Bachelor thesis

Degree in Informatics Engineering
Specialization in Computing

Study and development of navigation techniques for virtual reality

Author:
Sergi Ávila Sangüesa

Thesis supervisor
Nuria Pelechano Gomez (Computer Science)
Thesis co-supervisor
Alejandro Ríos Jerez (Computer Science)

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

ViRVIG
Visualització, Realitat Virtual i Interacció Gràfica

Defense date: 28/1/2020

Contents

Resum

La navegació en un entorn virtual és actualment una de les tasques més complicades. Real walk (caminar natural) és la millor interfície, però només és possible si l'entorn virtual i l'espai físic son de mides semblants. La majoria de sistemes amb navegació per grans escenaris provoquen marejos o desorientació. Walk in Place (caminar al lloc) és una tècnica en què l'usuari es mou de forma semblant al caminar natural però sense avançar físicament, de forma que es redueix el mareig. Quan el moviment generat pel sistema no és mateix que l'usuari espera, l'usuari es pot frustrar o marejar.

Aquest document comença mostrant millores sobre un sistema WiP ja existent. Tot seguit mostra una expansió als sistemes actual permetent un moviment ràpid, anomenat Run in Place (córrer al lloc) i un lent, de forma que es permet interactuar amb l'entorn de noves formes. Després es descriu una fase d'entrenament per tal d'obtenir informació especifica de cada usuari que serveix al sistema per adaptar-se als moviments de l'usuari i poder generar una velocitat de sortida millor ajustada a cada persona.

Finalment, s'explica un experiment fet amb diversos usuaris per obtenir informació del moviment dels peus durant WiP amb diferents velocitats de càmera, creant les bases per desenvolupar un nou mètode de WiP.

Abstract

In Virtual Reality navigating through an environment is currently one of the most challenging problems. Real Walking is the preferred interface, but it is only feasible when the virtual environment is similar to the physical space available. Most systems that allow movement in large scenes cause either motion sickness or disorientation. Walk in Place (WiP) is a technique that reduces motion sickness in a virtual environment by introducing user gestures similar to real walk but that does not result in a physical displacement. When the system generated forward movement does not match the user's expected output, the user can become frustrated or feel dizziness.

This document first presents improvements over an existing WiP system. Then presents an expansion to current WiP system by creating a system that allows a fast movement called Running in Place and a slow WiP movement, allowing for different ways to interact with the environment. After that the document explains the design of a training stage that uses user-specific data to prepare the system for the user's inputs in order to better match the expected virtual speeds.

It finally lays the groundwork to developing a new WiP method by showing a experiment with different users to collect feet data from performing WiP at different speeds

Paraules clau

realitat virtual, caminar, córrer, navegació, entorn virtual

Keywords VR, virtual reality, WiP, walk in place, run in place, walk, run, navigation, virtual environment

1 Introduction

1.1 Virtual Reality

Virtual reality, or VR for short, allows the user to interact with a 3D computer generated environment. VR offers an immersive or semi-immersive experience as opposed to simply having a screen monitor and interacting with keyboard, mouse or joystick. VR usually attempts to simulate as many senses as possible with the intention to make the user feel as if he/she were in a real environment. In contrast to Augmented Reality, which overlaps information or graphics with the real world, in VR everything the user experiences and sees is computer generated.

Currently, the most popular equipment for VR setups is a Head Mounted Display (HMD). This device is worn on the head, similar to a helmet, and has a screen located right in front of the user's eyes. The screen is separated in two halves, one for each eye. Each one showing the virtual environment from a slightly different viewpoint, creating the illusion of 3D. Other common devices used alongside the HMD include headphones, that may be integrated in the HMD itself, and controllers, usually one for each hand so that the user can interact with the surroundings using each hand independently in a natural manner.

Figure 1: HTC vive HMD and trackers. Images from the VIVE product webpage [HTC]

VR can be applied to a large range of fields. For most people, its main use is entertainment, as it can be used for video games. In medicine and others fields of research it can be used for visualising things that are hard to imagine as a 3D object unless you actually see them, such as the organs of the human body. When you need to build something, for example a car or a building, VR can also help to see how the final product will look like and allows for changes to be made without spending resources on actually building it.

Navigating in VR is probably the most challenging problem when it comes to natural interaction in VR. This is mostly because many navigation methods cause motion sickness. For example, techniques that use some kind of joystick or controller to move the camera, while the user's body remains static, can cause motion sickness due to the inconsistencies between what the user eyes are seeing, the optical flow, and the user proprioceptive system which is aware of the lack of movement. This problem can be lessen by having the user physically moving, ideally with normal walk, but this is not always possible because the virtual world can be much larger than the available physical space. In Walk in Place techniques, WiP, the user moves the legs up and down in order to simulate walking while not actually changing position. WiP techniques manage to provide the proprioceptive awareness of movement while not needing a large physical space. However if the mapping between leg movement and camera movement is not well adjusted, then some users will still experience this undesirable effect.

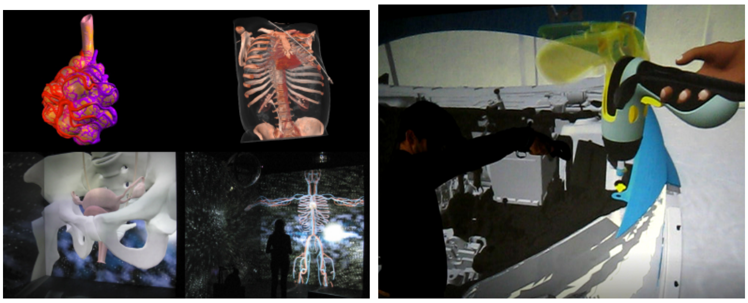


Figure 2: Applications of VR in the medical field and in the assembly process of a car. Images from the ViRVIG webpage [ViR]

1.2 Context and main goal

This project has been developed in the ViRVIG Research Center for Visualization, Virtual Reality and Graphics Interaction, a research center hosted at UPC that works very closely with the FIB, teaching most of the graphics courses in both the bachelor and the masters.

The project started as an extension of the work shown in the paper **Smooth transitioning between two walking metaphors for Virtual Reality applications** written by professors Alejandro Ríos and Nuria Pelechano and Isadora Salvetti, a master student [SRP19].[1] The paper describes a technique to switch between walking normally around a room with the HMD on, and Walking in Place in larger environments.

This project focused on improving the WiP part. Walking in place is very different to performing a normal walk, since the feet move up and down instead of up-forward-down. Therefore, finding a good mapping between the up/down movement of the feet and the movement of the virtual camera is not trivial. Any inconsistencies between what the user expects to observe and how the camera moves, can cause the user to suffer from motion sickness.

Thus the main goal of the project is to research methods to adjust the camera movement to the natural movement of each user in order to improve the immersion and to reduce motion sickness.

In order to adjust the virtual camera movement to each person we decided to

---

[1] Since the paper will be referenced multiple times throughout this document and also for better readability, it will be referred to as "the base paper" from this point on.

study the biomechanics of our gait cycle (e.g. the speed and frequency of our feet when walking). With this new method, we want to allow anyone who wants to develop VR application involving large scenes, to be able to have a navigation method that feels more natural and provides correct proprioceptive cues.

1.3 Project structure

The work done in this project can be divided in 4 separate stages. All of them added new research or functionalities to the previous stages, but they are different enough to deserve a different section in this document.

In the first stage **Improving the original project** the goal was to make small improvements to the original Unity project so that it better reflected the method shown in the base paper. Some internal usability tests with other ViRVIG members were also performed in order to decide what parts of the project were more interesting to research next.

The second stage **Creating new WiP modes** consisted on allowing to Walk in Place at different paces with a virtual flow that matched the user's expectations. Little research has been conducted on slow and fast WiP, which we will call Run in Place or RiP [2], and the base project did not perform well when trying to move at different paces. This stage required to visualize and analyze the data of our gait cycle in order to create a state machine that detected how the user was walking and to create a new function to map the user movement when Running in Place. While gait data was used to develop slow WiP and RiP during this stage, by the end of it the computer was still not able to adapt the movement to each user's specific natural movement.

In the third stage **Designing a training stage** we designed a training stage that adjusts the parameters that decide if the user is performing WiP or RiP. The stage required several tests to decide what parameters to adjust, what biomechanical information was required and how to extract it. In the final project the user first undergoes the training stage and then starts navigating the virtual environment with the adjusted virtual camera movements.

Finally we conducted an experiment to extract feet data from users Walking in Place with different optical flows. The experiments will serve as the groundwork to develop a new WiP method.

1.4 Software,hardware and environment

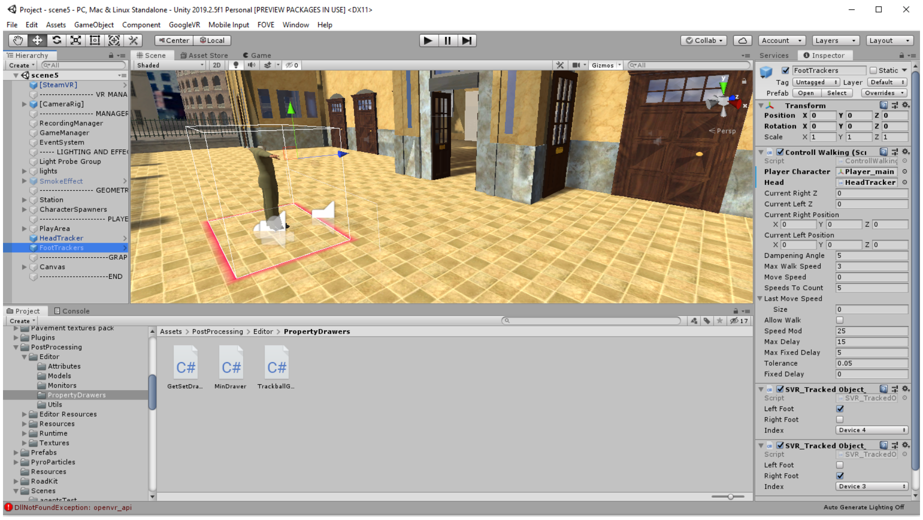
To develop the project, the normal version of the HTC VIVE was used, alongside the bases and the original 2017 version of the trackers. The pc where all the project was developed had and Intel Core i7-4970k CPU, NVIDIA GeForce GTX 970 gpu and a 16GB RAM. PC specifications are important for this project since VR devices have high PC requirements.

