is considered to be the closest to actual walking but not superior. WIP is superior to hand-controller based locomotion, but it is not as good as actual walking. Natural walking provides higher presence levels and less oculomotor discomfort. Actual walking is more natural, more effortless, and uncomplicated than other locomotion methods [50]. In WIP methods, the turning is indicated by head rotation, because of which all linear motion is in the direction of the subject's gaze, which prevents looking around when walking [30].

2.1.2 Hand-controller based techniques

Hand-Controller based locomotion involves various metaphors like flying, teleporting, world-in-miniature etc. Many tasks can be accomplished by combining controller inputs (joystick and buttons) with appropriate visual feedback. For example, pointing to a distant location in a virtual environment and then pressing a button to move to that location instantly [16]. Or selecting a location on a mini-map, and pushing a button to land there in the virtual world [48]. In this section, we will majorly focus on *point and teleport (P&T) techniques*. We observed that P&T techniques are very prevalent in VR apps.

The P&T techniques allow users to move from one location to another instantly. The user points at a target location, where they want to be in the virtual world. Several authors have proposed different P&T techniques based on the pointing technique. For example, linear teleport, parabolic teleport, curved teleport or high-point curved teleport [16].

Many studies found that teleportation causes lower levels of simulator sickness when compared to walk-in-place and actual walking in VR. However, presence is less, and spatial disorientation is higher in P&T [34]. Another disadvantage of teleportation is that after teleporting to a new destination, the user needs to re-orient them because it is challenging to predict orientation before teleporting to a new location [16].

There is another hand-controller based technique called world-in-miniature navigation, which relies on a hand-held, scaled-down duplicate of the entire VE, where the users current position is displayed, and an interface provided to introduce his/her subsequent movements [43].

Though teleporting is a popular and easy to use solution, it takes some time for the user to understand her new surroundings after teleporting, potentially leading to disorientation, which in turn can break the feeling of presence [15][37][49].

Hand-controller based techniques support only limited self-motion experiences. However, other techniques, such as redirected walking, provide promising solutions to enable near-natural walking while overcoming the limits of the physical space. [24]

2.2 Natural Locomotion Techniques

Walking in the virtual environment while sitting on a chair is likely to cause VR sickness because it is different from the feeling of walking in a natural environment [53]. Natural locomotion approximates the user's mental model of navigation in the real world to that in the virtual world, which may improve the user's sensation of presence within the VE. [17].

Achieving natural locomotion in VR is a challenging task. Tracking the user in the play area requires advanced tracking systems. Keeping the user away from real-world obstacles is another challenge. Also, it is not easy to secure a safe space for a large virtual environment in a residential setting [53].

We classify natural walking techniques into three broad categories: hardware-based, break-in-presence, and redirected walking.

2.2.1 Hardware-based techniques

One of the ways to achieve actual walking in VR is by using specialised hardware like an Omni-directional treadmill (ODT) [29]. It enables the user to walk in a realistic manner while remaining stationary. The disadvantage is that a treadmill like the ODT is too expensive for most consumers. [29]. Furthermore, given the latest VR industry trends, headsets are becoming more mobile than ever. ODTs appear to be aimed at traditional VR headsets, which were not mobile and could only track a small area.

2.2.2 Redirected walking

There are two types of redirected walking techniques: covert redirection and overt redirection. Covert redirection techniques manipulate a scene in such a way that the user is unaware of the manipulations. Overt redirection occurs when the user notices changes to the scene that keep them within the physical space. Both types have advantages and disadvantages.

2.2.2.1 Covert redirection

There are several ways in which covert redirection can be achieved. It can be done through - (1) acoustic manipulations [14], (2) by using visuo-haptic manipulations [26], (3) by manipulating natural to virtual movement translation gains [23][36], (4) by manipulating the architecture of the environment [45] or (5) by using non-euclidean architectures [32]. We found redirection through curvature and translational gains, and redirection through architectural manipulation prevalent in the literature.

Redirection through curvature and translational gains Curvature gains are rotational manipulations applied during walking. The user's rotation in the physical world is converted to that in the virtual world with some manipulation of their actual turning angle [36]. Translational gains increase the user's linear movement velocity in a virtual environment. So that their virtual step size is longer than their actual step size leading to covering longer virtual distance with less walking [36]. Rotational gains can redirect a user when standing still and rotating their head [20].

Redirected walking techniques demand specific real-space and virtual environments. Some techniques only allow the user to walk in circular paths [23]. In Telewalk redirected walking technique, the user explores a virtual environment by walking on a circular physical path in a room of min 3 meters X 3 meters dimensions while walking on a straight virtual path. It employs a combination of curvature and translational gains [36]. In the technique proposed by Steinicke et al., users walked in a circular path of 22 meters radius without being able to detect the manipulation [42]. [31] leveraged the information loss which happens during the saccadic suppression to redirect the user towards the centre of the play area. Researchers have experimented with various combinations of translation and curvature gains to minimise the radius detection threshold, and the space requirement [18][19].

The method of curvature gains is limited when the radius is less than 22 m [41]. Many techniques require at least 6 meters x 6 meters of the walkable area to not to be noticed by the user [39]. Unfortunately, this space requirement is high and the availability of such spaces are not common [26].

Redirection through architectural manipulations Redirected walking can also be achieved by manipulating the architecture of the virtual environment. The covertness is achieved by manipulation of doorways in such a way that users exit a room from a different direction than the one they entered with, thereby reusing the space available [44]; overlapping adjacent rooms [45]; generating dynamic corridors that connect rooms by leveraging the two previous approaches [52].

"Impossible Spaces" is one such technique which relies on architectural manipulation. It makes it possible to compress sizeable virtual room environments into a smaller physical space by employing self-overlapping architecture. Koltai et al. (2020) found that relatively small virtual rooms can overlap up to 56% without the user's knowledge, and larger virtual rooms can overlap by up to 31% when mapped to a physical space of 9.14m x 9.14m [45]. Many authors have worked on such techniques by improving the user perception of infiniteness. Suma et al. (2013) demonstrated that it is possible to avoid detection and increase the amount of overlap between virtual rooms by connecting these rooms using longer corridors with more corners [52]. Smoothly curved corridors may be more beneficial for spatial compression. However, these techniques require specific types of VEs, such as indoor room-based layout [39].

Redirection through visuo-haptic manipulation Visual information is prioritised when conflicting vision, proprioception, and vestibular sensations are input simultaneously [26]. Keigo et al. (2019) proposed a visuo-haptic based technique in which the user walks in an unending corridor while gripping a handrail in the virtual environment, and simultaneously, they grip a curved handrail in the real world. These stimuli enable visuo-haptic interaction. The user perceives the curved handrail as a straight handrail. This technique keeps the user within the limited space, walking around the handrail. However, a major drawback with this technique is the specialised hardware requirement – the curved handrail – due to which it requires a large space and is less mobile when compared to software oriented redirected walking methods. [26]

2.2.2.2 Overt redirection

Overt redirection means that the manipulations are done to the environment to redirect the user and are visible to the user. Alternatively, we can say that the user is well aware of the manipulations done in the environment to keep them within the physical space. In our literature review, we found only two techniques which employ overt redirection - Resetting [26][39] and SpaceBender [39].

Resetting In the resetting technique, the subject can walk through a virtual space larger than the physical space. The method consists of two phases – walking and resetting. During the first phase, resetting is functionally equivalent to natural walking. When a subject reaches a boundary of the tracked space, the system initiates the second resetting phase. During this phase, the actual reset takes place, and the rotational gain of the system is modified so that a virtual turn of 360 degrees is equivalent to a real-world turn towards the exact centre of the room. Thus, subjects believe they have turned completely around and maintained their headings while they have been reset away from the boundary [26]. A significant drawback of resetting is that it requires users to stop and turn. It thus prevents users from walking continuously, reducing the level of immersion in the VE [26].

SpaceBender employs overt manipulations to the environment to keep the user within the physical space. The user walks in a straight corridor. Unlike stop and reset techniques, in Space Bender, the corridor is bent to accommodate the environment within the physical space whenever the user is close to the boundary. This technique is significantly faster than the stop-and-reset technique [39]. One main issue with this technique is the repetition of the environment. The corridor is generated along the parameter of the physical space. It takes a 90-degree turn whenever the user is close to the physical boundary.

Limitless Path Generation Technique (proposed in this thesis) It is a software-oriented locomotion technique. It can be downloaded and installed on any 6-DoF VR headset, unlike other infinite natural walking systems requiring specialised hardware like an omnidirectional treadmill. This technique requires fewer computational resources, unlike covert redirection techniques. It renders an environment with a random path to walk on, unlike SpaceBender, which has repetition.

Chapter 3

Limitless Path Generation Approach

In this chapter, we detail the *limitless path generation approach*.

Virtual Reality (VR) environments are designed from a participant's perspective. In HMDs, participant is considered to be the center of the VR scene and the environment is designed around the participant. This idea of centrality helps define the boundary of the VR environment for a given scene. VR practitioners normally design a full-scale scene to orient the participant within a fixed bounded environment rather than building a dynamic bounded environment due to various reasons like HMD limitations, physical space limits, dependency on additional hardware support, poor scene baking, poor frame rate, etc. Bounded environments influence the VR practitioners to build navigation controls through hand-held devices by letting the participant stay stationary. This contradicts the idea of creating realness in VR scene as the participant is forced to navigate the scene through hand-held controllers but in reality s(he) is stationary. To help address this issue, we developed a software-oriented approach to let the participant navigate beyond the control of a hand-held device by taking into account the following two factors:

- let the participant physically navigate in the VR environment without the influence of external haptic hardware.
- let the participant navigate seamlessly in a limitless path within a limited virtual play area.

The above two factors can be addressed by considering the underlying technical aspects

- As part of the predefined constraints, physical environment and virtual environment ratio must be maintained
- An algorithm needs to be built to generate a limitless path in a virtual environment put up in a constrained physical space.
- VR environment assets need to be generated based on the path and orientation of the participant in the virtual environment

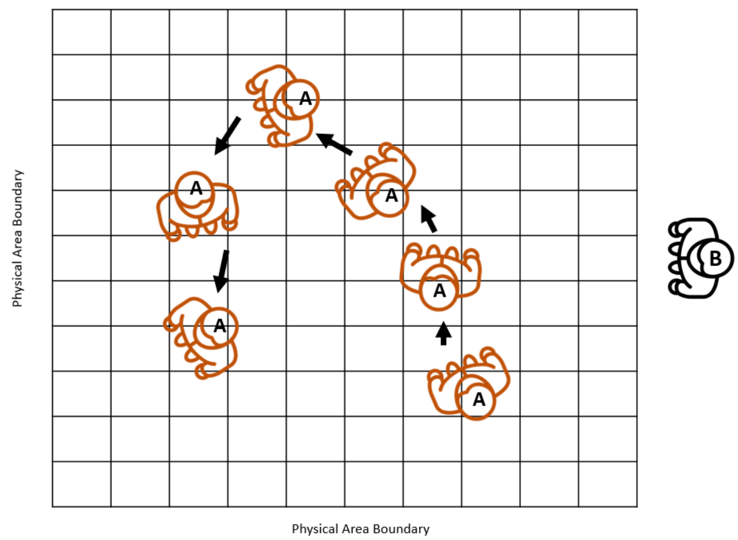

Figure 3.1 Person A experiencing limitless path in a finite physical environment

- VR environment should appear infinite to the participant. However, physically the participant will still be navigating in a limited predefined space.

We use the term ***'Limitless Path'*** in the context of obstruction-free navigation in a VR environment within a limited physical environment. We define limitless path as a dynamically generated never ending path which adapts to the space available in the VR scene. This path is limitless and unbounded in terms of length. The path progresses based on the user's location and the scene assets are generated inline with the generated path. For illustration consider figure 3.1, person A - standing in a 10ft x 10ft physical grid wearing a HMD and another person B outside this physical grid. Person A loads a VR scene with the support of limitless path implementation and infinitely walks in this limited space. Person A experiences an endless walk in the VR scene. For person B, person A appears to be walking within a 10ft x 10ft physical grid area continuously in a random path within the limited physical area.

3.1 The Concept

As any generated path must be within the specified boundaries of the VR scene pertaining to path generation and boundary detection play a key role in the progress of the participant along the limitless path. We designed and implemented both the path generation and boundary detection techniques in a virtual environment as part of our work.

Consider a use-case where a path has to be presented to the participant in a particular VR context. At the time of the VR scene generation, we are aware of the start position of the participant with in a virtual environment. As the participant walks forward, the path is generated by addition and removal of line segments in the path to create a perception of continuity. This path generation technique outputs

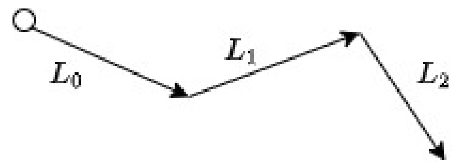

Figure 3.2 Line segments generated on start of algorithm. L_0 starts from player's position.

the path as a line made of multiple connected line segments, as shown in figure 3.2. The technique will generate 3 successive line segments (shown in figure 3.2 as L_0,L_1, and L_2) from the starting point. When the player reaches the end of L_1, the first segment L_0 is removed, and a new segment L_3 is added to the end of L_2 as shown in figure 3.3. This process keeps repeating until the participant terminates the program.

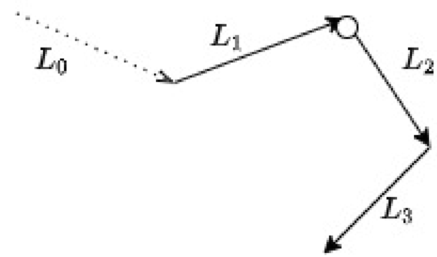

Figure 3.3 L_0 is removed and L_3 is added to the line when the player reaches the end of L_1.

Each produced line segment makes an angle with the previous line segment ranging from 45 to 135 degrees. The angle is determined by the position of the previous line segment's finishing point, the length required for the next line segment, and the size of the play area.

3.2 Path Generation Technique

In this section, we provide step-by-step details of path generation technique. We term *'Palling'* as a user action to move forward in the virtual environment for ease of terms. *1 Pal* unit is equal to 1 unit distance taken by the user in the virtual environment. For purposes of simplicity, we assume the dimension of the rectangular bounded area from $D(0, 0)$ to $D(x, z)$. If the user starts at $(0, 0)$, the user can take *z Pal* units to reach $(0, z)$ and *x Pal* units to reach $(x, 0)$. In order to generate a path, below are the necessary inputs:

- Player's starting position P_0 and head-yaw β_0 at that position.
- Dimensions of the boundary $D(x, z)$, here *x* and *z* are the limits of the boundary in a plane along *x-axis* and *z-axis*.

- Path properties include segment length l and path width w i.e. the perpendicular distance between parallel boundaries representing the width of the path in virtual environment.