---

[2] During the rest of the document, we will use the terms "WiP" and "RiP" when talking about the technique itself, and "Walk in Place" and "Run in Place" when referring to the movement performed by the user in the techniques.

Software-wise, the main program we used is Unity game engine [Unia]. Current game engines offer a set of tools designed to allow users to easily develop any graphical application, including games and VR environment. The tools allow the users to abstract themselves from the code required to do a lot of the most common tasks when working with graphics by providing a simple graphical interface. Some of the tasks that are easy to include in a project are: defining the geometry in a scene, materials, movement, position and properties of lights, and defining interactions between objects. Thanks to all this tools that Unity provides, the programmer can focus exclusively in developing new algorithms and implementing other aspects of the project that are not yet included in the game engine. Unity, as all other engines, provides a development environment in which users can control all aspects of the project from a graphical interface. It also provides a scripting API that allows programmers to write scripts to better define behaviours and interaction between objects. Unity requires the user to program in C#, an object-oriented programming language strongly influenced by Java and C++ and developed by Microsoft in the year 2000.

Finally we also needed to use Steam and Steam VR[Val]. Steam is a digital video game distribution storefront, created by Valve, developers of the HTC Vive. Steam VR is the program needed to use the Vive's HMD and other devices. It controls their state, behaviour and provides programmers with an API to interact with the devices



☑ FootTrackers ☐ Static
Tag: Untagged
Layer: Default
Prefab: Open | Select | Overrides

Transform
Position X 0 Y 0 Z 0
Rotation X 0 Y 0 Z 0
Scale X 1 Y 1 Z 1

☑ **Controll Walking (Sc**
Script: ControllWalking
Player Character: Player_main
Head: HeadTracker
Current Right Z: 0
Current Left Z: 0
Current Right Position: X 0 Y 0 Z 0
Current Left Position: X 0 Y 0 Z 0
Dampening Angle: 5
Max Walk Speed: 3
Move Speed: 0
Speeds To Count: 5
Last Move Speed
Size: 0
Allow Walk: ☐
Speed Mod: 25
Max Delay: 15
Max Fixed Delay: 5
Tolerance: 0.05
Fixed Delay: 0

☑ **SVR_Tracked Object_**
Script: SVR_TrackedO
Left Foot: ☑
Right Foot: ☐
Index: Device 4

☑ **SVR_Tracked Object_**
Script: SVR_TrackedO
Left Foot: ☐
Right Foot: ☑
Index: Device 3

DllNotFoundException: openvr_api

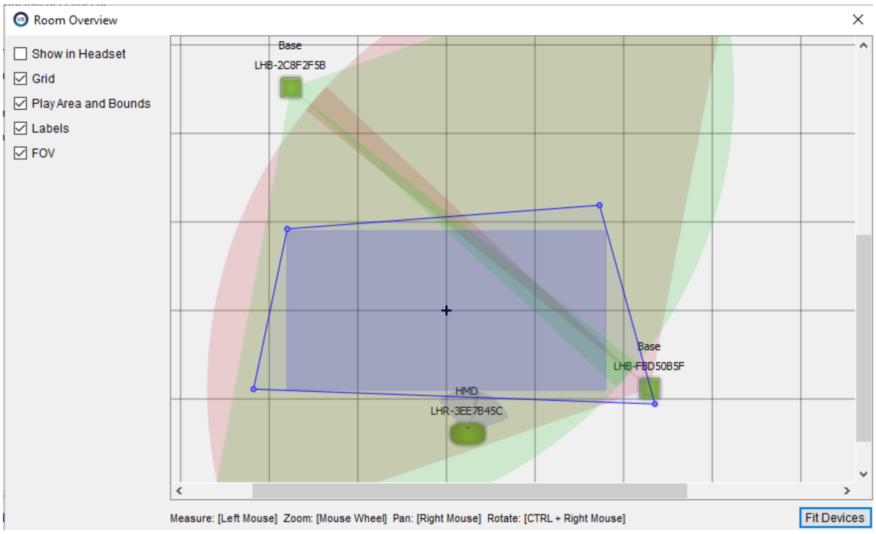
Figure 3: Unity environment with the project opened

The feet movement was captured by the bases using Vive's trackers attached to the feet.

For most of the project, the space available to perform the tests was relatively small, with a distance between the 2 bases of around 5 meters, and a usable area space of 3.5m x 1.7m, limited by walls and tables, figure 4.

The space allowed us to record our movements when walking normally, but it was very small in terms of gathering usable data for analysis, because we needed to discard the acceleration and deceleration phase of a normal walk. Therefore, this limitation left us with only a handful of steps. Those steps, while they were good enough for most internal testing, were too small to perform the final experiment. Therefore in order to carry out the experiment and also gather important recordings from the training stage, we moved the capture area to a larger space.

In this second setup of figure 5 the space between the bases is 7 meters and the usable area 6m x 2m. This larger space allowed us to capture better and more accurate recordings. The only problem during the development of this project, is that this larger room was not always available, since it is shared with other student projects and it is were the ViRVIG shows most of their projects to visitors.



Room Overview

☐ Show in Headset
☑ Grid
☑ Play Area and Bounds
☑ Labels
☑ FOV

Figure 4: Room used during most of the project

Room Overview

☐ Show in Headset
☑ Grid
☑ Play Area and Bounds
☑ Labels
☑ FOV

Figure 5: Room used in the experiments

2 Background and related work

How humans move in a virtual environment is a research topic that has drawn attention in recent years due to the sharp increase in computing power of widely used devices such as computers and mobile phones, which also allowed the development of the current VR devices. The main problem with movement in a virtual environment is that the proprioceptive and vestibular information that we receive with the virtual camera movement does not match the one present in the real world. When we move, by ourselves or in a vehicle, we receive a certain proprioceptive feedback, the sense of self movement and body position, and vestibular feedback, the sense of balance and spatial orientation. Research concerning VR often focuses on reducing or hiding the differences in proprioceptive and vestibular information thereby tricking our brains into reacting the same way it would do when moving in the real world.

2.1 Virtual camera movement methods

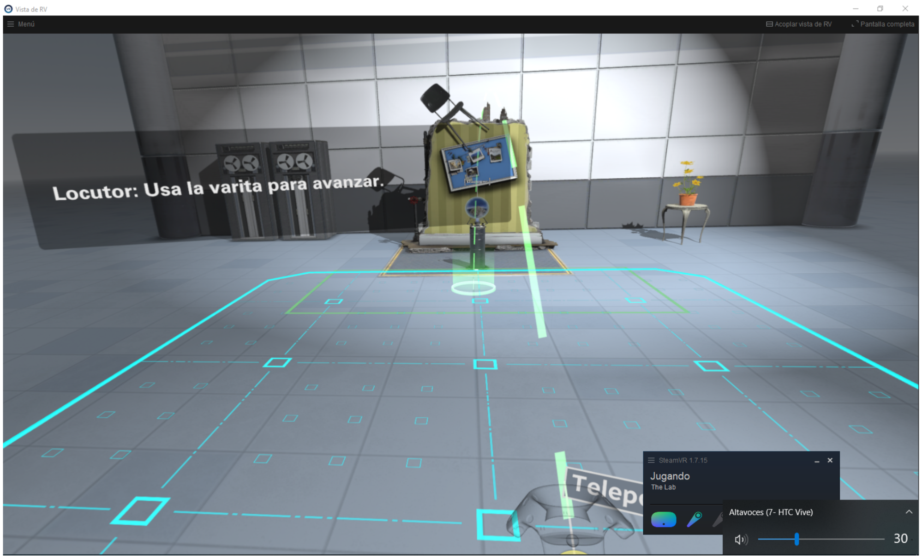
The HTC Vive uses 2 base stations that emit IR lasers and track the user's HMD. This technology allows to define a virtual room with the same size as the real world. Movement within the virtual room can be performed by walking normally in the real room, since the bases can directly map the movements in the real world to the virtual environment. This movement method is the most natural one, offers the highest immersion and the lowest chances of causing motion sickness. However, it is limited to small environments, since it cannot be used when the virtual space is much larger than the physical space.

To solve the problem, there have been different techniques to move the camera in the virtual world without requiring the user to change position.

The simplest movement option is moving in a vehicle. In this technique the user does not physically move, it controls a vehicle that moves through the environment and that can be controlled with a joystick. Moving in a vehicle allows the user to achieve easy and intuitive navigation because the movement can be controlled simply with a controller in the same way as it would in any videogame. However it can still produce motion sickness [RO16] because the optical flow does not match the vestibular system's senses.

In teleportation the user points at the floor with the controller and she is instantly transported to that point. Teleportation avoids the motion sickness by simulating discrete movements, however it tends to produce disorientation and loss of presence [Liu+18]. Since this technique is used to traverse large distances, disorientation can be a major problem. Probably due to the low motion sickness, teleportation + normal walk is also the navigation method used in Valve's the lab, a free compilation of VR demos, screenshot in figure 6. Since it is a free application made by the headset developers, it is usually one of the first VR games that most VIVE users will experience.

Movement in a treadmill is very similar to the movement in natural walk. Some techniques use an omnidirectional treadmill to move in the virtual world to allow the user to move in a natural manner. While it is supposed to biomechanically



Locutor: Usa la varita para avanzar.

Figure 6: The Lab screenshot

identical to normal walking, it alters users' perception of motion due to missing vestibular feedback and alters the user's gait cycle [DRT07]. It is also limiting, since when using a treadmill, you cannot combine them with other methods, most importantly normal walk, due to the required setup.

Some different approaches have also been reviewed, such as using the body as a joystick and leaning in the direction the user wants to move [MPL11].

2.2 WiP techniques

WiP was first introduced in 1995 [SSU95], where by introducing natural movement in the navigation they reduced motion sickness and increased the sense of presence.

In early implementations of WiP, when a step was detected, the virtual camera would move a certain distance and then stops as shown in figure 7. Three methods were tested (a) a fast displacement after a step is detected followed by an abrupt stop, (b) similar step function, but with slower speed during a longer time, and (c) abrupt movement starting after a step is detected and then a deceleration until it completely stops. These types of movement were unnatural, because when we walk in the real world, we observe a constant flow of movement without completely stopping between steps or suffering abrupt velocity variations.

In 2008 **LLCM-WiP: Low-Latency, Continuous-Motion Walking-in-Place** [FWW08] was developed. The paper described a WiP interface with 4 goals:

- Low latency: the virtual camera starts and stops moving soon after the user starts or stops walking.

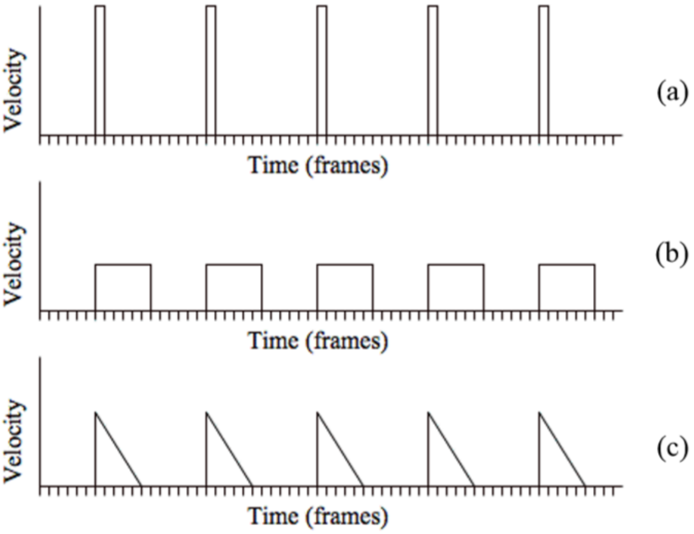


Figure 7: Models of viewpoint movement speeds used by early walking-in-place systems[FWW08]. In (a) each step results in a forward movement on the next frame. In (b) and (c) the movement is split between a few frames.

- Smooth locomotion between steps: when doing a real walk, the speed between steps never drops to 0, therefore it should not do it either in the virtual world.
- Continuous control of locomotion speed within each step: adjusting the virtual camera speed according to the user's movement within each step.
- Incorporation of real-world turning and short-distance maneuvering into virtual locomotion: not detecting minor feet movement and direction changes as steps.

To better match the movement of WiP to normal walking, the biomechanics of the two walking techniques have been studied in order to better detect when a user has started walking and to set a speed that adjusts to how we normally move [WWB10]. The diagrams in figure 8 show the cycle of walking in both normal walk and walk in place. This study along with LLCM-WiP served as the basis for incorporating biomechanics in our project, since we want the movement in the virtual world to be as similar as possible to the movements of real walking.

While we aim to replicate natural walking speeds in the virtual world, trying to move the virtual camera with the same speed as when walking normally may not be ideal for WiP. In [NSN14] it was questioned whether faithful reproduction of real walking speeds is desirable. The results suggested that to make the optic flow feel natural in the WiP from [WWB10], the visual walking speed should be of around 1.8 times the user's real walking speed. The study was only done for higher visual walking speed, since in all the related work where a treadmill was used the natural visual walking speed was higher than the real walking speed.

WiP techniques have been tested on a variety of devices, from simple HMD using

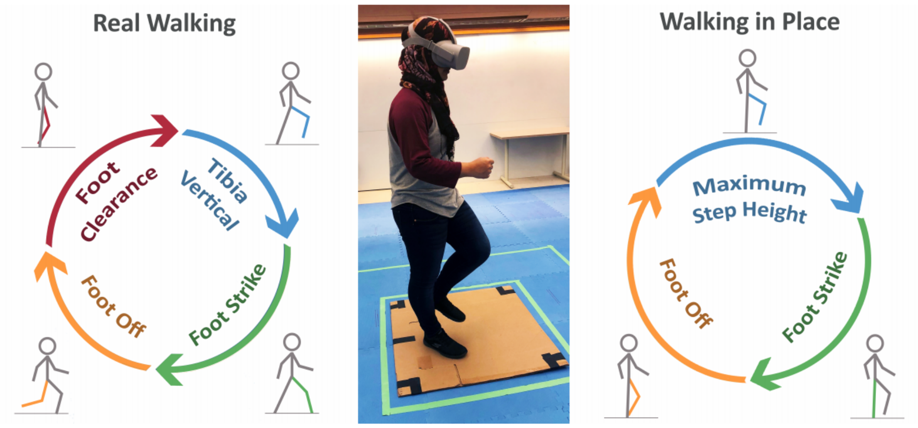


Figure 8: Simplified gait cycles for a single leg from [WWB10], simplified diagram from [Han+19]

a phones' screen and sensors [Han+19] to using sensors attached to the knees, waist, head and feet [TDS99], to different setups such as the CAVE [Raz+02]. Using more trackers provides more information that can be used to match the camera speed better to the user's movements, however it makes it more difficult for people to use the system due to the increased economical cost.

Recently, deep learning has also been tested by Hanson et al [Han+19]. They stated that a pattern recognition approach for WiP has not been attempted since its early days. Their work focused only on the change between normal walk and WiP, with the speed of the WiP movement being determined by a state machine for all tested methods, which means very little or no research has been done in using pattern recognition specifically on the speeds of WiP. They hypothesized that deep learning works well in WiP techniques because the gestures are unnatural and thus it becomes tiresome and people resort to a more relaxed type of motion. Deep learning is more resistant to small variations in the motions because it uses data of people walking for a considerable period of time.

3 Improving the original project

3.1 Understanding the base paper

In order to explain the limitations and the changes made to the Unity project, we first need to understand what the code was calculating. The base paper has 2 parts, first it explains a method to differentiate between normal walk and WiP, and then how to calculate the required optical flow from feet movement.

The method to determine the walking technique is based on the fact that during normal walk the user displaces the head over the horizontal plane, significantly more than during WiP. Therefore, the condition to switch from normal walk (NW) and WiP is determined as follows:

$$S = \left\{ \begin{array}{ll} NW & if\left(p_t^H - p_{t-1}^H\right) > (\delta + \epsilon) \\ WIP & if\left(p_t^H - p_{t-1}^H\right) < (\delta - \epsilon) \end{array} \right\} \tag{1}$$

where p_t^H is the projection over the horizontal place of the position of the HMD at time t, δ is a threshold and ϵ is a window to avoid constant flickering.

In WiP the virtual camera direction is calculated as the sum of the unit vectors of the feet orientation provided by the HTC trackers.

The magnitude of the walk velocity is computed as:

$$v_t^{fwalk} = \alpha v_t^{walk} + \frac{\gamma}{n} \sum_{i=1}^{n} v_{t-i}^{walk} \tag{2}$$

where v_t^{walk} is the absolute sum of the feet velocities at an instant t and α and γ are user defined parameters such that $\alpha + \gamma = 1.0$ (with the best values found empirically being $\alpha = 0.25$ and $\gamma = 0.75$). The parameter γ determines how much importance we give to the current feet velocity, and α determines how much we attempt to follow the current velocity. This equation calculates the movement magnitude by using the feet velocity at a certain instant and at the previous n instants ($n = 5$ is found to be a good value) in order to smooth the movement and avoid halting completely the camera if no movement is detected for a single instant.

3.2 Changes made

The first change that was made in the original project was to fix the feet orientation. Vive's trackers changed orientation when, for example, a controller was connected, for reasons that we still do not know since such information is not provided in by the manufacturers. A quick software solution was programmed to revert the change. The Vive's trackers were a source of problems at several times during the project, therefore we had to write software solutions to solve them and stop them from introducing noise into our system. We decided to simply develop quick solutions to each single problem as opposed to one large robust solution, because given how fast this technology is improving, we realized that by the time we had reached a robust software solution, a new hardware version without said problems would probably already be on sale.

A small but important change introduced during this stage was multiplying the final camera speed. In [NSN14] it is argued that virtual camera speed in a treadmill or WiP system need to be multiplied by a gain factor because we percieve virtual speeds as slower than they really are. Using an implementation of the WiP from [WWB10] results showed that a WiP system requires a gain factor of around 1.8 for virtual speeds to feel natural. After some tests we determined that in our system a gain factor of 1.5 was enough. In all the charts of this document that show virtual camera speeds, the value shown is already multiplied by the gain factor.