As shown in figure 3.4, upon start of the application, the first line segment L_0 has a starting point P_0 and terminal point P_1 i.e. it is represented as L_0 = $\overline{P_0P_1}$. Similarly, for L_1, P_1 and P_2 are starting and terminal points respectively. Eq 3.1 generalizes this for L_i. Currently the length is considered to be a constant value and provided as a necessary input. [1]

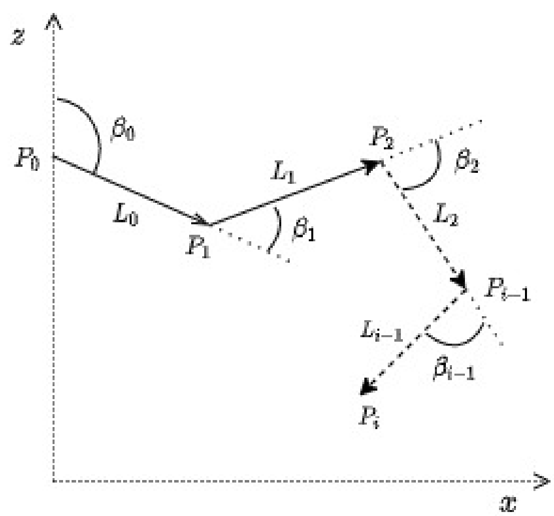

Figure 3.4 Locomotion of a participant in the play area

$$L_i = \overline{P_iP_{i+1}} \tag{3.1}$$

In the given play area, the coordinates of position P_{i+1} are calculated as shown in following equations 3.2 and 3.3, where $0 <= i <$ total number of segments generated since the start of the application. Here β_0 is head-yaw (HMD's orientation along y-axis in virtual environment) of the user.

$$P_{i+1}(x) = l * \sin \beta_i + P_i(x) \tag{3.2}$$

$$P_{i+1}(z) = l * \cos \beta_i + P_i(z) \tag{3.3}$$

When we recursively run equations 3.2 and 3.3, we generate a limitless path in a defined boundary of $D(x, z)$. The generated path is instantaneous, nonlinear, and limited by certain Pal units due to the bounded area. The upcoming P_i needs to be generated by detecting the proximity of P_{i-1} to the boundary. When P_{i-1} is not close to the boundary, β_i is set to a random value in range $\{\beta_{i-1} - \pi/2, \beta_{i-1} + \pi/2\}$. The procedure of generating the next line segment L_i for a given position P_i at a boundary is defined on β_i value.

[1] In the chapter 6, we have proposed enhancements in which the length of each line segment is set dynamically according to the size of the play area and the ending location of the previous line segment.

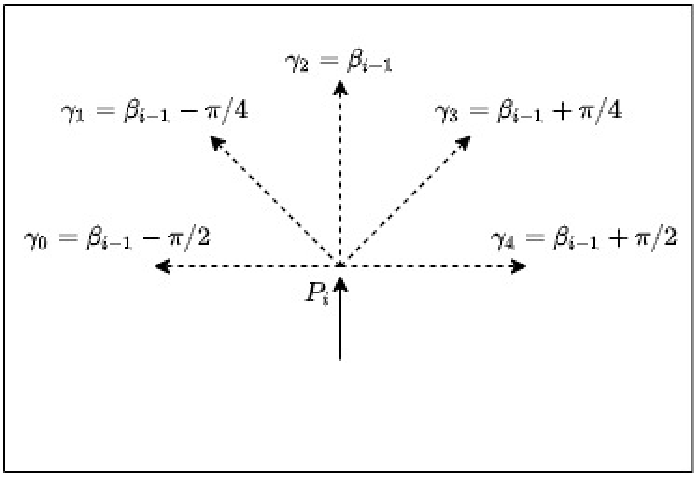

Figure 3.5 Dashed lines represent rays, solid line represents the direction of the line segment L_{i-1} of the existing path.

3.3 Boundary Detection Technique

In this section, we detail our boundary detection technique. For ease of terms, we refer to a user instance on detecting a boundary or hitting a boundary as *'Pragging'*. *1 Prag* unit is equal to one hit at a boundary.

To ensure that the player doesn't cross the boundary of the given application, the generated path must not span beyond the bounded-area. To achieve this, the bounded-area is enclosed into wall-like colliders. We define *'j'* as a value that equally divides a 180° range into multiple possible rays as shown in figure 3.5. Whenever a new point P_{i+1} has to be generated, *j+1* number of rays are projected in multiple directions with certain angles called as γ where k is in range $\{0, j\}$ and $j > 0$. The range of angle in figure 3.5 is $\beta_{i-1} - \pi/2, \beta_{i-1} + \pi/2$.

$$\gamma_j = \beta_{i-1} - \pi/2 + ((\pi/j) * k) \tag{3.4}$$

In the equation 3.4, if the value of j is equal to 4, we have j+1 rays i.e. 5 rays at equal angles between $\beta_{i-1} - \pi/2, \beta_{i-1} + \pi/2$ are called $\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4$ as shown in figure 3.5. Here the source of the rays is P_i and length is equal to $path\_length + path\_width/2$

If a ray collides with the play area boundary, we term it as 1 Prag unit. If *j=a*, we have *a+1* rays generated. Out of these *a+1* rays, one of the γ_i direction is chosen to generate a path L_i. Below are the possible cases in which the generated rays can collide with a boundary:

- If γ = *a+1 Prag* units, i.e. all generated rays hit the boundary as shown in figure 3.6. This means that the P_i is at the corner of the bounded area. In this case, ±135° is required to go away from the corner. Two more rays are shot in directions $\gamma_0^* = \beta_{i-1} - 3\pi/4$ and $\gamma_1^* = \beta_{i-1} + 3\pi/4$ to come out of the corner. The ray which don't hit the boundary is chosen as the direction for L_i. If P_i is equidistant from boundaries then one of the ray from γ_0^* and γ_1^* is chosen randomly.

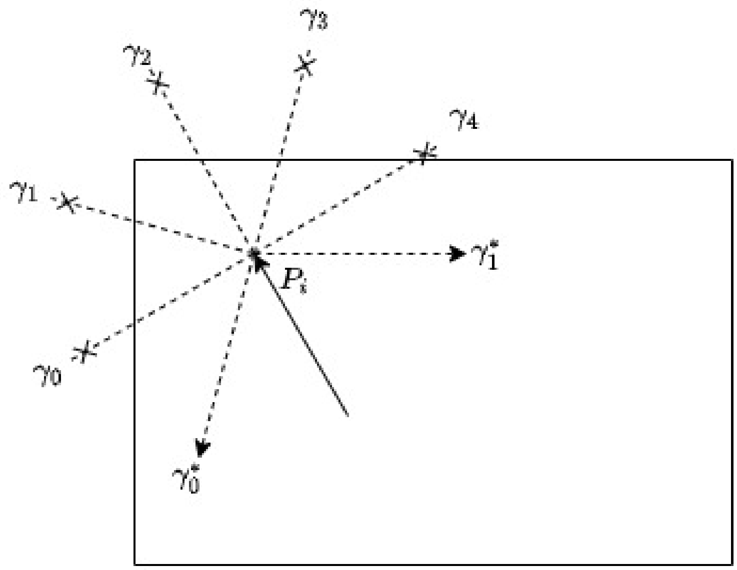

Figure 3.6 When all the rays $\gamma_0....\gamma_{j+1}$ hit the boundaries then two more rays are generated to decide the direction of L_i.

- If $1 \leq \gamma \leq a\ Prag\ units$ i.e. not all but atleast one ray hits the boundary as shown in figure 3.7 and anyone of the non-hitting ray's direction can be used to generate L_i.
- If γ = *0 Prag* units, i.e. none of the rays hit any of the boundaries. Here the path is free from boundaries as shown in figure 3.5. Then the β_i is randomly chosen from range $\beta_{i-1} - \pi/2, \beta_{i-1} + \pi/2$ for generating the upcoming L_i.

3.4 The Limitless Path Generation Algorithm - "PragPal"

We conduct path generation and boundary detection simultaneously to generate a limitless path for a user in a virtual environment. We call the resultant algorithm as "PragPal". The pseudocode of the "PragPal" algorithm used in the simulation of the limitless path generation approach is shown in figure 3.4. [2].

[2] https://github.com/raghavmittal101/path_gen_sys/

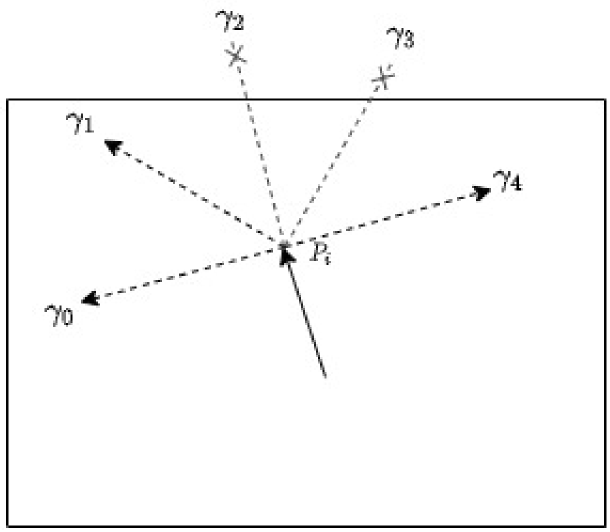

Figure 3.7 The γ for next L_i is chosen from γ_0, γ_1, γ_4.

Figure 3.8 PragPal Algorithm Pseudocode

```
Input: user position, head yaw, line segment length, path width
Output: List of 2D points x,z which represent a path
Function GeneratePoint_Pal(beta, point):
    point.x = segment_length*sin(beta) + point.x // Refer to eq. 3.2
    point.z = segment_length*cos(beta) + point.z // Refer to eq. 3.3
    return point

Function GenerateBeta_Prag(beta, point):
    valid_directions = [ ]
    for ( k = 0; k <= 4; k++ ) {
        ray_direction = beta - π/2 + ((π/4) * k)  // Refer to equation 3.4
        ray = Generate ray in direction ray_direction from point
        if ray do not hit boundary then
            valid_directions.push(ray_direction)
    if valid_directions.size == 5 then
        // Point is away from boundaries. Refer to fig 3.5
        beta = randomly choose from range {beta − π/2, beta + π/2}
    else if valid_directions.size > 0 then
        beta = randomly pick element from valid_directions;
    else
        // Point is in the corner of the play area. Refer to fig 3.6
        ray = generate ray in direction beta + 3 * π/4
        if ray hits boundary then
            beta = beta − 3 * π/4
        else
            beta = beta + 3 * π/4
    return beta

// Game engine's inbuilt method called to initialise a scene.
Scene.Start:
    point = current position of player: [x, z]
    beta = current head-yaw of player in radians
    points_list = [ ]          // Points represent the shape of current path
    points_list.Append(point)
    // Show three line segments in the environment
    for ( i=0; i<=3; i++ ) {
        point = GeneratePoint_Pal(beta, point)
        points_list.Append(point)
        beta = GenerateBeta_Prag(beta, point)
// Game engine's inbuilt method called on frame refresh.
Scene.Update:
    points_list.pop_front()                 // removes the first element
    point = GeneratePoint_Pal(beta, point)
    beta = GenerateBeta_Prag(beta, point)
    points_list.Append(point)
```

Chapter 4

Limitless Path Generation System

In this chapter, we describe the architecture of a system to generate a virtual environment in HMD using the PragPal algorithm. While planning the system, we had multiple use cases in mind, for example, virtual art gallery, infinite cycling/running track and an environment for mental health therapies for agoraphobia patients. We found the following requirements to be common across the three use cases – (1) User should be able to configure the environmental properties like path width, textures, room size; (2) User should be able to share these configurations as presets with other users.

Figure 4.1 is a visualisation of the proposed concept. It shows a user walking through a corridor placed around the dynamically generated path. There are rectangular shaped *assets* placed on the rectangular walls (wall assets) forming the corridor. An *asset* is a visual object which has a location, an orientation and may have an interactive behaviour.

The system consists of three types of users - *Participant, Designer* and *Developer*. *Participant* intends to experience a virtual environment. *Participant* can walk continuously on a limitless path within a bounded physical space. *Designer* provides virtual scene properties like texture, path's width and minimum length. They can create, store and access different path configurations in form of metadata documents. *Designer* can share the metadata documents as configuration preset with the *participant* in a convenient way. *Developer* is someone who intends to build upon or extend the path generation system. We have implemented a proof-of-concept of our work called the Virtual Art Gallery discussed in the chapter 5.

4.1 System Design

Figure 4.3 is a diagram of all the modules and their interaction with each other. Figure 4.2 shows an ideal flow of interaction between the users and the system. Below we describe each of the module in detail.

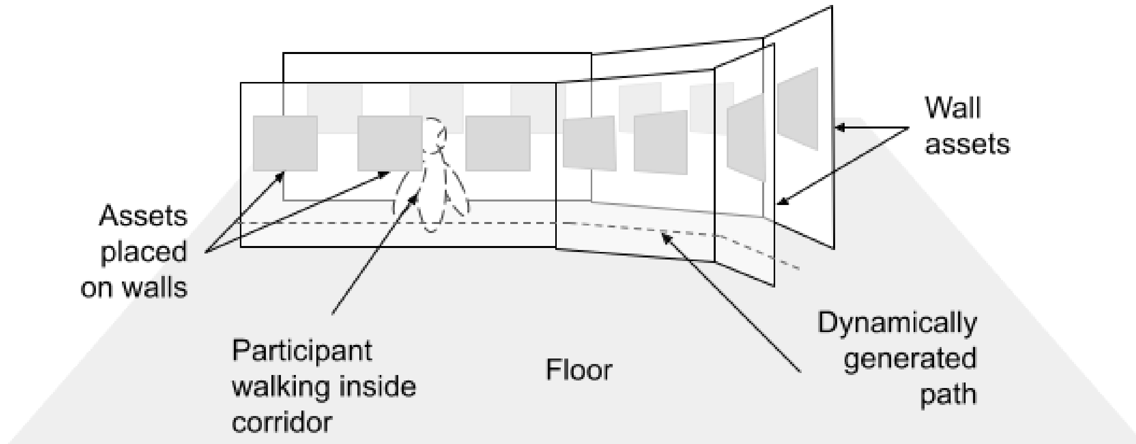

Figure 4.1 Visualisation of the output scene.

4.1.1 Scene Renderer Module

It is a rendering engine which provides tools and API support to build 3D scenes, add user interaction support and run the scenes on VR headsets. Game engines are often used in digital game development. In our project, we have used the Unity3D gaming engine. The system can render two scenes – **(1)** ***Main Scene*** in which the path is rendered, the *participant* walk on the path; **(2)** ***Metadata doc_id Input Scene*** in which the *participant* can enter a numeric input called *doc_id* (more is discussed in later sections).

4.1.2 Path Generation Module

This module consists of four sub-modules - **(1)** ***Line Generator (LG)*** responsible for line segment generation. **(2)** ***Boundary Detector (BD)*** responsible for boundary detection. **(3)** ***Assets Spawner (AS)*** responsible for placing the textures and assets on the output from *PG*. **(4)** ***Resource Fetcher (RF)*** provides path configuration called *path_info*, and assets related information called *texture_info* to the *AS* and *LG*.

LG outputs *path* which is a sequence of 2D coordinates based on the *play area dimensions*, *path_info* and scene interaction feedback. *AS* converts *path* into a visual and interactive form by placing assets based on *texture_info*. *RF* fetches the *path_metadata_doc* from the *Metadata Input Module* by sending a *doc_id* which is an input from the *Scene Renderer Module*. *RF* also derives the *path_info* and *texture_info* from *path_metadata_doc*.

4.1.3 Metadata Input Module

This module allows the *designers* to share their creations with the *participants*. It consists of a web interface, web server and a database. This module allows the *designers* to make new experiences without editing the source code of the package.

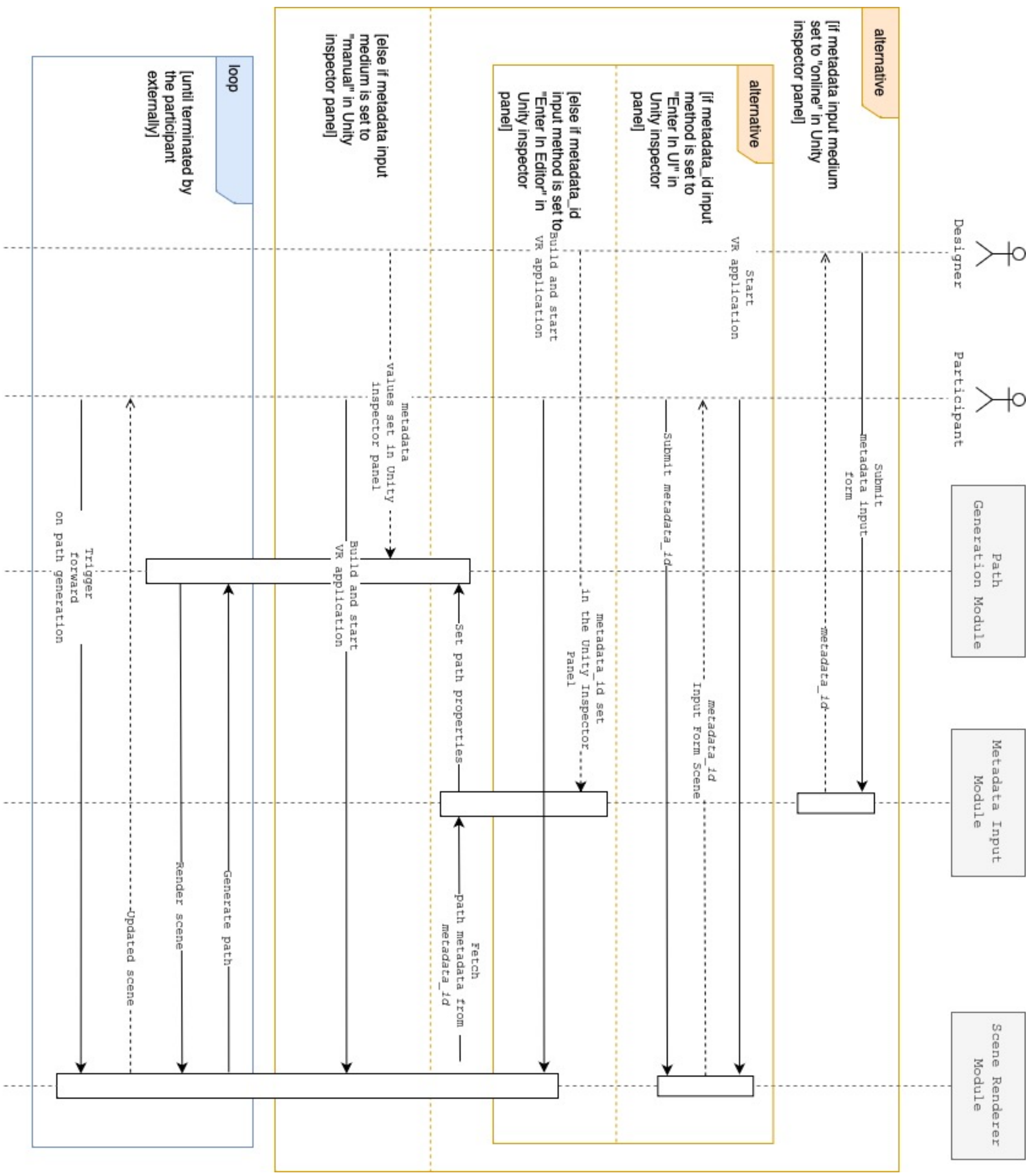

Figure 4.2 Ideal flow of interaction between the users and modules.

The *designer* can configure the specifications of the path to be generated. These configurations are called *path_metadata_doc* (refer to figure 4.10). *Path_metadata_doc* consists of the following attributes:

- Path segment width
- Path segment length
- Number of visible path segments at any time.
- Number of boundary detection rays
- List of images/textures

It is not necessary to use this module if the *designer* has access to the source code. They can set the same attributes from the inspector panel in Unity3D.

4.2 Assets Spawner

4.2.1 Corridor Generation

The limitless PragPal algorithm generates 2D-points P_i, P_{i+1}, P_{i+2} ..... and so on. It outputs points that can form a line to be used by a participant to traverse the limitless path in a virtual environment. In order to generate a corridor for walking, as shown in figure. 5.1 and figure. 5.2, we generate points Pr_i and Pl_i on right and left side of each point P_i respectively. These two new points Pr_i and Pl_i are equidistant from P_i and the distance between P_i and Pr_i or Pl_i is half the path width ($path\_width$). In the current version of our approach, the $path\_width$ is a static value defined by the programmer and is set as a default width of the visible or imaginary corridor. However, the path width may be changed depending the size of the boundary and future iterations of our work shall accommodate dynamic generation of path-width based on the VR scene and bounded area size. As P_i, P_{i+1} are generated by PragPal algorithm, the arrow on the solid line in figure 4.4 from P_i is normal vector to Pr_i and Pl_i towards P_{i+1}. Once we reach P_{i+1}, we move towards P_{i+2} along the arrow on the solid line. However, to position Pr_{i+1} and Pl_{i+1}, we generate a average vector (shown as dotted line arrow) of $\overrightarrow{P_{i+1} - P_i}$ vector and $\overrightarrow{P_{i+2} - P_{i+1}}$. The resultant average vector is dotted line arrow. Using this average vector, we generate Pr_{i+1} and Pl_{i+1}. Similarly, Pr_{i+2} and Pl_{i+2} and Pr_{i+3} and Pl_{i+3} so on will be created along with PragPal points. Joining all Pr_i's and Pl_i's points will provide us the corridor path required for our virtual art gallery. The code-implementation of this corridor generation[1] is included as part of our virtual art gallery scene. Using this corridor generation method, we generate boundary to the path width for better path visualization.

[1]https://github.com/SebLague/Curve-Editor/tree/master/Episode%2006

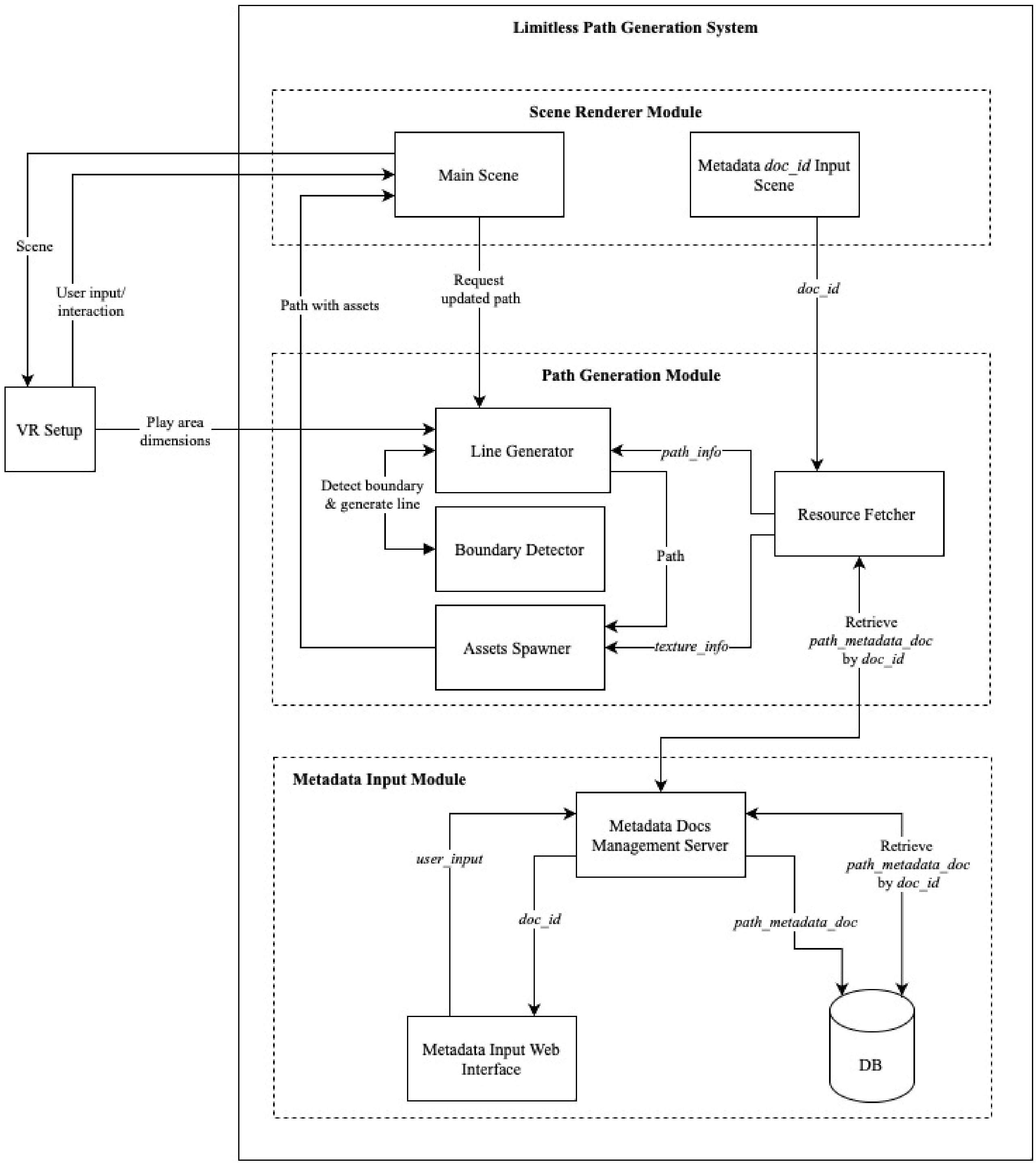

Figure 4.3 Module Diagram

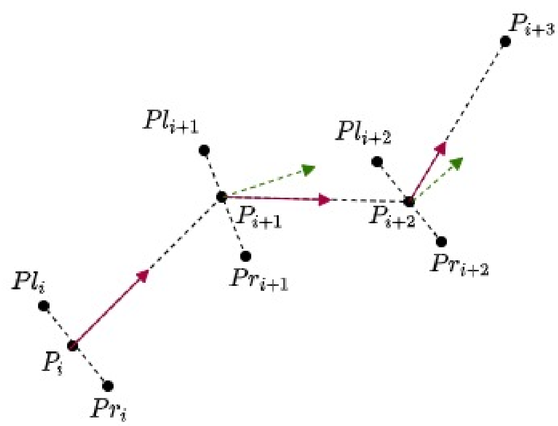

Figure 4.4 Placement of left and right points for corridor generation.

4.2.2 Positioning of paintings on walls

To generate an 'art-gallery' scene, we render virtual walls stretched between walls P_i, P_{i+1} (refer to figure 4.4. The art pieces are placed on the walls, as shown in figure 4.1. To achieve this, we first figure out the number of paintings which the wall can accommodate based on the width of the painting and width of the wall, and the spacing required between the paintings.

$$number\_of\_paintings = wall\_width/(painting\_width + 2 * spacing\_between\_paintings)$$

The $number\_of\_paintings$ is calculated by the program based on the number of paintings provided. $wall\_width$ is calculated by the program. The $painting\_width$ and $spacing\_between\_paintings$ have to be provided by the user.

4.3 Limitless Path Generation Package

To make it easy for the *developers* to reuse and tweak our project, we have created a Unity package which can be downloaded and opened in the Unity3D editor. Below we will explain the usage of this package.

4.3.1 Use cases

- A developer can access the program files which implement the *Path Generation Module* and the *Scene Renderer Module*.
- A developer can change the values of the path generation system variables from the inspector panel in Unity without the need to edit the program files.
- A developer can edit the scenes.

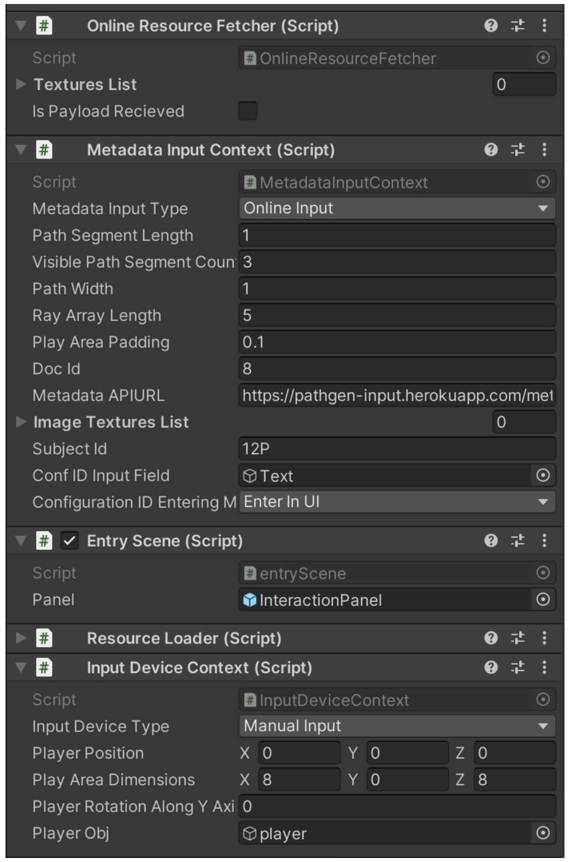

Figure 4.5 Unity inspector panel for Limitless Path Generation Package.

4.3.2 Features

- Metadata input options: Refer to figure 4.6, you can choose *Online Input* option to fetch metadata from a provided *Metadata Docs Management Server*. Or, choose *Manual Input* option to enter the same values in the inspector panel shown in the figure 4.5.

- Configuration input options: If you choose the *Online Input* option for metadata input, then *configuration input options are activated*. Choose *Enter in Editor*, if you want to enter the *doc_id* in the editor itself. Otherwise, choose *Enter in UI* option to let the *participant* enter the *doc_id* in the virtual environment (shown in the figure 4.7).

- Device input options: This option helps in running the scene on desktop instead of Oculus Quest. It helps the *developers* in quickly prototyping and testing the scenes without the need of deploying it to a VR headset every time. Refer to figure 4.8. Choosing the *Manual Input* option also allows you to define the play area dimensions manually.

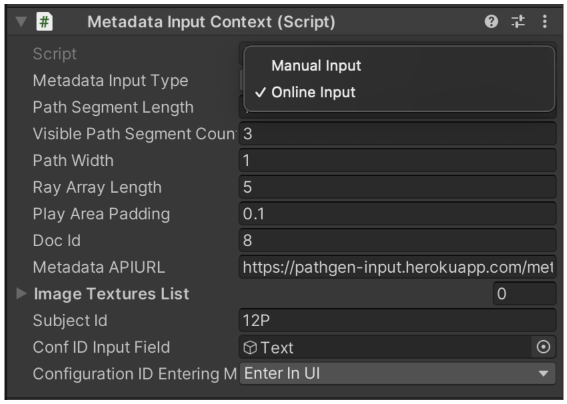

Figure 4.6 Options for metadata input.