In order to further smooth the user experience, we then made some changes to make sure the system achieved similar goals as LLCM-WIP [FWW08]. The goals that our base system did not satisfy when this TFG started were:

- 1-**Low latency** (when stopping) Shown in figure 9
- 4- **Incorporation of real-world turning and short-distance maneuvering into virtual locomotion.** Shown in figure 10

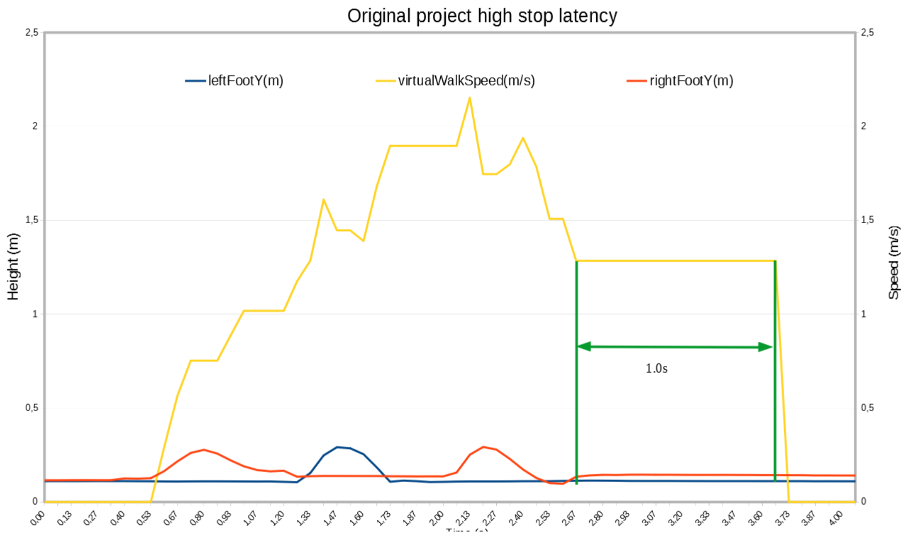


Figure 9: Example of the high latency when stopping. After 3 steps the user stops completely but the virtual camera keeps moving for 1 second (waiting for a possible next step). The camera movement while the user has stopped is very likely to cause motion sickness

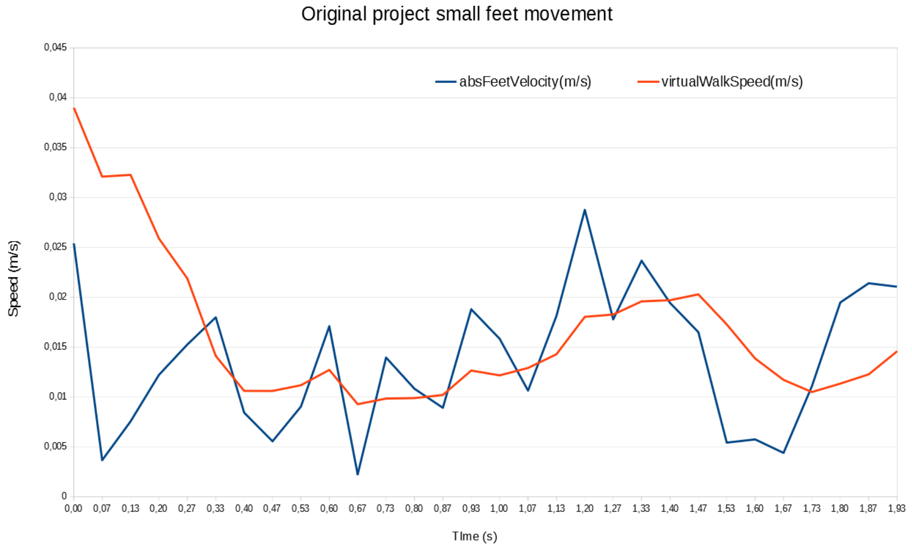


Figure 10: Example of not allowing small movements. Even very small feet movement cause a small camera movement. While the camera movement seems insignificant, the user feels that there is a movement even when the motions performed were not meant to generate any.

The high stopping latency was fixed by implementing a linear equation to slow down when no movement has been detected for a certain period of time instead of keeping a constant movement, while maintaining the same total time before virtual camera speed reaches 0 (figure 11). The short distance maneuvering was allowed by introducing a height threshold of 2cm, value extracted from

[WWB10] (figure 12). If the feet do not move above the threshold, the camera does not move, and thus some maneuvering and small movements are allowed without affecting the visual flow.

Another improvement that we noticed was needed, was to make the transition from WiP to normal walk more responsive. The base method would switch between WiP and normal walk, based exclusively on the feet movement. However, in reality, for this particular movement the head is translated because the feet have moved along the horizontal plane (one feet is separated from the other). Therefore, we can extend the previous method and include also feet translation. The new equation to determine the walking state is shown in equation 1. In this new equation we have added an additional condition to change the state based on the feet distance over the horizontal plane:

$$S = \left\{ \begin{array}{ll} NW & if\left(p_t^H - p_{t-1}^H\right) > (\delta + \epsilon) \;\; or \;\; \|p_t^L - p_t^R\| > d_{max} \\ WIP & if\left(p_t^H - p_{t-1}^H\right) < (\delta - \epsilon) \end{array} \right\} \qquad (3)$$

Where p_t^L and p_t^R are the Left and Right foot position projection on the horizontal plane, $\|p_t^L - p_t^R\|$ is the distance between the feet, and d_{max} is a distance threshold that we set to 0.4m (we empirically found that when users perform WiP, this distance never goes beyond that value).

The slow down stage of WiP is an important and troublesome part of mapping feet movement to camera speeds. In normal walk, we do not halt instantly, we have to first slow down. In WiP however, there is no slow down, we finish a step and simply do not move again. Therefore we have to decide how much time to wait when both feet are on the floor before starting to halt, since maybe the user has not stopped, and is going to move the other foot next. If we wait too little the movement does not feel smooth since the camera speed changes too much (e.g. it could drop velocity drastically between consecutive steps), if we wait too long, the system feels unresponsive (e.g.the camera keeps moving forward when the user wants to stop, making the user lean forward unconsciously). Both cases are potential sources of motion sickness. Finding a good trade-off between latency and responsiveness is a problem that will persist in the next stages of the project.

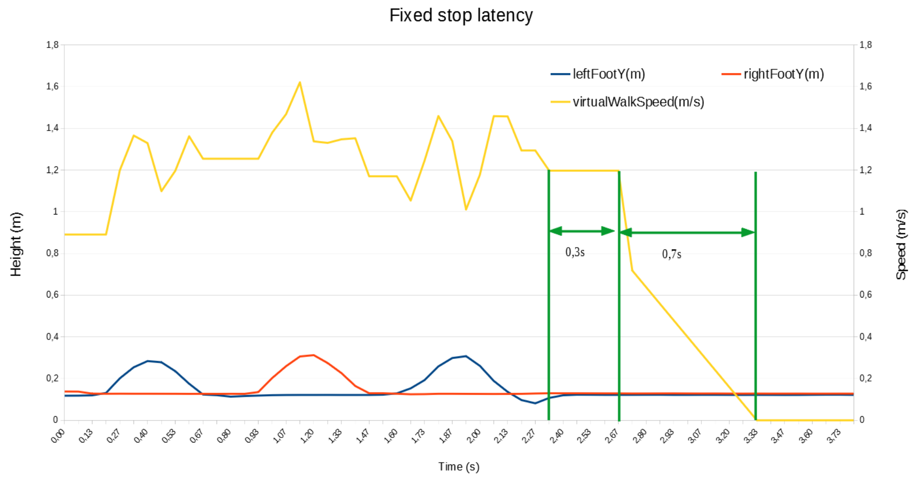


Figure 11: Fixed stopping latency. After 0.3 seconds of no feet movement the camera reduces the speed and starts slowing down. The time it takes to completely halt remains 1 second, but the slow down makes it feel more natural.

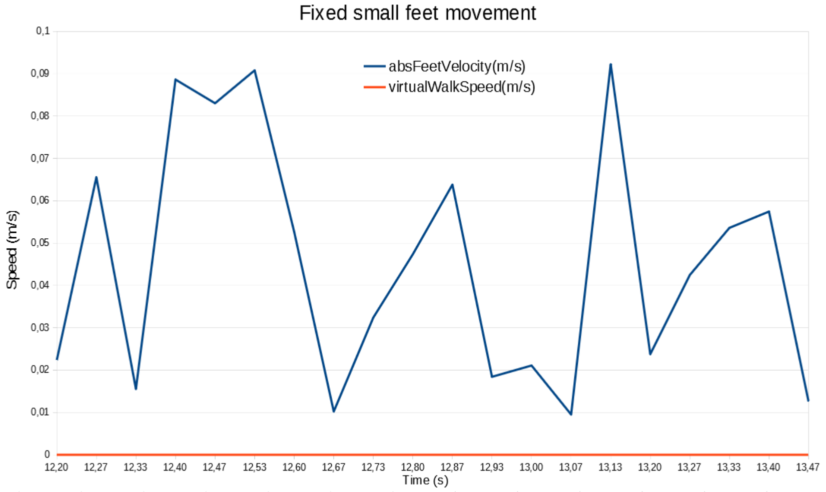


Figure 12: After adding a height threshold, small movements do not move the virtual camera.

4 Creating new WiP modes

4.1 Need for slow and fast WiP

While researching for the creation of the base paper, it was found that current WiP systems may not be sufficient for all situations that require movement. A clear example is putting the user in an emergency situation like a fire in a building and telling him/her to escape using a WiP interface. In real life, a person would run or walk quickly in a significantly different way compared to how he/she was moving before, and so in a WiP system the user should also be able to Run in Place. If the user keeps just walking in place, the sense of urgency is likely to be lost, thus reducing immersion and presence. Slow WiP may also be needed for some situations, for example walking in a crowd or in an environment with a lot of elements that the user may want to look at or do some maneuvering to avoid. In that situation the user does not want to completely stop, since it would delay completing main goal, but it cannot move too fast either since it could lead to collisions.

In the FUTURE WORK section of [WWB10] it was stated that:

> "GUD WIP – as with most WIP systems – only accepts Walking-In-Place inputs. If a user's gestures indicate running, our GUD WIP implementation prints an error to the screen. In our experience achieving consistent in-place running is considerably more difficult than consistent in-place walking – more so than the apparent differences between Real Running and Walking would suggest. **However, VE users almost certainly wish to run through virtual environments. 'Running-In-Place' systems are very research-worthy.**"

Their statement further reinforces the idea that investigating different WiP modes is interesting and can allow for more WiP applications to be developed.

First of all, we tried to Walk in Place slowly and fast in our system to see how well it was able to handle an input that it was not designed for. The results in figures 13 and 14 show that indeed, our base system also did not respond well to attempting to run or walk slowly in place, and thus studying the differences between walking in place at different paces is a good research topic.