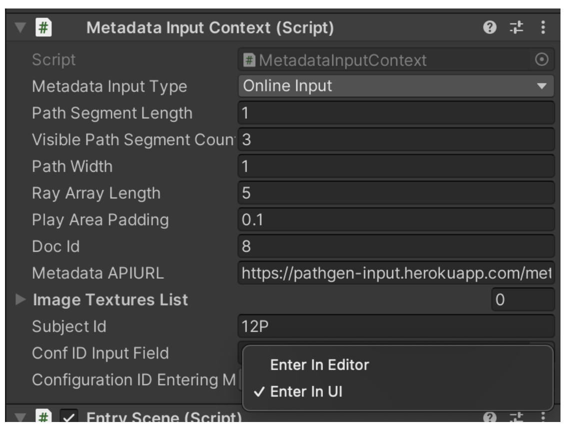

Figure 4.7 Options for entering the *doc_id*.

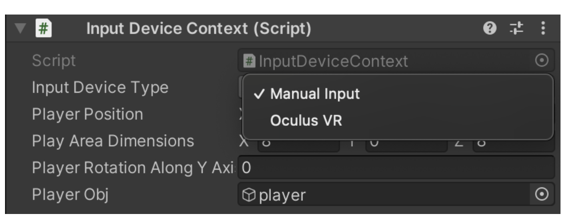

Figure 4.8 Option for choosing the medium for running the virtual environment.

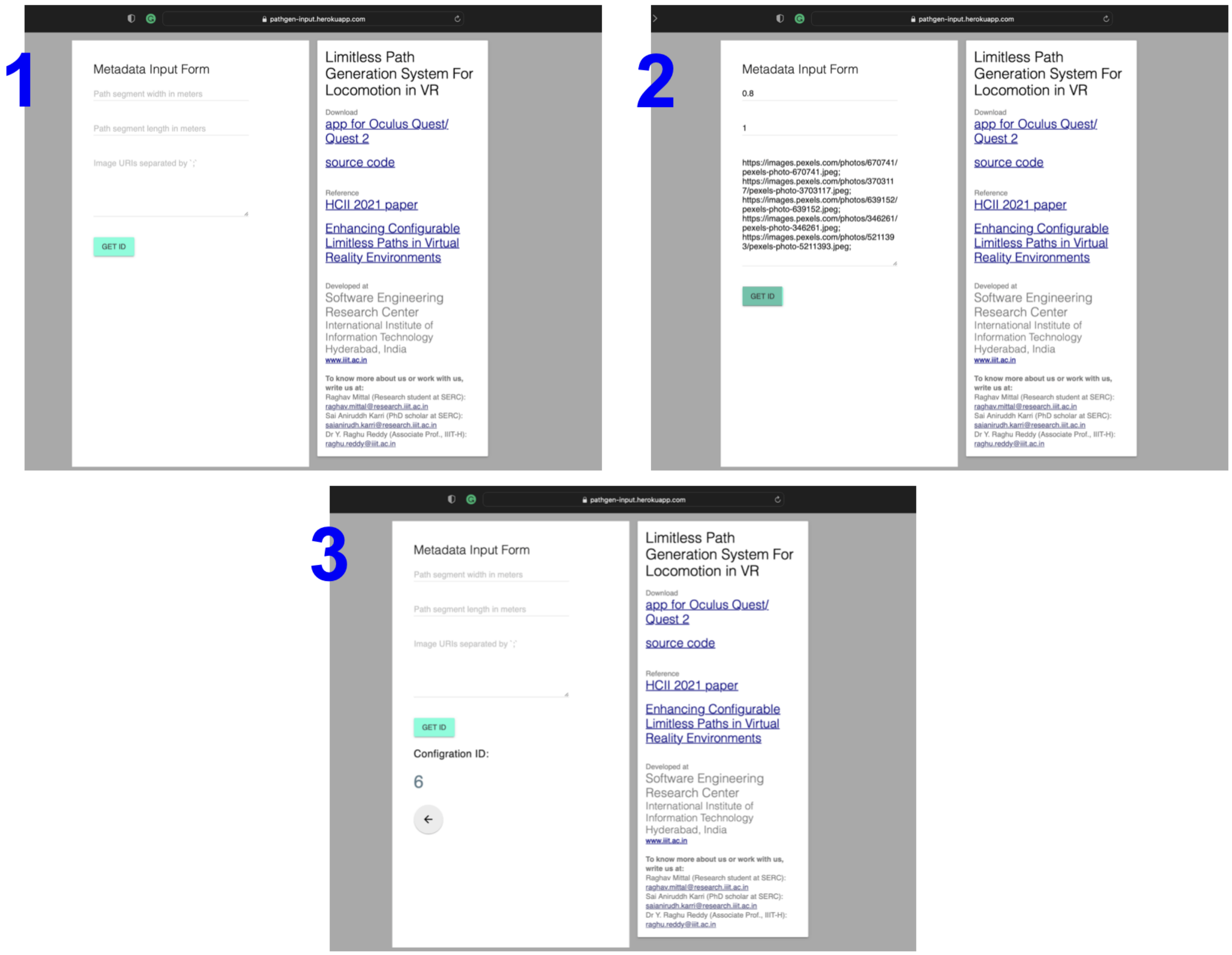

Figure 4.9 The web interface for inserting metadata into the DB. It is called *Metadata Input Web Interface*. On submission, it returns a unique ID (shown in the third step).


```
_id: 6
pathWidth: "0.8"
pathSegmentLength: "1"
imageURI: Array
   0: "https://images.pexels.com/photos/670741/pexels-photo-670741.jpeg"
   1: "https://images.pexels.com/photos/3703117/pexels-photo-3703117.jpeg"
   2: "https://images.pexels.com/photos/639152/pexels-photo-639152.jpeg"
   3: "https://images.pexels.com/photos/346261/pexels-photo-346261.jpeg"
   4: "https://images.pexels.com/photos/5211393/pexels-photo-5211393.jpeg"
visiblePathSegmen… : 3
rayArrayLength: 5
playAreaPadding: 0.3
```


Figure 4.10 The document format in which the metadata is stored on MongoDB cloud.

Chapter 5

Art Gallery - An endless art exhibition in VR using PragPal

To demonstrate a use-case of the proposed locomotion technique, we implemented a virtual art gallery. It is an endless corridor in which art items like paintings, pictures are placed on the walls on both the sides of the corridor. The user can walk through the corridor to explore the art pieces. Using the Metadata Input Module's web interface, the user can create a gallery by inputting a list of all the art items they want to show inside the gallery.

To build an art gallery, we provided a list of paintings as metadata to the Assets Spawning Module. We also provided path configuration as metadata. The rendered output environment can be seen in the figures 5.1 to 5.4.

In figure 5.1, you see the first person view of the environment. On the side walls the paintings/pictures are visible. The corridor has a turn to right. As the user walks further, the corridor will grow further. A top view of the environment in shown in the figure 5.2 in which you can see the overall topology of the current environment same as in 5.1. In figure 5.3, the top-view and first person view are shown side-by-side, the arrow signals the direction in which the player is moving and the camera represents the user walking in the environment. In the figure 5.4, a third person view is shown to give you an idea of how the environment grows in action.

This use-case is a simple instance of validating the limitless path approach. Our approach can be applied to various other use-cases or domains like entertainment, scientific research, studying spatial cognition, education, etc. to validate limitless paths.

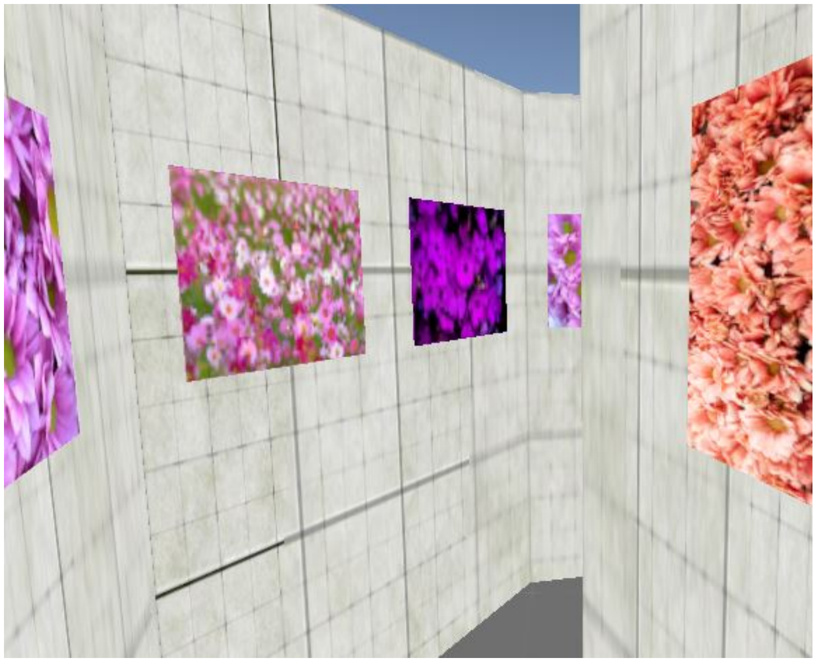

Figure 5.1 Inside view the virtual art gallery.

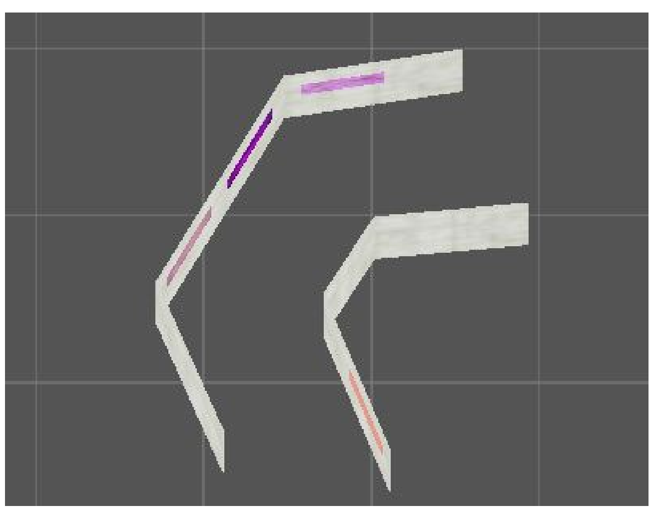

Figure 5.2 Top view of the virtual art gallery.

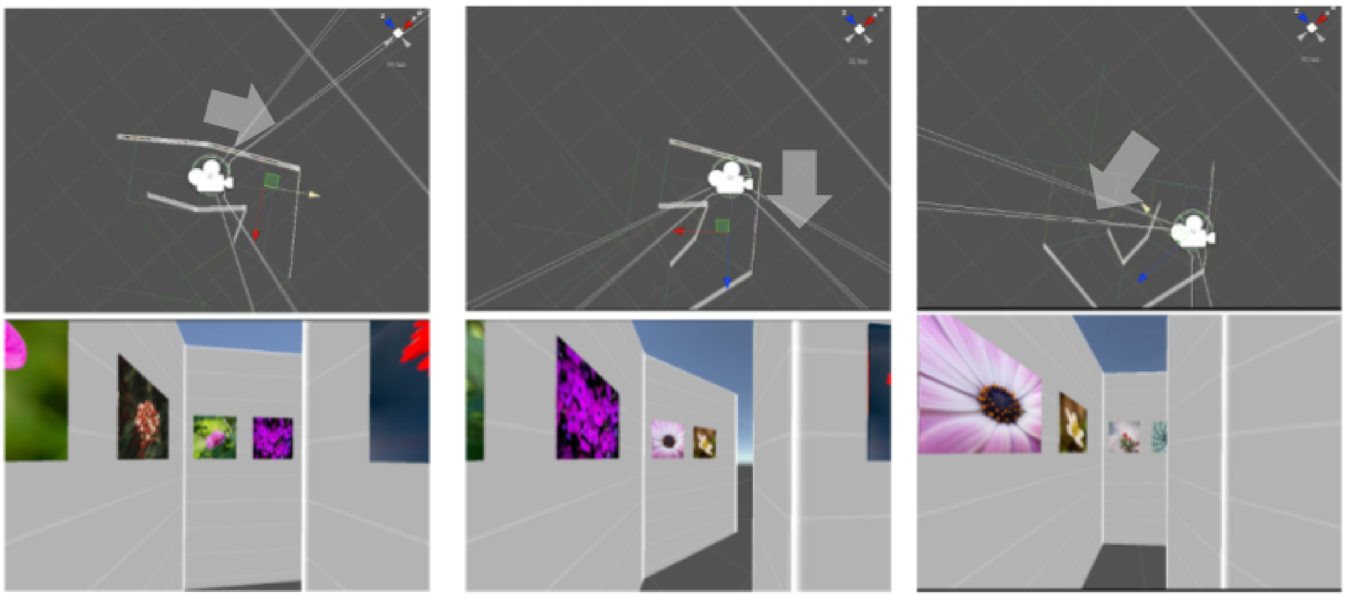

Figure 5.3 Top-view and a first-person view of the environment being generated. The arrow denotes the locomotion direction.

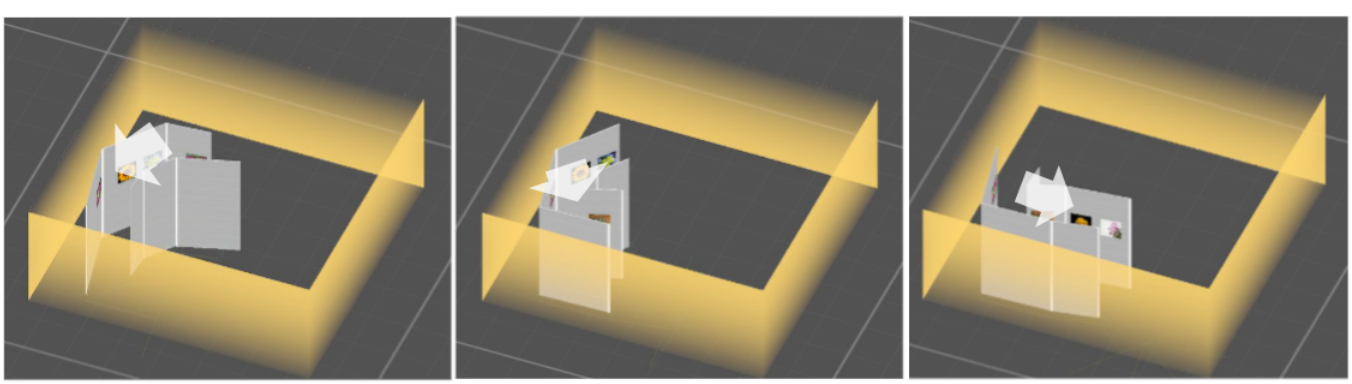

Figure 5.4 The third-person perspective of the environment being generated within room boundaries. The arrow denotes the locomotion direction.

Chapter 6

"Dynamic" Path Generation - Second Iteration of PragPal Algorithm

To better understand the behaviour of the "PragPal" algorithm, we ran multiple simulations of it. We found three cases in which there is a scope of improvement. In this chapter we propose and implement solutions to achieve those improvements.

6.1 Limitations of the initial version

6.1.1 Non-parallel walls

The left and right boundaries of the generated path were not parallel. As depicted in figure6.1, P_w is a fixed constant value and is not anticipated to change during the turns in the path. This distorts walls w_x and w_y causing irregular path width P_w. Logically, the width of the turn should be a bit higher than the regular straight path. 6.1.

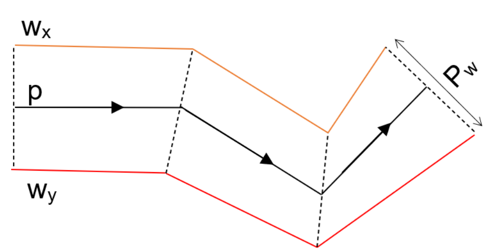

Figure 6.1 Narrow walls of path-*'P'*

6.1.2 Intersecting walls

The path collides with itself at corners of the play area under a unique case. In PragPal algorithm, a path is typically generated in the form of a line separated by walls with equal unit distance from the centre of the line as shown in figure 6.2. Here, p is the path generating line shown in black, w_x and w_y are the walls associated with the path line p. The walls w_x and w_y shown in red color are separated by path width P_w. Here, the path generated during the navigation in a virtual environment tends to collide with itself when the path has a 90-degrees turn followed by a subsequent 45-degrees turn or vice-versa as shown in figure 6.3.

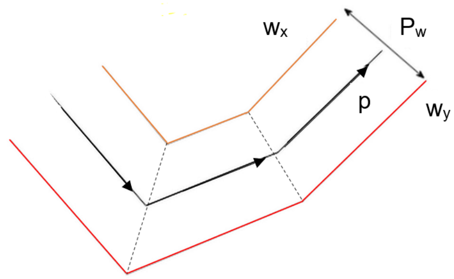

Figure 6.2 Path with walls.

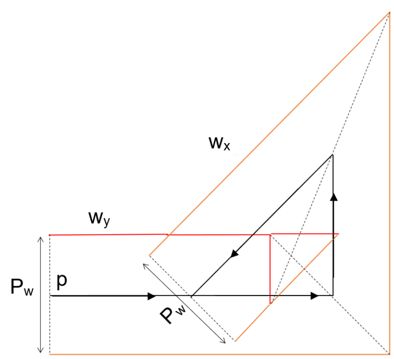

Figure 6.3 Walls intersecting when there is a 90-degrees and a subsequent 45-degrees turn.

6.1.3 Underutilised play area

Previously, the path segment had predefined constant length as part of PragPal algorithm. As a result, the play area at certain times may not be thoroughly utilized. Figure 6.4 illustrates this case with an example. This can be avoided if the longer walls are generated so that the participant in the virtual environment does not encounter frequent turns and shorter paths.

6.2 Enhancements

This section illustrates the enhancements to the observed deficiencies of the initially proposed PragPal algorithm.

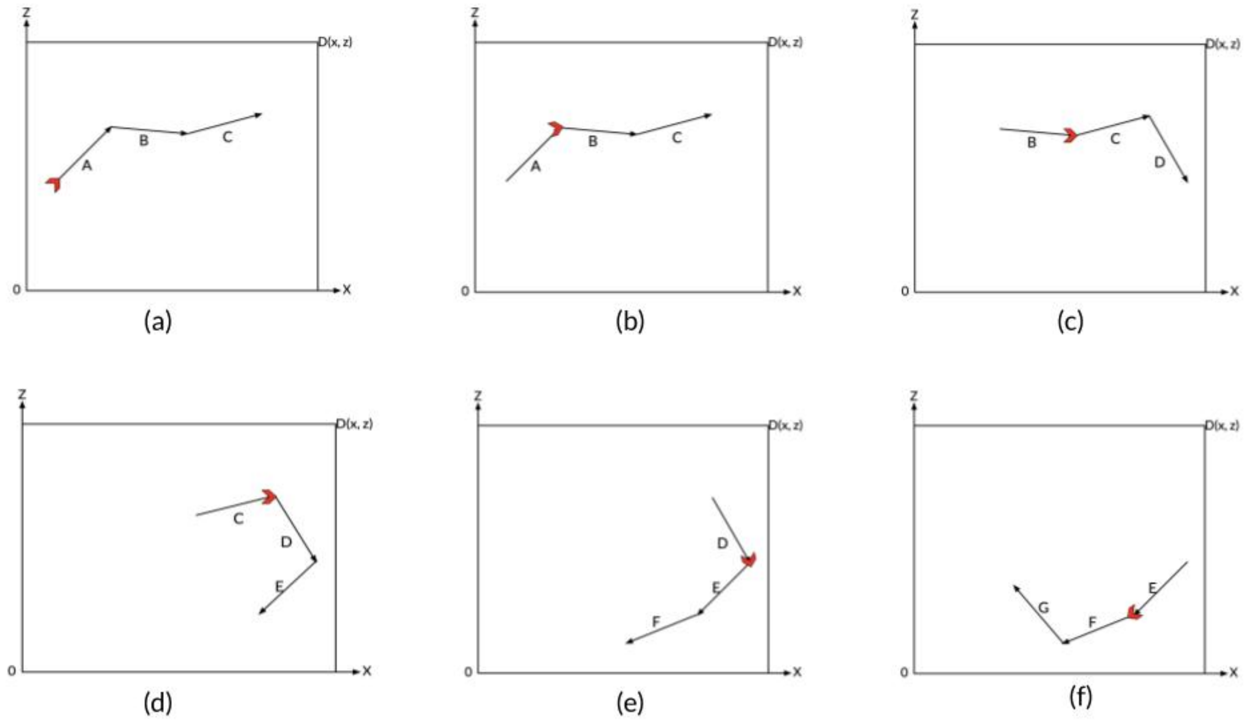

Figure 6.4 Underutilized physical room area with equal path segment (the line represents the path and the arrow is the participant's position and orientation)

6.2.1 Intersecting Walls - L to P_w Ratio

Considering the intersection of walls problem from subsection 6.1.2, L to P_w ratio can be deduced as a solution. Here L is the length of line-segment of the path, P_w is path width. The collision of the path walls w_x and w_y can be avoided at a turn to 90 degrees, followed by an abrupt turn at 45-degrees at a corner using a constant ratio. This constant ratio is between the length of the line segment (L) and the path width (P_w). It is defined in equation ratio. Here ρ is a constant multiplier. The value of the ρ is configured such that the walls are not set to intersect in such corresponding case.

$$path\_width(P_w) = \rho * path\_length(L) \tag{6.1}$$

Figure 6.5 shows the ideal case of the issue with intersecting walls. Here, the w_x wall is represented in an orange line, whereas the w_y wall is represented in a red line. The Path p is illustrated by a black line and is enclosed between w_x and w_y walls. Comparing figure 6.3 with figure 6.5, the w_y wall in red is distorted due to the intersection of itself. In an ideal case, w_y wall should be formed as a triangle as shown in figure 6.5. To avoid the obstruction, the path width P_w and path length L are to be set in a certain ratio as shown in equation ratio. Here, this ratio in normalized using the constant called ρ. The value of ρ ensures that the walls never collide in such an abrupt turn of 90 degrees followed by 45 degrees or vice versa. As shown in figure 6.5, L_1, L_2 and L_3 are the lengths of the path segment (p) rendered here are equal i.e. L_1 = L_2 = L_3. To avoid the collision of w_y wall (in red), the distance of AB with AC of the path p. As BD, DE and EB form a right angle triangle, AB, BC and CA also forms a right angle triangle. Considering $\triangle BDE$, as $BE = L_1$, $ED = L_2$ and $AD = L_3$, then $AB = (\sqrt{2} - 1) * L_3$. Then ρ can be derived as follows,

$$(\sqrt{2} - 1) * L = P_w/2 + P_w/\sqrt{2} \tag{6.2}$$

Here $P_w/2$ is length of AF and $P_w/\sqrt{2}$ is length of FC. Thus the ρ can be further defined as:

$$(2 * (\sqrt{2} - 1)/(1 + \sqrt{2})) * L = P_w \tag{6.3}$$

$$\therefore \rho = (2 * (\sqrt{2} - 1))/(1 + \sqrt{2}) = 0.343 \tag{6.4}$$

Using equation ratio, the collision of w_y wall can be avoided in a case of abrupt turn. For a reasonable walkable path, the path width and the path length needs to be big enough to populate the assets in virtual environment. Thus there may be a need for a larger physical play area to accommodate the full environment as illustrated in figure 6.5.

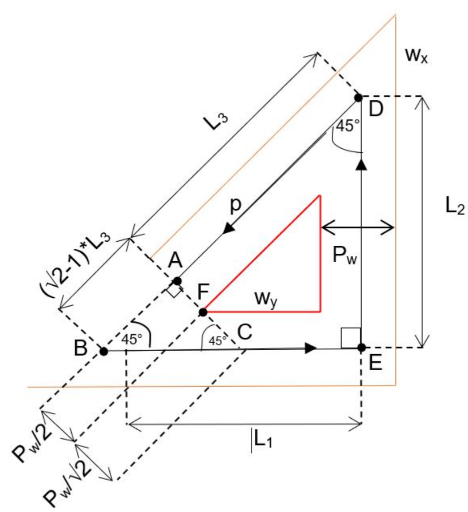

Figure 6.5 Ratio between path width and segment length to avoid collision.

6.2.2 Narrow Walls - Path Width at Turns (P_{tw})

While validating the initial PragPal algorithm, it was observed that the algorithm maintained a constant path width throughout the scene in the virtual environment. It means that even in the turns, the path width remained constant, due to which the walls of the path converging to some extent, causing a user experience issue for a participant to navigate across such a path as observed in figure 6.1. To address this case, the path width P_w at the turns must be adjusted based on the angle of the turn rather than being a fixed constant. Consider figure 6.6, P_w is the path width which is constant when path segment (p) is a straight line, P_{tw} is the path turn width which is a non-constant value at the turns. P_{tw} is defined as per below equation.

$$P_{tw} = P_w/(sin(\theta/2)), where\, \theta \;=\; 180 \;-\; angle\, of\, turn. \tag{6.5}$$

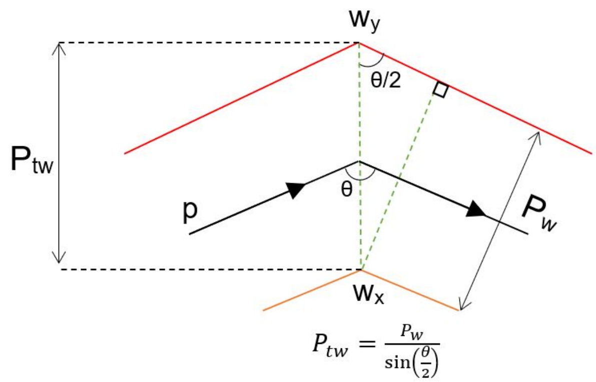

Figure 6.6 Calculating the width of path at turns.

Using P_{tw}, narrow walled paths can be avoided while generating limitless navigation in virtual environment.

6.2.3 Underutilized Play Area - Dynamic Path Segment Length

Even if the physical room area is larger enough, the generated path segment may be considerably smaller. As the path segment length was fixed as constant as per the PragPal algorithm, the maximum length of a generated path is also fixed as constant. This will lead to inefficient utilization of the physical room area. This causes frequent turning paths causing confusion among the participants who wanted to experience limitless navigation in the virtual environment. To avoid such a situation, the algorithm can be updated to dynamically set the segment length based on the proximity of the end of the existing path. This current path is generated dynamically from the boundary in the direction in which the new segment has to be generated, as shown in figure 6.7. Here, dP_l is the dynamically generated path length segment, P_l is the path line segment length which is fixed value, bd is the boundary distance which is the projected line that eventually hit the boundary from the end position of the last path segment generated for the participant to locomote in the virtual environment. The length of the new path segment will be randomly picked as per the following equation

$$dP_l = RAND(P_l, bd) \tag{6.6}$$

figure 6.7 illustrates the participant in red, navigating in the virtual environment by advancing further with unequal path segment lengths with less frequent turns. These path segments are dynamic in length, a random length unit value between fixed path segment length and boundary distance. Using the above

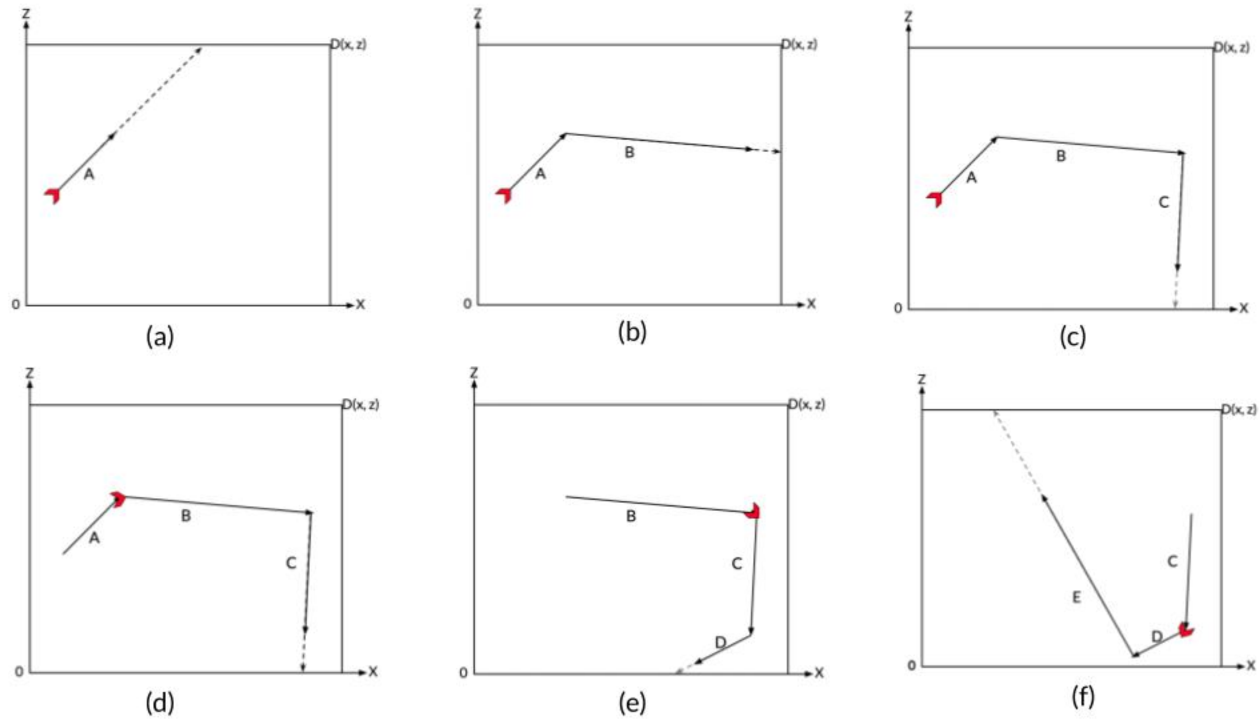

Figure 6.7 Dynamically set path segment length for efficient utilization of room area (the line represents the path and the arrow is the participant's position and orientation)

equation, underutilized physical room space can be avoided to a greater extent while developing a virtual environment scenes.