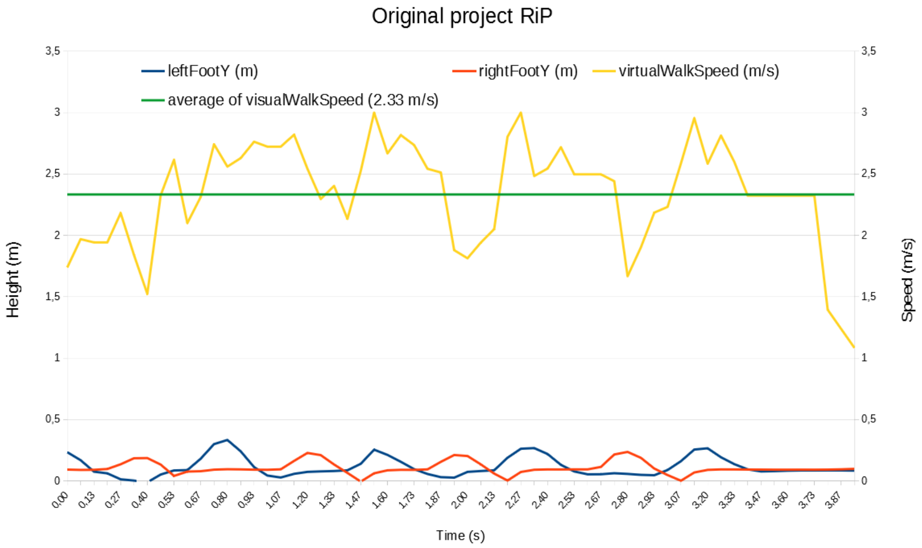


Figure 13: Attempt to Run in Place in the original project. While the optical flow is faster than when walking in place, it is still much slower than the output we expect when running in place. The result is a confusing experience where we see movement, but not the movement we expect. In the recording of this data sample, the virtual camera speed caused irregular feet speed and frequency due to the user trying to keep running in place. The virtual camera did not match the user's expectations.

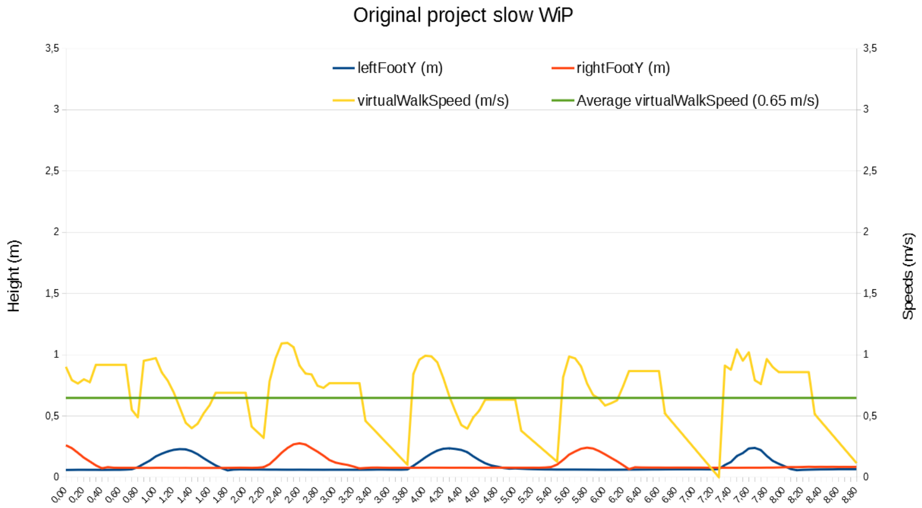


Figure 14: Attempt to walk in place slowly in the original project. The virtual camera slows down significantly during feet change due to the user staying with both feet on the ground for a longer period of time. The change from moving one feet up to moving it down also takes more time than the system expects so it also reduces the speed. The user experiences very sharp changes in the camera speed, which tend to produce motion sickness.

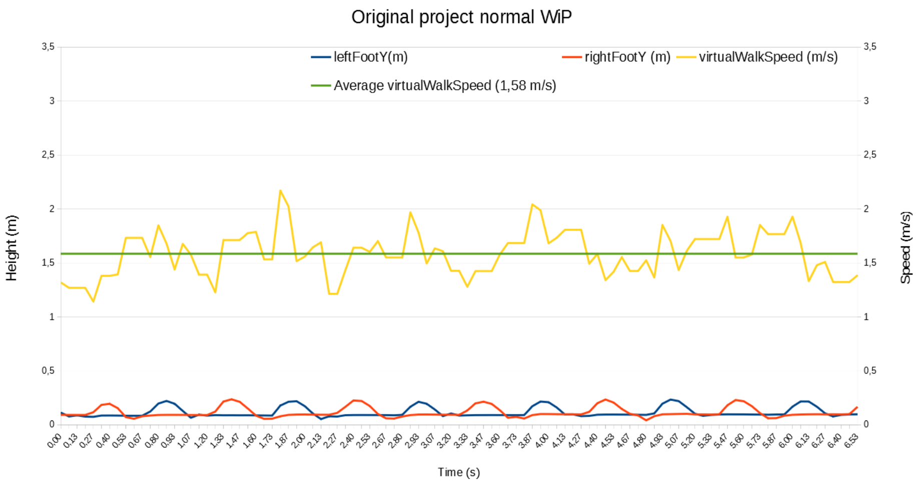


Figure 15: Normal WiP feet and virtual camera speed.

4.2 The state machine and new mapping function

After seeing that the WiP system could not handle fast and slow WiP there were two ways to continue: either to change the mapping function from the feet movement to the virtual camera speed in order to better adapt to the user's variety of movements or to create a state machine with 3 states: slow, normal and fast each with a different mapping function.

The original mapping function used feet speed as the only biomechanical information to determine the speed magnitude. This way of calculating the virtual camera speed had two problems when handling Running in Place:

1. Feet speed when performing a Run in Place is not significantly higher than when doing a Walk in Place
2. The combination of the room setup, the Vive's bases precision and tracker's wobbling caused a significant number of incorrect readings during Run in Place as shown in figure 16

The reason behind all the noise that appears in the tracker, is that the VIVE trackers combine two technologies to determine the position of trackers. Firstly, the bases (light houses) send a moving infrared beam which is detected by each tracker and used to calculate its position in global coordinates. Secondly the trackers use inertia sensors to update their position at a higher rate that infrared updates. Inertia sensors are very sensitive to fast movements and to the strong contact forces that appear after stomping the feet on the floor, which is what causes the errors in the Y values.

While feet speed alone could not be used in RiP, we noticed that feet frequency did change a lot compared to WiP . If we compare the Run in Place and the

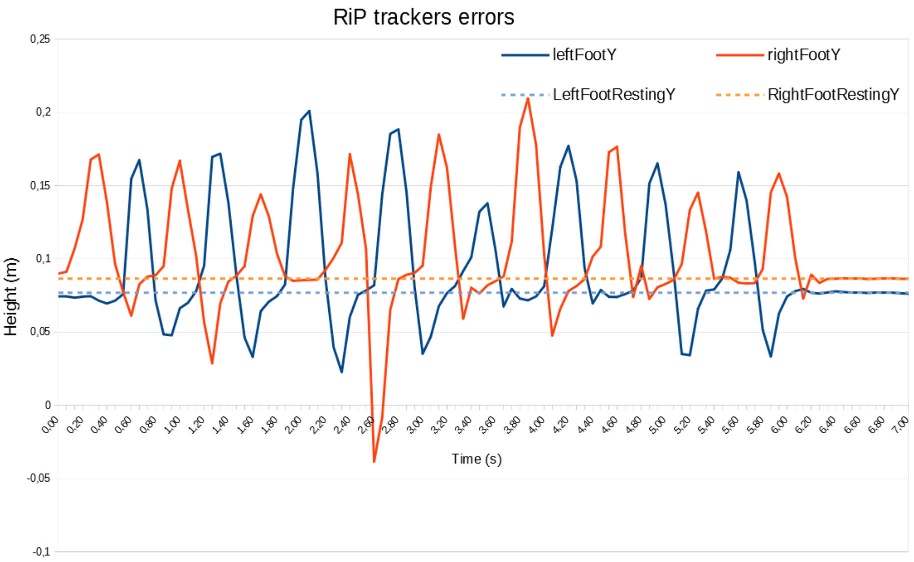


Figure 16: Feet position when running in place and trackers position when feet are on the floor. If all readings were correct, the minimum height should only be slightly below resting position due to wobbling, however values go far beyond that point and sometimes even show negative values despite the floor being calibrated as height 0.

Walk in Place of figure 17 we can see that normal Walking in Place shows some resting time between steps, however fast Running in Place shows little to no resting. After some further testing we concluded that we could reliably measure step frequency by adding a filter to discard false steps produced by noise and if we only used step frequency in RiP, feet position readings below the resting height would not affect the speed. Thus this is the state machine that we developed based on frequency:

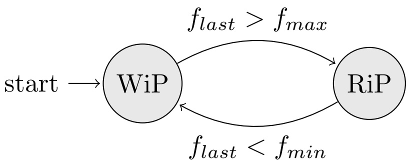


Figure 18

where f_{last} is the frequency calculated from the last step, f_{last} is the inverse of time since the last step happened $(1/t_{last})$ and f_{max} and f_{min} are thresholds to change the state. Therefore the transition from WiP to RiP can take place every time the user completes a step, but the transition from RiP to WiP will happen mid-step to avoid the need to wait for a whole slow step.

The state machine is simple and its performance relies on how accurately the thresholds are adjusted to each user, since it was already designed with a future training stage in mind. The goal of the training is to adjust the values of the thresholds to each user style of walk and run in place.