Figure 6.8 PragPal 2.0 Algorithm Overview

```
Input: user position, head yaw, line segment length, path width
Output: List of 2D points x,z which represent a path
Function GeneratePoint_Pal(beta, point):
    ray = ray from point in direction beta
    boundary_distance = length of ray from ray origin to ray collision
     point
    segment_length = random value in (line segment length,
       boundary_distance) // Refer to eq. 6.6
    point.x = segment_length*sin(beta) + point.x // Refer to eq. 3.2
    point.z = segment_length*cos(beta) + point.z // Refer to eq. 3.3
    return point

Function GenerateBeta_Prag(beta, point):
    valid_directions = [ ]
    for ( k = 0; k <= 4; k++ ) {
        ray_direction = beta - π/2 + ((π/4) * k)  // Refer to equation 3.4
        ray = Generate ray in direction ray_direction from point
        if ray do not hit boundary then
            valid_directions.push(ray_direction)
    if valid_directions.size == 5 then
        // Point is away from boundaries. Refer to fig 3.5
        beta = randomly choose from range {beta − π/2, beta + π/2}
    else if valid_directions.size > 0 then
        beta = randomly pick element from valid_directions;
    else
        // Point is in the corner of the play area. Refer to fig 3.6
        ray = generate ray in direction beta + 3 * π/4
        if ray hits boundary then
            beta = beta − 3 * π/4
        else
            beta = beta + 3 * π/4
    return beta

// Game engine's inbuilt method called to initialise a scene.
Scene.Start:
    point = current position of player: [x, z]
    beta = current head-yaw of player in radians
    points_list = [ ]          // Points represent the shape of current path
    points_list.Append(point)
    // Show three line segments in the environment
    for ( i=0; i<=3; i++ ) {
        point = GeneratePoint_Pal(beta, point)
        points_list.Append(point)
        beta = GenerateBeta_Prag(beta, point)
// Game engine's inbuilt method called on frame refresh.
Scene.Update:
    points_list.pop_front()                 // removes the first element
    point = GeneratePoint_Pal(beta, point)
    beta = GenerateBeta_Prag(beta, point)
    points_list.Append(point)
```

Chapter 7

Evaluation of Effectiveness of Limitless Path Generation Approach

We conducted one qualitative and two quantitative user surveys to evaluate the effectiveness of our work in terms of perceived naturalness of our environment, presence experience, and simulator sickness. All the subjects who participated in the experiments are university students/staff recruited randomly.

In the first trial, we investigated the participants' perceived presence and VR sickness. It was a quantitative study in which participants had to walk in the environment and fill out two questions about their sensations of sickness and presence. We employed the "Dynamic" version of the path creation algorithm in the second and third studies. In these tests, we tested our hypothesis that the size of the room had no effect on the overall user experience in the virtual environment. Each subject had to examine the identical setting set up in two different room sizes for the second investigation, which was qualitative. The respondent was individually questioned following each exploration phase. Similar to the second study, the third study was quantitative. The participant was required to complete various survey forms at the conclusion of each exploration round to score their perception of sickness, presence, and how naturally they moved.

We used the Oculus Quest-1 HMD in study-1 and Quest-2 HMD in study-2 and 3 for running the experiments. These are wireless VR headsets with 6-DoF tracking capability. Quest-1 includes a display resolution of 1440 x 1600 pixels per eye, 72Hz refresh rate, Qualcomm Snapdragon 835 processor and 4GB RAM. Quest-2 includes a display resolution of 1832x1920 per eye, 72Hz refresh rate, Qualcomm Snapdragon XR2 processor and a 6GB RAM.

7.1 Questionnaires

7.1.1 Simulator Sickness Questionnaire (SSQ)

The Simulator Sickness Questionnaire (SSQ) is employed to measure the sickness brought on by VR settings. On a scale from 0 (no perception) to 3 (severe perception), participants are asked to rate the subjective intensity of 16 symptoms. Three categories are created from the ratings for the individual symptoms: nausea, oculomotor disturbance, and disorientation. Each category's score is calculated by adding the scores for all of its symptoms and multiplying that result by a fixed scaling factor. Additionally, the three sub-scales are added together to create a total simulator sickness score. Higher ratings on each scale, in general, suggest stronger impressions of the underlying illness symptoms and are thus undesirable. Based on a large sample of SSQ data gathered from military pilots, it is suggested that a simulator resulting in total scores above 20 is considered bad [27]. However, the thresholds vary a lot across different groups of participants [6]. Due to this reason, we have not used any multipliers in our studies.

7.1.2 iGroup Presence Questionnaire (IPQ)

The IPQ is a fourteen-item questionnaire with three subscales and one additional general item. This general item assesses the general sense of being there. The items are grouped into three subscales - spatial presence, involvement, and experienced realism. The spatial presence assesses the sense of being physically present in the virtual environment. The involvement subscale measures the attention devoted to the VR and experienced involvement. The experienced realism scale measures the subjective experience of realism in the virtual environment. Each item is a seven-point-scale from zero to six. In our work, we used a five-point-scale.

7.1.3 Custom Questionnaire (CQ)

The custom questionnaire was prepared by us. It is majorly inspired by a similar questionnaire used in [39]. It recorded the following parameters:

- perceived awareness of the user's position in the physical area
- perceived naturalness of the environment and locomotion technique
- perceived comfort
- perceived safety

Apart from the questions related to the above parameters, we also provided an option to type in feedback and suggestions.

7.2 Preliminary User Study 1

7.2.1 Aim

The aim of this user experience study was to understand our implementation's caveats, immersion, and ease-of-locomotion.

7.2.2 Experiment Setup

The virtual environment was setup in a 24 feet x 17 feet room space at our university campus.

We used the Oculus Quest-1 HMD. As part of our VR scene, 20 images were randomly displayed on the walls in the virtual art gallery. The set of images was repeated until the termination of the environment.

7.2.3 Participants

Among the participants, there were six males and four females. The mean age was 24.5 years. Before undergoing the study, we asked them a few questions to understand their prior exposure to VR. We observed that 35% of the participants had used video/computer games regularly, and 29% had prior VR experience.

7.2.4 Tasks

After obtaining participant consent and educating them about VR induced sickness, we asked them to complete the following tasks in the given order:

- **Demographic Survey**[1]: Participants were asked fill out data regarding age, gender, Gaming experience, and VR experience.

- **Exploration Task**: Participants had to locomote in the virtual art gallery VR Scene for about 5 minutes. Post 5 minutes, the scene was terminated externally by the experimenter after alerting the participant verbally.

- **Questionnaires**: Participants were requested to fill Simulator Sickness Questionnaire(SSQ) [22] to understand simulation sickness while navigating the algorithm generated path, Igroup Presence Questionnaire (IPQ) [2] to understand the participants' presence experience in the scene generated by the algorithm.

7.2.5 Results

Table 7.1 provides the results of our survey study. We observed that the subjects had a good understanding of spatial presence and sense of being in the virtual reality scene despite automatically generating the subject's path in the virtual environment. The subjects experienced some disorientation and minor nausea levels. There is a reasonable amount of involvement of participant while navigating in the VR scene.

[1]https://forms.gle/CW5kAWjAz7oTsyb26

Table 7.1 Mean and Weighted % of User Experience of Virtual Art Gallery

Factors	Mean	%Weighted Response
Nausea	1.3	6.19%
Oculomotor	3.2	15.2%
Disorientation	2.8	13.3%
Sense of being	5.2	74.2%
Spatial Presence	25.9	74%
Involvement	18.7	66%
Experienced Realism	14.8	52%
Presence	64.6	65%

7.3 Preliminary User Study 2

In this user study, we have used the revised implementation of the path generation algorithm (discussed in the chapter 6).

7.3.1 Aim

Our aim was to understand the user locomotion experience in terms of overall comfort and naturalness, and gather feedback/suggestions. Our hypotheses were:

- H1: Walking on the generated path is comfortable and natural.
- H2: The locomotion experience is unaffected by the size of the room in which the path is being generated.

Here comfortable means, the user is able to walk without feeling sick. Here natural means, the user felt that the locomoting in the environment was similar to walking in the real world. Also, the user felt immersed in the environment.

7.3.2 Experiment Setup

We set up two physical rooms: Room-A of 20 feet X 20 feet area and Room-B was 15 feet x 15 feet area at our university campus to run the virtual art gallery scene. The path width in room-A was 1.2 meters and in room-B was 1 meter. We used the enhancements in both the rooms. The minimum path length in room-A was 1m and in room-B was 0.8 meter. We used Oculus Quest 2 HMD. 60 images were presented in the virtual art gallery in random order. The image set was repeated till the termination of the environment. The data collection was done though qualitative interviews. The coordinates of the generated path were also logged for each participant.

7.3.3 Participants

Among the participants, there were ten males and three females. The mean age was 21.1 years. Before undergoing the study, we asked them a few questions to understand their prior exposure to VR. 4 out of 7 was the mean score for the question "How often do you play video/computer games" and 2.9 out of 7 was the mean score for the question "Have you experienced virtual reality before? How often?"

7.3.4 Tasks

Each participant did two rounds of the study, one in each room. To counter balance the carry-over effect of starting from a specific room, we divided the participants into two groups, 7 participants were first sent to the room-A and then then to room-B. The remaining six participants were sent to room-B first.

In the first round, after obtaining participant's consent and educating them about VR induced sickness, we asked them to fill the Demographic Survey form[2] in which they were asked to fill out data regarding age, gender, Gaming experience and VR experience. In both the rounds, participants had to complete the following tasks in the given order:

- **Exploration Task**: Participants had to walk in the virtual art gallery VR Scene for about 3 minutes. Post 3 minutes, the scene was terminated externally by the experimenter after alerting the participant verbally.

- **Qualitative interview**: Participants were asked to verbally give us feedback of their overall experience of walking in the environment. The interviewer asked questions regarding the comfort of locomotion in the environment, the experience of realism and presence while walking. The conversation was voice recorded on a mobile phone after consent from the participant.

7.3.5 Results

We have categorised the collected feedback into three categories: Naturalness and comfort of locomotion; Spatial awareness; Impact of room-size on the subject's performance.

7.3.5.1 Naturalness and comfort of locomotion

- All the participants except one, reported that they were able to walk normally and the environment did not impact their walking behaviour.

- One participant reported they walked cautiously and unnaturally because of abrupt turns.

- A few participants suggested having curved turns to make the environment more comfortable.

[2]https://forms.gle/CW5kAWjAz7oTsyb26

7.3.5.2 Spatial awareness

Just after a participant completed an exploration task, we asked them to estimate their position in the real world before removing the VR headset. They had to report their position in reference to the door though which they entered the room.

Only one participant was aware of their physical location in the physical room. This was probably because of some noise coming from the entrance which might have contributed to a better sense of space.

7.3.5.3 Impact of room-size on performance

- Most of the participants were able to distinguish between the small and large play areas. This was probably because of the different path widths and lengths in both the rooms.

- Most of the participants reported that in the small room (room-B), the path width was less.

- Most of the participants felt that in the large room (room-A) the path length was less.

We think that the above results were satisfactory in terms of comfort and naturalness of locomotion. Also, the participants were not very aware of their position in the real world which means that the experience was immersive. But, the participants were able to differentiate between the room-sizes. This might be because of a design mistake in this experiment. The path dimensions in Room-A were different from Room-B. Ideally, to correctly measure the impact of room size, we should have kept the path dimensions constant. We also observed that the 15ftx15ft room (room B) was too small for accommodating the generated environment.

7.4 Final User Study

7.4.1 Aim

The aim is same as the user study 2. In this experiment, the study design is fine-tuned based on the feedback and flaws in the previous experiment.

7.4.2 Experiment Setup

A few changes were done to the setup of the experiment-2. The room dimensions were changed. In this experiment Room-A was of 25ft x 25ft and Room-B was 20ft x 20ft. We wanted to add a third room for more trustful results but due to logistical issues, we could not do it. Unlike experiment-2, the path properties were constant in both the room. Path width was 1m and minimum path length was 1.2m.

7.4.3 Participants

All the subjects who participated in this study were university students/staff recruited randomly. Among the participants, there were eighteen males and three females. The mean age was 21.5 years. Before undergoing the study, we asked them a few questions to understand their prior exposure to VR. 2.3 out of 5 was the mean score for the question "How often do you play video/computer games" and 2.0 out of 5 was the mean score for the question "Have you experienced virtual reality before? How often?"

7.4.4 Tasks

Each participant did two rounds of the study, one in each room. To counter balance the carry-over effect of starting from a specific room, we divided the participants into two groups, 10 participants were first sent to the room-A and then then to room-B. The remaining six participants were sent to room-B first.

In the first round, after obtaining participant's consent and educating them about VR induced sickness, we asked them to fill the Demographic Survey form[3] in which they were asked to fill out data regarding age, gender, Gaming experience and VR experience. In both the rounds, participants had to complete the following tasks in the given order:

- **Pre-exposure SSQ**: Participant had to fill the Simulator Sickness Questionnaire (SSQ) [22] to report the pre-exposure simulator sickness.

- **Exploration Task**: Participant had to walk in the virtual art gallery VR Scene for about 3 minutes. Post 3 minutes, the scene was terminated externally by the experimenter after alerting the participant verbally.

- **Post-exposure SSQ**: Participants had to fill Simulator Sickness Questionnaire (SSQ) [22] to report the simulation sickness happened due to their experience in the virtual environment,

- **Presence Questionnaire**: Participant had to fill the Igroup Presence Questionnaire (IPQ) [2] to report their experience of presence during the exploration task.

- **Custom Questionnaire**: Participant had to fill a questionnaire[4] prepared by us to record their overall experience with comfort and naturalness of walking in the environment. They can also type-in feedback and suggestions.

[3]https://forms.gle/wqN7927HuKdzeoGu7
[4]https://forms.gle/67rwTGgsZHQzJfwAA

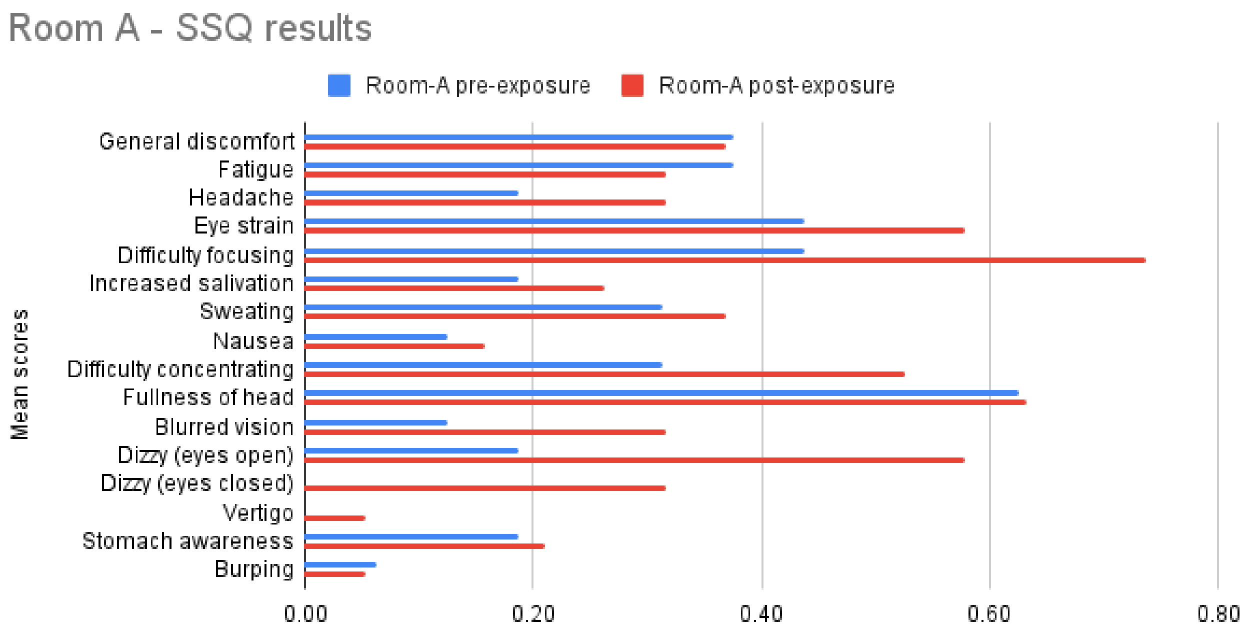

Figure 7.1 Simulator Sickness Questionnaire results

7.4.5 Results

7.4.5.1 Simulator Sickness Questionnaire

We ran Wilcoxon Signed-Rank Test on the results to find if there is any significant difference found between pre and post exposure simulation sickness results. We found no significant difference in all the components of the SSQ. This means that the path generation system generated environment induced no sickness in the participants. The mean scores are summarised in the table 7.2. The mean values are plotted in the figure 7.1 and figure 7.2.

7.4.5.2 Presence Questionnaire

We ran Wilcoxon signed-rank test. No significant differences were found in the scores between room A vs room B. However, in the 7.3, we can observe an overall low mean score on realism experienced by the participants. This indicates that the environment didn't seem to be very realistic to the participants.

IPQ scores are summarised in the table 7.3.

7.4.5.3 Custom Questionnaire

The custom questionnaire consisted of 6 questions - 4 question with a 5-point score scale and 2 multiple choice questions (refer to 7.5. We found no significant differences in the scores of question

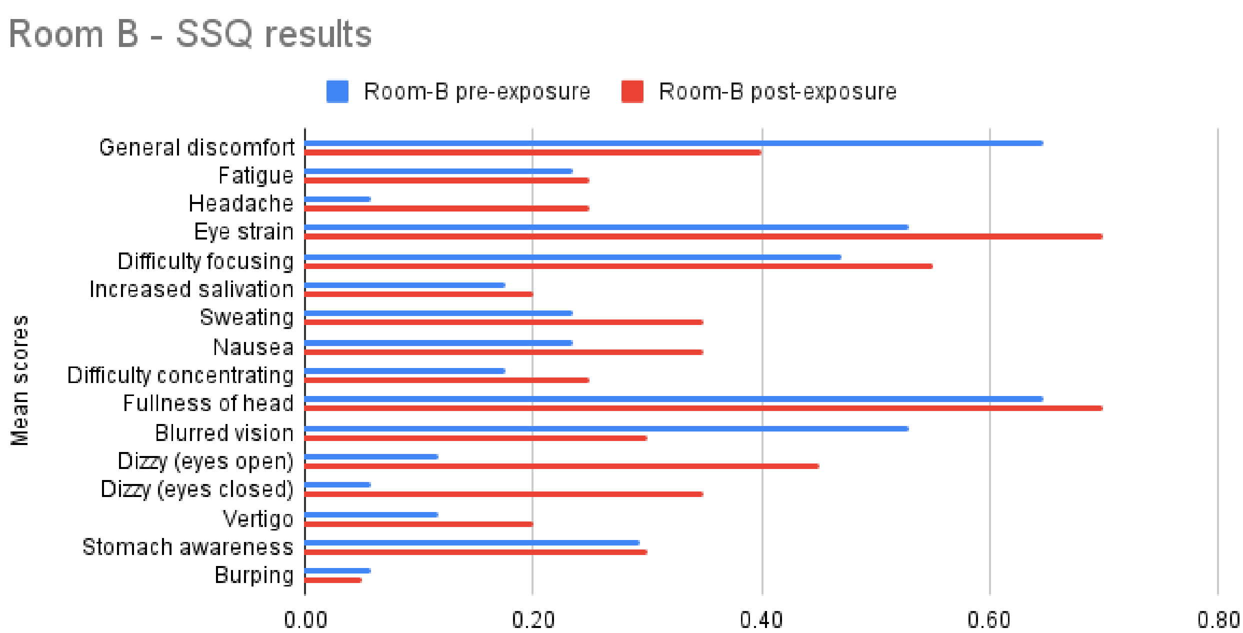

Figure 7.2 Simulator Sickness Questionnaire results

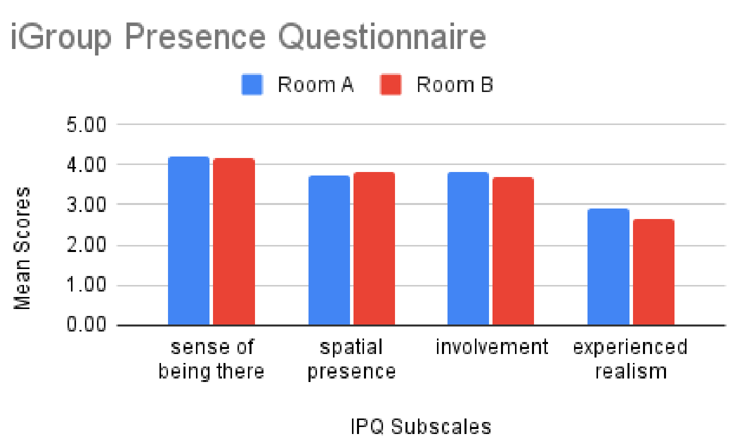

Figure 7.3 iGroup Presence Questionnaire results

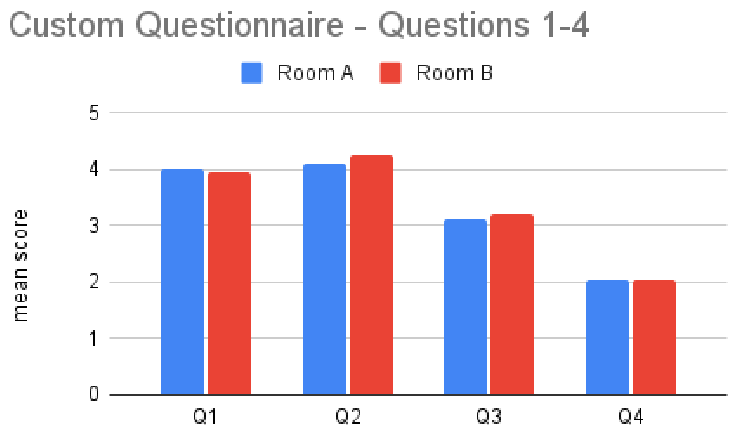

Figure 7.4 Custom Questionnaire - questions 1-4 results

1 to question 4 (refer to figure 7.4 and table 7.4) when compared between room A and room B. High mean scores ($>= 4$) on Q1 and Q2 represent that most of the participant felt that they were able to walk normally in the environment. A low score (< 2.5) in Q4 represents that participants were not aware of their position in the room. Which is a good sign for us, as it indicates that the topology of the generated path didn't give clues about the topology of the play-area. A slightly higher score (> 3) in question 3 indicates that some participants were able to notice the overt manipulations happening in the environment. Though we were expecting a higher score in question 3 because our technique is an overt manipulation technique, the current score can be considered as a positive outcome.

In Q5 (refer to figure 7.5 and table 7.4), we found highest votes (7 out of 19) for each of the two options, (I) "I walked across the whole room, covering most of the room space while walking in the environment" and III "I mostly walked in circle".

In Q6 (refer to figure 7.6 and table 7.4), option VII - "Sometimes, I could see the end of the path" has got the 7 votes and in room B, option VI - "The path seemed to be complete and unending" has 9 votes. Room A was bigger in area than room B, but still participants reported that they were able to see the end of the path in room A more than in room B. However, if we look at the combined results of the options VI and VII, we are assured that participants only sometimes saw the end of path which is acceptable given the overt nature of the environmental manipulations. This also might be the cause of overall low experienced realism observed in the presence questionnaire (refer to 7.3).

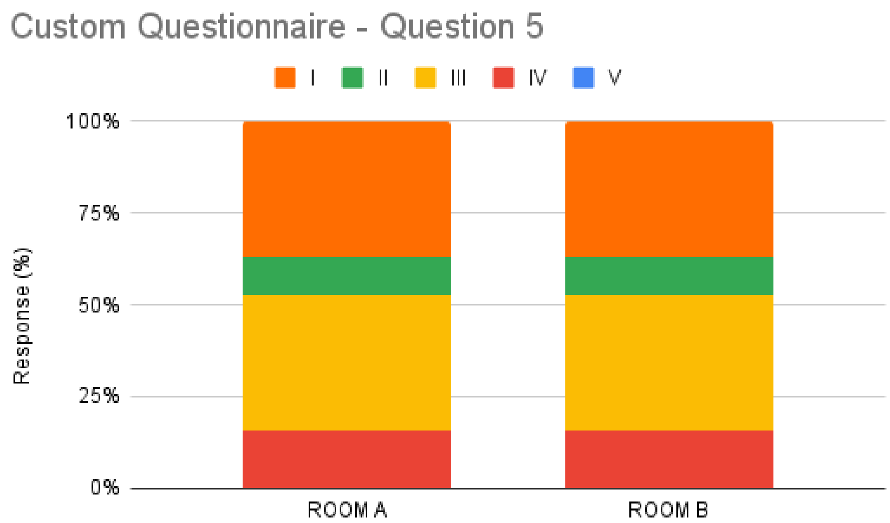

Figure 7.5 Custom Questionnaire - questions 5 results (refer to the table 7.4 to interpret the results)

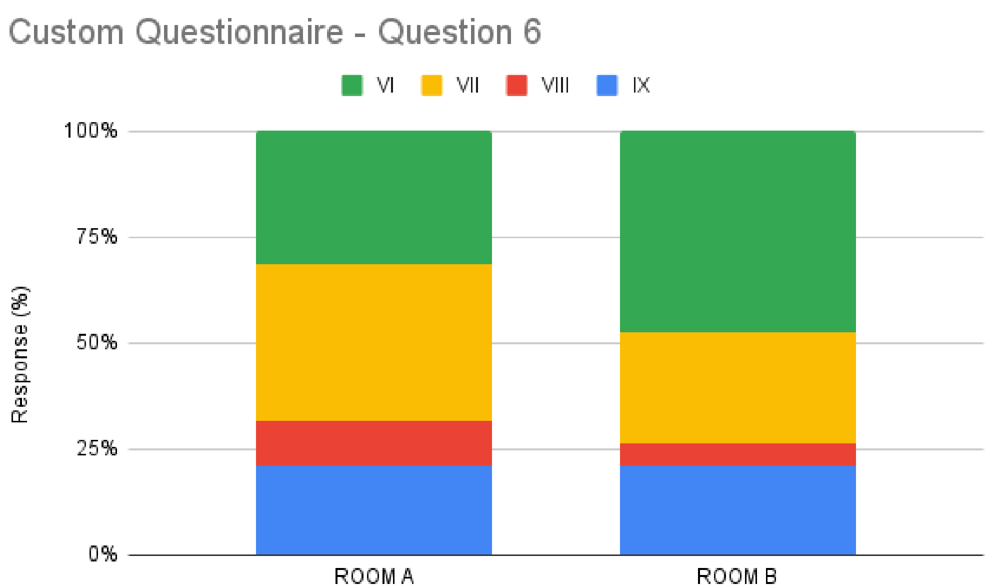

Figure 7.6 Custom Questionnaire - questions 6 results (refer to the table 7.4 to interpret the results)

	Room A		Room B	
Component	**Pre**	**Post**	**Pre**	**Post**
General discomfort	0.38	0.37	0.65	0.40
Fatigue	0.38	0.32	0.24	0.25
Headache	0.19	0.32	0.06	0.25
Eye strain	0.44	0.58	0.53	0.70
Difficulty focusing	0.44	0.74	0.47	0.55
Increased salivation	0.19	0.26	0.18	0.20
Sweating	0.31	0.37	0.24	0.35
Nausea	0.13	0.16	0.24	0.35
Difficulty concentrating	0.31	0.53	0.18	0.25
Fullness of head	0.63	0.63	0.65	0.70
Blurred vision	0.13	0.32	0.53	0.30
Dizzy (eyes open)	0.19	0.58	0.12	0.45
Dizzy (eyes closed)	0.00	0.32	0.06	0.35
Vertigo	0.00	0.05	0.12	0.20
Stomach awareness	0.19	0.21	0.29	0.30
Burping	0.06	0.05	0.06	0.05

Table 7.2 Pre-exposure and Post-exposure mean scores obtained from Simulator Sickness Questionnaire.

	Mean(SD)	
	Room A	**Room B**
Sense of being there	4.22(1.11)	4.17(0.92)
Spatial presence	3.72(0.55)	3.81(0.80)
Involvement	3.81(0.76)	3.71(0.72)
Experienced realism	2.90(0.63)	2.67(0.55)
Presence	3.55(0.54)	3.48(0.54)

Table 7.3 Mean and standard deviation of IPQ sub-scales.

	Label	Room A	Room B
Question 1 to Question 4		**Mean(SD)**	
The way I walked in the VR reflects the way I would normally walk in real life	**Q1**	4(1.29)	3.95(1.27)
The way I appeared to move in VR is the same as the way I was actually moving	**Q2**	4.11(1.10)	4.26(1.05)
I noticed how this walking technique manipulated the environment to allow me to continue moving	**Q3**	3.11(1.73)	3.21(1.47)
While I walked in VR, I was aware of where I was in the room	**Q4**	2.05(1.22)	2.05(1.35)
Question 5		**Number of responses**	
I walked across the whole room, covering most of the room space while walking in the environment	**I**	7	7
I mostly walked around the parameter of the room	**II**	2	2
I mostly walked in circles	**III**	7	7
I mostly walked in a small portion of the room space while walking in the environment	**IV**	3	3
I did not walk at all	**V**	0	0
Question 6			
The path seemed to be complete and unending	**VI**	6	9
Sometimes, I could see the end of the path	**VII**	7	5
Most of the times I could see the end of the path	**VIII**	2	1
I always noticed that the path was incomplete	**IX**	4	4

Table 7.4 Custom Questionnaire Results

Question	Input type and labels
Q1. The way I walked in the VR reflects the way I would normally walk in real life	scale: 1-5; (1)I walked very differently in VR; (5)I walked exactly the same
Q2. The way I appeared to move in VR is the same as the way I was actually moving	scale: 1-5; (1)I think the way I appeared to be moving in VR does not reflect at all the movements I physically made; (5)I think the way I appeared to be moving in VR exactly reflect movements I physically made
Q3. I noticed how this walking technique manipulated the environment to allow me to continue moving	scale: 1-5; (1)I did not notice any manipulation; (5)I was very aware of the manipulation
Q4. While I walked in VR, I was aware of where I was in the room	scale: 1-5; (1)I had no idea of where I was in the real room; (5)I was always aware of where I was in the real room
Q5. Please select the answer that most closely represents the way you walked	choose 1 out of 5 options; (1)I walked across the whole room, covering most of the room space while walking in the environment; (2)I mostly walked around the parameter of the room; (3)I mostly walked in circles; (4)I mostly walked in a small portion of the room space while walking in the environment; (5)I did not walk at all
Q6. While walking in the environment	choose 1 out of 4 options; (1)the path seemed to be complete and unending; (2)Sometimes, I could see the end of the path; (3)Most of the times I could see the end of the path; (4)I always noticed that the path was incomplete
Q7. Suggestions/Feedback/Anything else you want to tell	Long-answer textbox

Table 7.5 Custom questionnaire items.