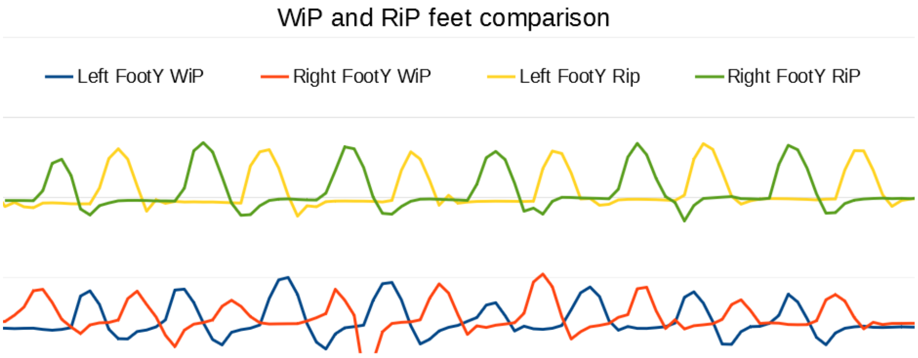


Figure 17: Comparison between WiP and RiP feet movements

To the best of our knowledge, a RiP system has not been developed yet. Thus we have to create a new feet movement to camera speed function for RiP, and use only step frequency since, as previously argued, it is the most reliable biomechanical parameter we are able to use. Since our ultimate goal is to mimic real world speeds as closely as possible, we will first define what "real world running speeds" are. We want to allow the creation of applications where the user can run for a long period of time and not just a few seconds. Therefore we will first determine the speed range of an adult running at an aerobic intensity, which can be performed for an extended period of time. An adult person running with aerobic intensity in a flat terrain will run at a speed of around $2.2m/s$ ($8km/h$) to $5.6m/s$ ($20km/h$) with the first being a very light pace, and the second being a very intense pace that only professional athletes can keep for extended periods of time. We don't expect a user to Run in Place at the equivalent pace of running at $5.6m/s$, but the value will serve as an upper bound. Thus our RiP system will be designed to output speeds of $2.2m/s$ when stepping at relatively low frequencies and output $5.6m/s$ when stepping at very intense frequencies. After some preliminary tests we determined that a good match to speeds of $2.2m/s$ is a step frequency of $1.5s^{-1}$ and an intense and fast stepping was of around $6s^{-1}$, which will be the match for $5.6m/s$. Finally, we have to decide how to match the values in the middle. Literature suggests that the speed of real walk is a square root function of the energy spent on the motion [Bob60]. In the article, for some people step frequency was also measured, which showed a logarithmic relation with speed. During RiP, the way to increase speed is to increase step frequency, and thus we believe that using a logarithmic function will be a better match to real world movement. The final function used in Run in Place is:

$$v_t = 3.156\ ln\,(1.338\ f_{last}) \tag{4}$$

3.156 and 1.338 were obtained by calculating a logarithmic function with $speed = 2.2$ when $frequency = 1.5$ and $speed = 5.6$ when $frequency = 6$

During RiP, the camera speed only updates after a full step has been completed. In WiP mid-step adjustments were necessary. However since RiP mode only activates when step frequency is already high, the latency is already low and there is no need for further adjustments during the steep

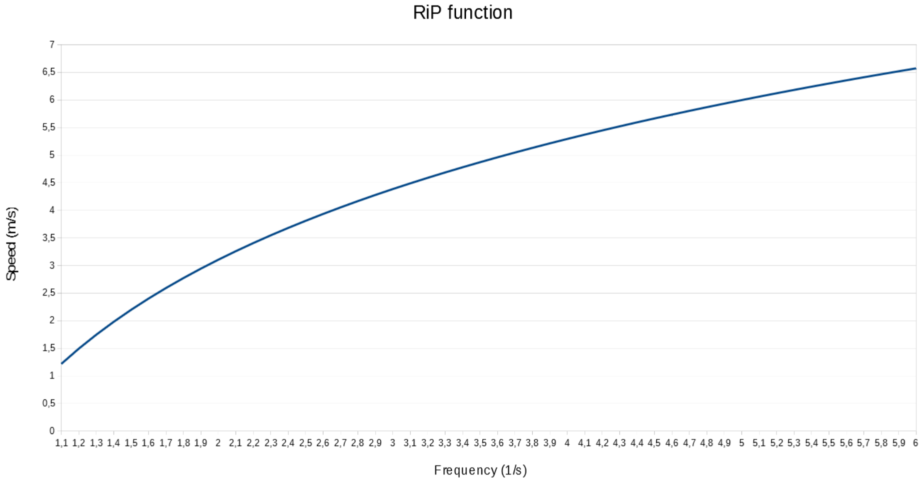


Figure 19: Run in Place Function.

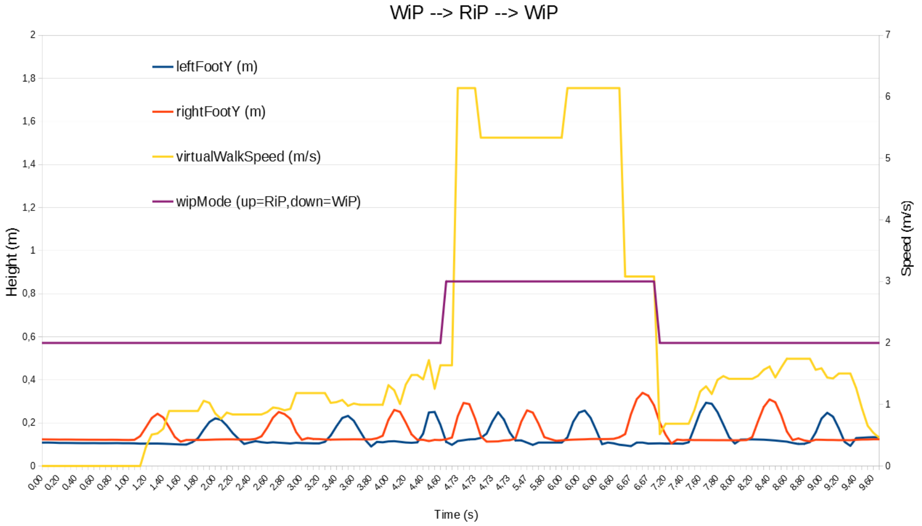


Figure 20: Changes between normal and fast WiP. In the chart, the user first walks normally, then runs and goes back to normal WiP. The virtual camera speeds during fast WiP are much higher.

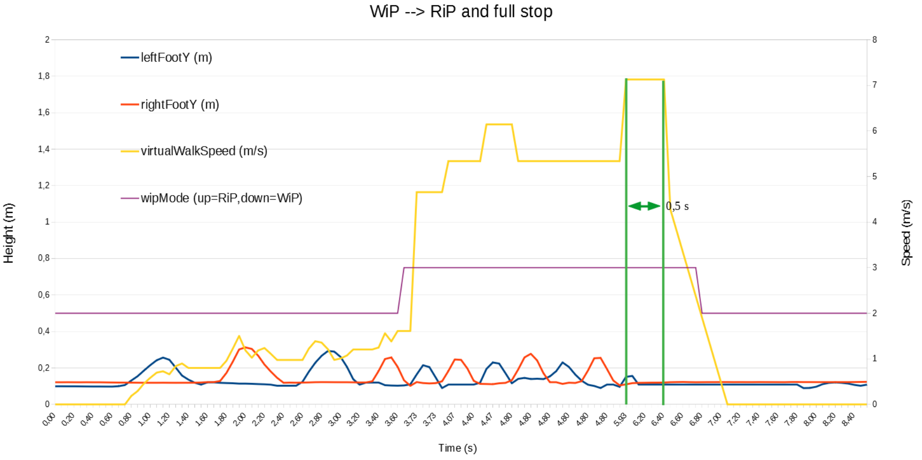


Figure 21: Slowdown in RiP. When both feet are on the floor, the camera keeps moving waiting for another step and then sharply reduces the speed.

Preliminary experiments with our model for Run in Place showed good results. The sharp speed increase when switching modes allowed users to quickly traverse the environment with speeds feeling like a natural output to the movements they were doing. However users still felt motion sickness due to the stopping latency. Initially, the slowdown started 0.5s seconds after both feet were on the floor, as shown in figure 21, approximately the same latency that did not have a negative effect on WiP. When walking or running in place, we do not have a slowdown period, we just do not move a foot up again. In normal movement, as speed increases, the time before fully stopping also increases and the slowdown is less abrupt (e.g. you can stop immediately in the middle of walking, but not in the middle of running as the inertia force would make you fall forwards). We have noticed that when navigating in an environment using a WiP interface the opposite is true, that is: as virtual camera speed increases, the stopping latency should decrease and the speed decrease should be sharper.

This is because when you walk or run in place, no matter how fast you are performing the movement, since it is limited to an up/down feet movement, there is no inertia making you lean forward if you stop abruptly, and thus you expect the camera movement to stop the forward movement in a more abrupt manner.

In our implementation the camera does not fully halt until the threshold to change to normal WiP is reached, however the speed is significantly reduced after 0.2s with both feet on the floor.

Our initial idea was to create a third state for slow WiP, but we later decided to combine slow and normal WiP into a single state. We did not find a consistent way to distinguish the movements of slow and normal WiP. The initial experiments showed that feet frequency was not different enough to be the key parameter to decide state changes, so we looked into the changes in feet speed. While there were some differences and we implemented the third state by changing modes when feet speed was low for a specific period of time, the mapping function was too similar to normal WiP. Thus instead of creating a new state we merged some aspects of slow WiP into the normal mode affecting it as little as possible.

In the original project there were 2 stages of walking in place where speed would decrease sharply: when changing feet, and when one feet was transitioning from moving up to moving down. To avoid reducing the speed too much during mid step, we added a minimum camera speed so that if one feet is not on the floor, the camera will never stop moving. Doing so allows the user to perform the step slowly while keeping the camera movement smooth, and does not interfere with normal WiP. The only possible scenario where this method does not work is if the user keeps one feet in the air for a long time, which we sincerely hope will not happen. The original project already sampled the last 5 feet speed in order to avoid sharp speed changes mid step, however it was not enough when feet movements were too slow, as previously shown in figure 14. Increasing the number of sampled speeds would lessen the speed decrease, however it would also reduce the system responsiveness, so it would not be a good solution.