Chapter 8

Conclusion and Future Work

The following section concludes and summarises the results from the research work presented and the contributions made to the existing knowledge base related to the field of locomotion in virtual reality.

The inside-out tracking technology and lightweight wireless virtual reality (VR) devices have opened multiple avenues for locomotion techniques in VR. With aggressively increasing VR applications, the requirement for novel locomotion techniques is more than ever. No locomotion technique provides an experience precisely similar to natural walking, but this is what creates a space for innovations. Many researchers have proposed locomotion techniques by building external hardware or developing software. However, hardware-based methods are often rigid, less mobile, expensive or inaccessible. On the other hand, software-based techniques have limitations on space and computational requirements. Some methods require specific technologies, while others are heavily dependent on computational resources.

Among the first few, our contribution focuses on the new generation of VR devices with long-range 6-DoF tracking and high mobility. We believe our contribution has given a new direction to the locomotion techniques development approach, which is overt, software-oriented and customisable.

8.1 Contributions

- An algorithm which dynamically generates a continuous path within a limited/constrained space.
- A system which can be used to generate an environment employing the limitless path generation algorithm.
- Developed a use-case “Virtual Art Gallery” to demonstrate the potential of our work.
- Validated our approach through multiple user studies.

8.2 Limitations

- During the user studies, we have found that our approach requires a minimum play area of 20 feet by 20 feet to function normally.
- The limitless path generation system is only compatible with devices which support Oculus VR SDK.
- The algorithm requires a predefined play area in the virtual space. For example, a guardian should exist in Oculus headsets before the path generation system is running.

8.3 Future work

There is a scope for enhancing the path generation algorithm further in the following ways:

- by adding an ability to generate curved paths instead of edge paths.
- generates paths which consist of junctions so that the user can choose the direction for moving forward.

There is a scope for adding more features to the path generation system in the following ways:

- Current system only provides static images placed in the gallery. One can develop the system further by adding interactive items to the gallery.
- One should explore an alternative implementation of our algorithm for generating limitless paths in non-euclidean environments. It would allow covert manipulations in the environment.

Our work is among a rare class of locomotion techniques called overt redirection techniques. When we started, we had not enough prior literature available. It will be a significant contribution if one rethinks our approach from a fresh perspective, which may bring better methodologies for limitless path generation.

Publications

8.4 Relevant Publications

- **Raghav Mittal**, Sai Anirudh Karre, Y Pawankumar Gururaj, Y. Raghu Reddy, **Enhancing Configurable Limitless Paths in Virtual Reality Environments**, In proceedings of the *15th Innovations in Software Engineering Conference 2022*, Gandhinagar, India, February 2022.

- **Raghav Mittal**, Sai Anirudh Karre, Y. Raghu Reddy **Designing Limitless Path in Virtual Reality Environment**, In proceedings of *Virtual, Augmented and Mixed Reality - 13th International Conference, VAMR 2021, Held as Part of the 23rd HCI International Conference, HCII 2021*, July 24-29, 2021, Proceedings. Lecture Notes in Computer Science 12770, Springer 2021, pp: 80-95, ISBN 978-3-030-77598-8.

8.5 Patent (Applied)

- Y. Raghu Reddy, **Raghav Mittal**, Sai Anirudh Karre **System And Method For Generating A Limitless Path In Virtual Reality Environment For Continuous Locomotion**. Filing/Priority Date: May 28, 2021.

8.6 Other Publication

- Sai Anirudh Karre, **Raghav Mittal** and Y. Raghu Reddy, **Requirements Elicitation for Virtual Reality Products - A Mapping Study**, In proceedings of the *16th Innovations in Software Engineering Conference 2023*, Prayagraj, India, February 2023.

- Sai Anirudh Karre, Vivek Pareek, **Raghav Mittal**, Y. Raghu Reddy **A Role Based Model Template for Specifying Virtual Reality Software**, In proceedings *International Workshop on Virtual and Augmented Reality Software Engineering, in conjunction with Automated Software Engineering (ASE 2022)*, Mon 10 - Fri 14 October 2022 Oakland Center, Michigan, United States.

- Y Pawankumar Gururaj, **Raghav Mittal**, Sai Anirudh Karre, Y. Raghu Reddy, Syed Azeemuddin, **Towards Conducting Effective Locomotion Through Hardware Transformation in Head-**

Mounted-Device - A Review Study [POSTER], In proceedings of *IEEE Virtual Reality Conference (IEEEVR 2022)*, 12-16 March 2022, Virtual.

- Y Pawankumar Gururaj, Sai Anirudh Karre, **Raghav Mittal**, Y. Raghu Reddy, Syed Azeemuddin, **Customizable Head-mounted Device for Detection of Eye Disorders using Virtual Reality**, In proceedings of *35th International Conference on VLSI Design and 19th International Conference on Embedded Systems, VLSID 2022*, Virtual, 20th - 24th Feb, 2022. IEEE 2022.

Bibliography

[1] Holoride - adding thrill to every ride — https://www.holoride.com/.

[2] Igroup presence questionnaire — http://www.igroup.org/pq/ipq/. Accessed: 2021-02-04.

[3] Virtual reality (vr) market share, growth: Research report, 2028 — https://www.fortunebusinessinsights.com/industry-reports/virtual-reality-market-101378.

[4] T. Alsop. Vr headset unit sales worldwide 2024.

[5] C. Anthes, R. J. Garca-Hernndez, M. Wiedemann, and D. Kranzlmller. State of the art of virtual reality technology. In *2016 IEEE Aerospace Conference*, pages 1–19, 2016.

[6] P. Bimberg, T. Weissker, and A. Kulik. On the usage of the simulator sickness questionnaire for virtual reality research. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*, pages 464–467, 2020.

[7] G. Bishop and H. Fuchs. Research directions in virtual environments: report of an nsf invitational workshop, march 23-24, 1992, university of north carolina at chapel hill. In *COMG*, 1992.

[8] R. CARTER. Vr headset tech: Its more affordable and better than ever, Nov 2020.

[9] P. Cipresso, I. A. C. Giglioli, M. A. Raya, and G. Riva. The past, present, and future of virtual and augmented reality research: A network and cluster analysis of the literature. *Frontiers in Psychology*, 9:2086, 2018.

[10] C. Cruz-Neira. Virtual reality overview. In *Siggraph*, page 2, 1993.

[11] M. Di Luca, H. Seifi, S. Egan, and M. Gonzalez-Franco. Locomotion vault: The extra mile in analyzing vr locomotion techniques. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.

[12] N. Dumaska, P. Strojny, and A. Strojny. Can simulator sickness be avoided? a review on temporal aspects of simulator sickness. *Frontiers in Psychology*, 9, 2018.

[13] F. A. D. Farías. Six dof tracking system based on smartphones internal sensors for standalone mobile vr. 2019.

[14] T. Feigl, E. Kõre, C. Mutschler, and M. Philippsen. Acoustical manipulation for redirected walking. VRST '17, New York, NY, USA, 2017. Association for Computing Machinery.

[15] S. Freitag, D. Rausch, and T. Kuhlen. Reorientation in virtual environments using interactive portals. In *2014 IEEE symposium on 3D user interfaces (3DUI)*, pages 119–122. IEEE, 2014.

[16] M. Funk, F. Müller, M. Fendrich, M. Shene, M. Kolvenbach, N. Dobbertin, S. Günther, and M. Mühlhäuser. Assessing the accuracy of point amp; teleport locomotion with orientation indication for virtual reality using curved trajectories. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, CHI '19, page 112, New York, NY, USA, 2019. Association for Computing Machinery.

[17] A. Garg, J. A. Fisher, W. Wang, and K. P. Singh. Ares: An application of impossible spaces for natural locomotion in vr. CHI EA '17, page 218221, New York, NY, USA, 2017. Association for Computing Machinery.

[18] T. Grechkin, J. Thomas, M. Azmandian, M. Bolas, and E. Suma. Revisiting detection thresholds for redirected walking: Combining translation and curvature gains. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '16, page 113120, New York, NY, USA, 2016. Association for Computing Machinery.

[19] V. Interrante, B. Ries, and L. Anderson. Seven league boots: A new metaphor for augmented locomotion through moderately large scale immersive virtual environments. In *2007 IEEE Symposium on 3D User Interfaces*, 2007.

[20] J. Jerald, T. Peck, F. Steinicke, and M. Whitton. Sensitivity to scene motion for phases of head yaws. APGV '08, page 155162, New York, NY, USA, 2008. Association for Computing Machinery.

[21] G. Jijiashvili. Omdia research reveals 12.5 million consumer vr headsets sold in 2021 with content spend exceeding $2bn, Dec 2021.

[22] R. S. Kennedy, N. E. Lane, K. S. Berbaum, and M. G. Lilienthal. Simulator sickness questionnaire: An enhanced method for quantifying simulator sickness. *The International Journal of Aviation Psychology*, 3(3):203–220, 1993.

[23] E. Langbehn, T. Eichler, S. Ghose, K. von Luck, G. Bruder, and F. Steinicke. Evaluation of an omnidirectional walking-in-place user interface with virtual locomotion speed scaled by forward leaning angle. In *Proceedings of the GI Workshop on Virtual and Augmented Reality (GI VR/AR)*, pages 149–160, 2015.

[24] E. Langbehn, P. Lubos, and F. Steinicke. Evaluation of locomotion techniques for room-scale vr: Joystick, teleportation, and redirected walking. In *Proceedings of the Virtual Reality International Conference - Laval Virtual*, VRIC '18, New York, NY, USA, 2018. Association for Computing Machinery.

[25] J. Lee, S. C. Ahn, and J.-I. Hwang. A walking-in-place method for virtual reality using position and orientation tracking. *Sensors*, 18(9), 2018.

[26] K. Matsumoto, T. Narumi, Y. Ban, Y. Yanase, T. Tanikawa, and M. Hirose. Unlimited corridor: A visuo-haptic redirection system. VRCAI '19, New York, NY, USA, 2019. Association for Computing Machinery.

[27] T. Mazuryk and M. Gervautz. Virtual reality - history, applications, technology and future.

[28] M. Nabiyouni and D. A. Bowman. A taxonomy for designing walking-based locomotion techniques for virtual reality. In *Proceedings of the 2016 ACM Companion on Interactive Surfaces and Spaces*, ISS '16 Companion, page 115121, New York, NY, USA, 2016. Association for Computing Machinery.

[29] Y. S. Pai and K. Kunze. Armswing: Using arm swings for accessible and immersive navigation in ar/vr spaces. MUM '17, page 189198, New York, NY, USA, 2017. Association for Computing Machinery.

[30] R. Paris, M. Joshi, Q. He, G. Narasimham, T. P. McNamara, and B. Bodenheimer. Acquisition of survey knowledge using walking in place and resetting methods in immersive virtual environments. In *Proceedings of the ACM Symposium on Applied Perception*, SAP '17, New York, NY, USA, 2017. Association for Computing Machinery.

[31] E. Pinson, K. Pietroszek, Q. Sun, and C. Eckhardt. An open framework for infinite walking with saccadic redirection. In *26th ACM Symposium on Virtual Reality Software and Technology*, VRST '20, New York, NY, USA, 2020. Association for Computing Machinery.

[32] V. A. Pisani, O. Hurd, N. Hawthorne, and S. Kurniawan. Navigation by walking in hyperbolic space using virtual reality. CHI PLAY '19 Extended Abstracts, page 611618, New York, NY, USA, 2019. Association for Computing Machinery.

[33] D. M. Plasencia. One step beyond virtual reality: Connecting past and future developments. *XRDS*, 22(1):1823, Nov. 2015.

[34] A. Prithul, I. B. Adhanom, and E. Folmer. Teleportation in virtual reality; a mini-review. *Frontiers in Virtual Reality*, 2, 2021.

[35] H. Rheingold. *Virtual Reality*. Simon & Schuster, Inc., USA, 1991.

[36] M. Rietzler, M. Deubzer, T. Dreja, and E. Rukzio. Telewalk: Towards free and endless walking in room-scale virtual reality. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 19, New York, NY, USA, 2020. Association for Computing Machinery.

[37] R. A. Ruddle, E. Volkova, and H. H. Bülthoff. Walking improves your cognitive map in environments that are large-scale and large in extent. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 18(2):1–20, 2011.

[38] J. Schlueter, H. Baiotto, M. Hoover, V. Kalivarapu, G. Evans, and E. Winer. Best practices for cross-platform virtual reality development. In *Defense + Security*, 2017.

[39] A. L. Simeone, N. Christian Nilsson, A. Zenner, M. Speicher, and F. Daiber. 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)*, pages 598–606, 2020.

[40] M. Slater and S. Wilbur. A framework for immersive virtual environments five: Speculations on the role of presence in virtual environments. *Presence: Teleoper. Virtual Environ.*, 6(6):603616, Dec. 1997.

[41] F. Steinicke, G. Bruder, K. H. Hinrichs, and A. Steed. Gradual transitions and their effects on presence and distance estimation. *Comput. Graph.*, 34(1):26–33, 2010.

[42] 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.

[43] R. Stoakley, M. J. Conway, and R. Pausch. Virtual reality on a wim: interactive worlds in miniature. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 265–272, 1995.

[44] E. A. Suma, S. Clark, D. Krum, S. Finkelstein, M. Bolas, and Z. Warte. Leveraging change blindness for redirection in virtual environments. In *2011 IEEE Virtual Reality Conference*, pages 159–166, 2011.

[45] E. A. Suma, Z. Lipps, S. Finkelstein, D. M. Krum, and M. Bolas. Impossible spaces: Maximizing natural walking in virtual environments with self-overlapping architecture. *IEEE Transactions on Visualization and Computer Graphics*, 18(4):555–564, 2012.

[46] I. E. Sutherland. The ultimate display. In *Proceedings of the IFIP Congress*, pages 506–508, 1965.

[47] S. Tregillus and E. Folmer. Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environments. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, page 12501255, New York, NY, USA, 2016. Association for Computing Machinery.

[48] S. Truman and S. von Mammen. An integrated design of world-in-miniature navigation in virtual reality. In *International Conference on the Foundations of Digital Games*, FDG '20, New York, NY, USA, 2020. Association for Computing Machinery.

[49] M. Usoh, K. Arthur, M. Whitton, R. Bastos, A. Steed, M. Slater, and P. B. Frederick. J. 1999. walking > walking-in-place > flying, in virtual environments. In *Proc. ACM SIGGRAPH 1999*.

[50] M. Usoh, K. Arthur, M. C. Whitton, R. Bastos, A. Steed, M. Slater, and F. P. Brooks. Walking >walking-in-place >flying, in virtual environments. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, page 359364, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[51] E. Vagner. What does dof mean in vr? — motive.io, Aug 2021.

[52] K. Vasylevska, H. Kaufmann, M. Bolas, and E. A. Suma. Flexible spaces: Dynamic layout generation for infinite walking in virtual environments. In *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 39–42, 2013.

[53] W. Wakita, T. Takano, and T. Hadama. A low-cost omni-directional vr walking platform by thigh supporting and motion estimation. In *Proceedings of the 24th ACM Symposium on Virtual Reality Software and Technology*, VRST '18, New York, NY, USA, 2018. Association for Computing Machinery.

[54] P. T. Wilson, K. Nguyen, A. Harris, and B. Williams. Walking in place using the microsoft kinect to explore a large ve. VRCAI '14, page 2733, New York, NY, USA, 2014. Association for Computing Machinery.