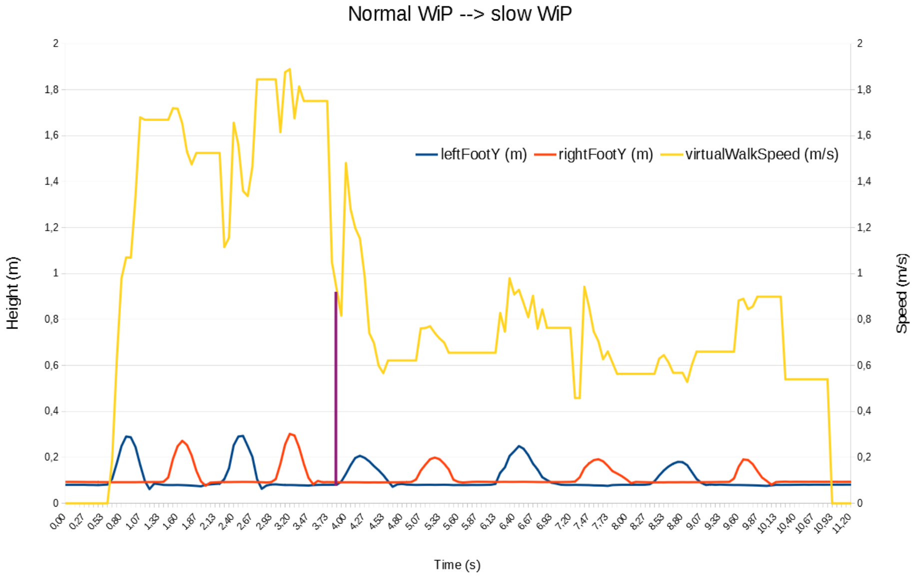


Figure 22: Slow and normal WiP after the changes. The user first performs some normal steps then some slow steps. During the slow steps, speed changes are smooth and during feet change speeds do not get close to 0 like in figure 14. The vertical line indicates when the user started to perform slow WiP

The close to 0 speeds when alternating feet was solved by using the halting function from the original project, that is: simply keeping a constant speed for a second and then stopping. This was implemented by tracking the last feet movement speeds and using one slow down function or another depending on whether the feet speeds before finishing a step were slow or not. Since we only keep the constant movement when the camera speed is already low, the user does not feel motion sickness even if he/she stops after a slow WiP, and the constant speed allows for a more smooth experience. The results can be seen in figure 22

5 Designing a training stage

5.1 Project state so far

At this point of the project, we already have the full VR movement interface. The different states are shown here:

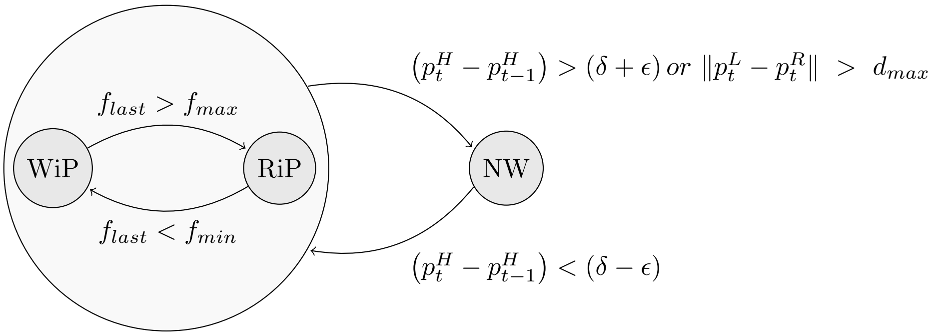


Figure 23: Full state graph of the project. Includes transition between normal walk and WiP of equation 3 and between WiP and RiP from figure 18

However, every parameter is manually adjusted, so the system works the same way for every user. In [SRP19] and [WWB10] it is left as future work to adapt the systems to every user, adjusting certain parameters and function, which suggests that doing so will improve our system too.

5.2 Training stage development

The general goal of the training stage is to first make the user move in a controlled environment under controlled conditions so that we can extract certain biomechanical information of his/her movements to adjust the parameters of our new model. Once the training stage is finalized and the parameters are extracted, the user can start to navigate an open environment using the WiP model adjusted to his/her own walking preferences.

Since we want to focus on certain biomechanical parameters, it is important that we design a WiP where we have as much control as possible of the user's movements while still being able to record certain data. Our way of achieving this has always been the following: move the virtual camera with a certain speed and let the user walk in place in a way that he/she feels comfortable for the given input optical flow. However, the different parts of the training stage, and how we obtain the speeds that are shown to the user has changed several times. We will first describe how the training stage has changed, then we will provide some important findings that we discovered during our experimental studies and finally we will explain how our final training stage is.

We started by creating a training stage with only one part where we would set virtual camera forward movements and record how the user would try to

Walk in Place to adapt to the movement in order to extract data. Our initial idea was to record a person's head movement during normal walk and replicate those movements in the virtual camera. Thus we recorded the normal walk at a medium speed of a few participants (from the ViRVIG group) and tested how participants reacted when the virtual camera was moving with the recorded speed. In one of the recordings, the movement was not very natural and the walk stages were more pronounced than they should have been. However, we still kept the recording as it had some distinctive effects when used as the virtual camera. Figures 24 and 25 show an example of a natural and pronounced walk.

Using each of the movements as the virtual camera resulted in very different results. As for the movement of figure 24 the camera speed changes were sharp and very easy to notice by the user. Before performing internal tests we theorized that people would sync their step frequency to the movement they were seeing, but they would finish a step when the virtual camera speed was lower, since it would intuitively be more natural. All the test subjects showed the same behaviour: they all synced their frequency to the steps they were watching, however at which point they would do it changed subject to subject. Some finished steps when the speed was higher and some when the speed was lower, as shown in figure 26. Even though we only tested this experiment with 5 participants, we believe that the fact that they all synced their steps with the input camera frequency means that this work requires further research. In future work, it would be interesting to further explore the fact that it appears to be possible to manipulate the user's step frequency during WiP, so that we could force the user to perform a constant frequency while measuring other biomechanical parameters.

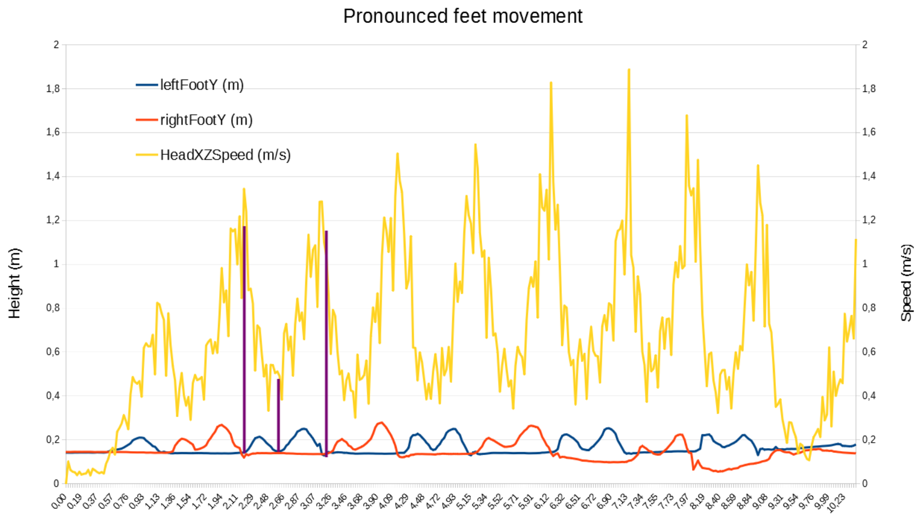


Figure 24: Pronounced feet movements. During normal walk we start the step with the leg we have behind. To move forward the leg has to first reach the other one, and when that happens, the feet height goes down to almost 0 since it gets close to the ground during the swing phase. The feet then rises again to go forward and finally goes down to the floor again before the cycle repeats itself. The maximum head speed happens at the end of the step, when the user switchs the body weight from one foot to the other (swiches standing foot).

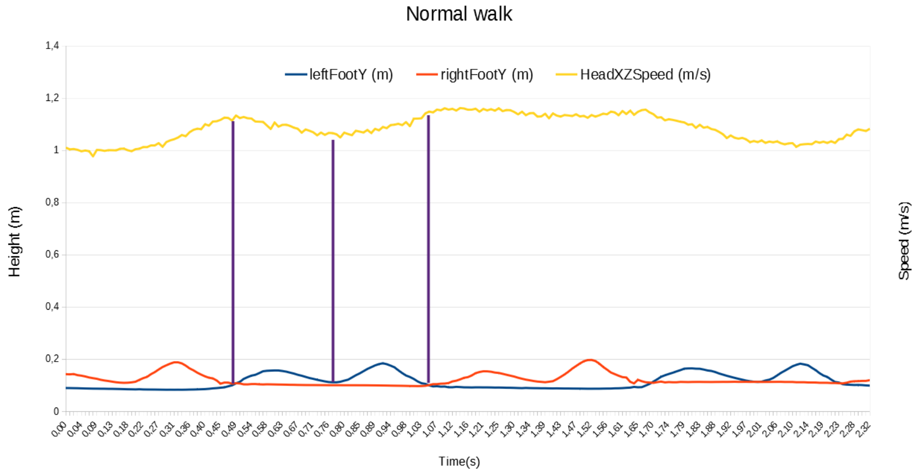


Figure 25: Natural normal walk. It shows the same patterns as the more pronounced chart. A lowpass butterworth filter, implementation extracted from [Ber], has been applied to create the chart so that the pattern is easier to see. The vertical lines show the moments when a step starts, when it ends and when a foot is at the lowest point

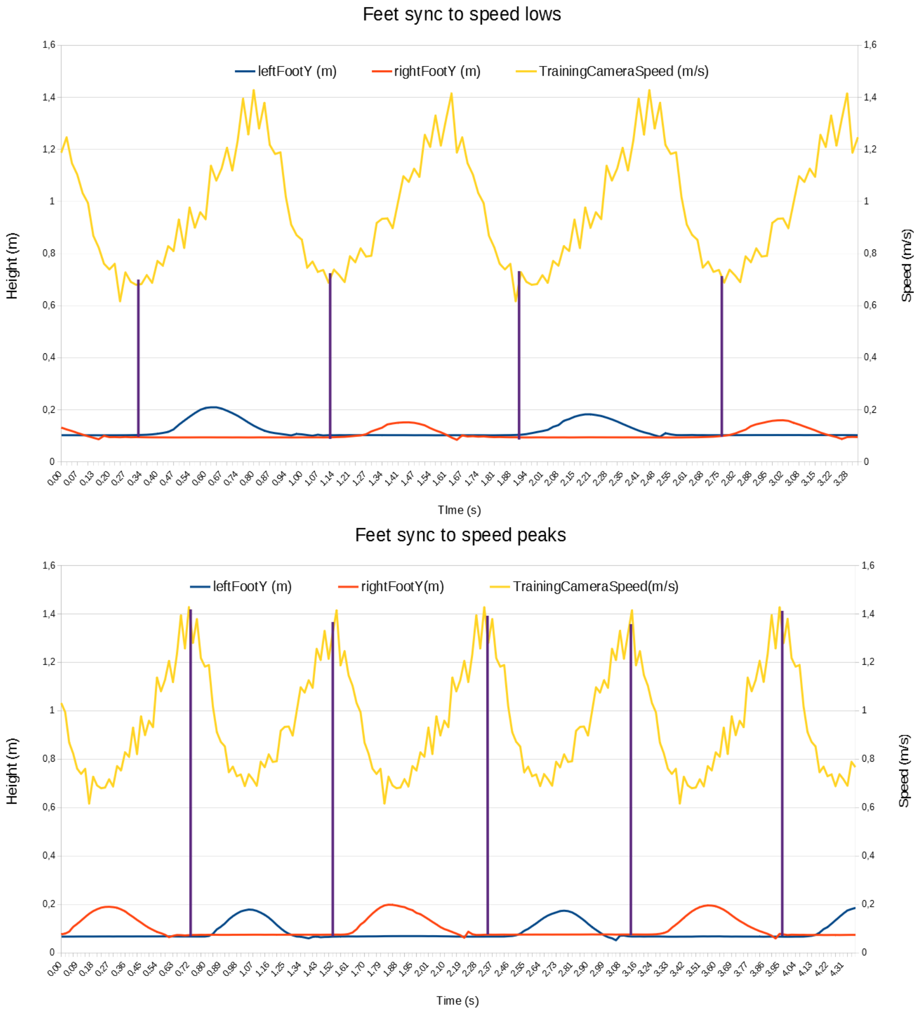


Figure 26: Different users syncing their feet to the virtual training camera in different ways, the first user syncs the step end to the camera's lowest speed while the second user syncs to the camera's highest speed. The vertical lines indicate the synchronization points

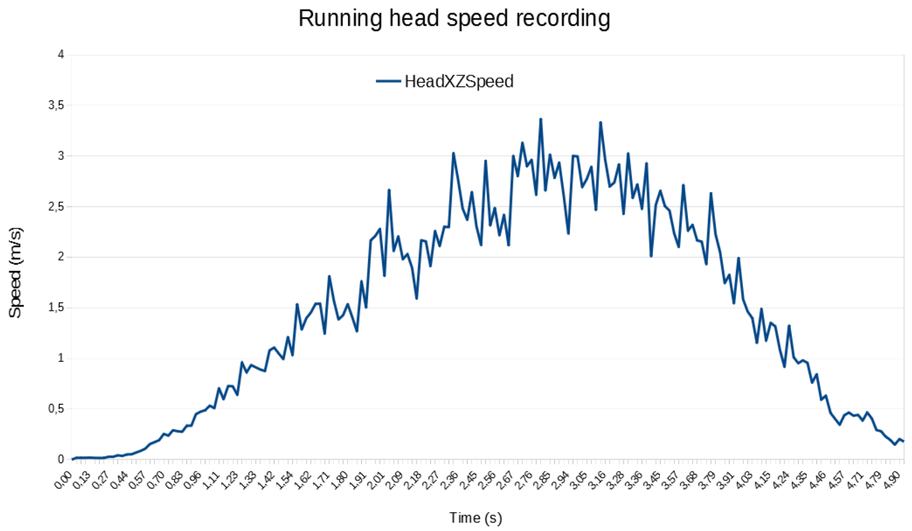


Figure 27: Running takes some time to reach a constant speed. The chart only shows an acceleration and deceleration and thus we can not extract useful information from it regarding constant run speed.

When replicating the natural movement of a participant (e.g. without such noticeable speed changes) as the input camera speed, no user synced their steps to the steps of the movement. We believe this is due to the fact that even if we can see patterns in the movement, natural walk speeds are very close to being constant. Figures 24 and 25 show this: users could feel a change from 1.2m/s to 0.4m/s but not from 1.2m/s to 1.1m/s. Therefore, the movement felt like it had a constant speed and we decided to drop the idea of replicating the exact recorded speed. The process to save a movement to replicate required to record the movement and then manually cut it and edit it slightly so that it would feel natural when it looped, making the process automatic would have required too much effort. Now that this was not needed we could simply record the user's average speed and then replicate a constant movement that would still be based on the way we move.

The most important part of the training stage was deciding what biomechanical parameters we needed to extract and what we would use them for. We wanted to at least adjust the WiP and RiP transition thresholds and regulate the speed in a way that would depend on the frequency and the feet height during each step. Since the state transition thresholds seemed to be easier to adjust we first designed a complete training stage with only those parameters in mind and then expand upon it to also adjust the speed.

In order to extract the information to train the RiP mode we would have ideally recorded the user running, however due to space limitations, even in the room shown in figure 5 the data we tried to record was not accurate, as in figure 27. Thus we had to record a fast walk and apply a bigger gain factor to the speed shown in order to match the speed that the user would have when running.

5.3 Final training stage

The final training stage has two parts, first the user performs normal walk in 3 speeds, normal, fast and slow through the room and then Walks in Place and Runs in Place using the speeds obtained from the previous parts. To get more natural recordings in the normal walk parts, the user has the HMD on top of the head, so that he/she is not watching the screen. This is better to take realistic measurements because the user's movement are affected by seeing the virtual environment instead of the real world, and there is plenty of evidence in the literature that humans tend to walk slower in virtual reality even when the virtual world is an exact match of the real one [NSN14] [Jan+19] . The following diagram shows the full training stage:

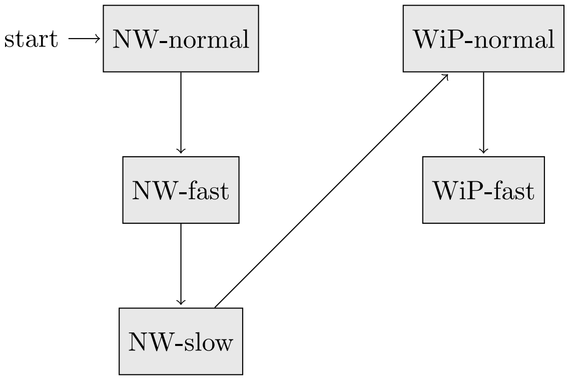


Figure 28

- **NW-normal**: Average head speed is recorded to be used in **WIP-normal**.
- **NW-fast**: Average head speed is recorded to be used in **WIP-fast**
- **NW-slow**: Average head speed is set as the minimum camera speed in the WiP system.
- **WIP-normal, WIP-fast**: feet frequency is used to calculate WiP and RiP thresholds using equation 5

During the WiP parts, the virtual camera automatically moves at a constant speed and the user is asked to perform WiP motions until it feels that the movements match the virtual camera, it is at that moment that the recording starts. We found that a recording duration of 15 seconds is enough to obtain the information needed, but short enough so that the user is not tired after the training.

To calculate the thresholds between WiP and RiP using the data from the training stage we use the middle point between the frequency of walking in place and running in place. To further adjust to each person and avoid flickering we introduce a window. The window size is calculated from the standard deviation of the step frequency data samples obtained after each step is completed. This way the window size increases if the user's step frequency can change significantly . The final equation is thus:

$$f_{mid} = avg(f_{norm}) \ + \ (avg(f_{fast}) \ - \ avg(f_{norm}))\,/2$$

$$\begin{aligned} f_{max} &= min(\ f_{mid} + stdev(f_{norm}) \ , \ avg(f_{fast}) \ ) \\ f_{min} &= max(\ f_{mid} - stdev(f_{fast}) \ , \ avg(f_{norm}) \ ) \end{aligned} \tag{5}$$

Where f_{norm} and f_{fast} are the recordings of the frequencies resulting from the user's steps during the WiP and RiP training respectively. The average of the frequencies also acts as upper and lower bounds in the case that the standard deviations are too big.

6 A new WiP method

The system described so far is the final completed WiP and RiP interface developed in this project. The system allows to Walk in Place, Walk in Place slowly and Run in Place, and contains a training stage that adjusts the state transition thresholds to each user. The goal of the training stage was also to change the virtual camera speed for each user, and initially we were going to try to adjust some parameters in order to achieve this goal. But as the design of the training stage advanced and more tests were performed, we started to consider the possibility to design a completely new WiP method instead of just doing some adjustments to the current one.

Developing a completely new method takes much longer and exceeds the scope and time frame of this project and therefore, at the time of writing this document, it is still a work in progress. However, the development of this project allowed us to run several user studies to investigate in depth the different biomechanical aspects that should be taken into account for the development of a new model. In the final stage of the project, instead of performing a usability study of the current system, we have conducted an experiment with users from both inside and outside the ViRVIG group in order to collect WiP data that will be later used to create a new mapping function for WiP.

The following subsections argue the need for a new WiP function and explain in detail how the experiments were performed.

6.1 Step frequency and feet movement

Throughout all this project, our goal has been to simulate our natural movement during normal walk to navigate in a virtual environment but without the need to physically move. In normal walk the movement speed ($|v|$) is produced by the step frequency (f) and the step length (l) [WWB10]:

$$|v| = f \times l \tag{6}$$

Of course, there is no direct equivalent of this equation for WiP, since there is no physical displacement over the horizontal plane. WiP that are based on walking biomechanics aim to find a similar function so that camera movement feels natural, but each method tries to do so in a different way. In [FWW08] and the base paper [SRP19] the main parameter used was feet speed. In [WWB10] however, it is argued that what we perceive to be more important in movement speed is feet frequency. In the final version of this project, feet speed is used in WiP, but feet frequency was found to be more useful in RiP, so both parameters have been useful. Maximum step height is another parameter that is mostly overlooked in WiP methods and could also be important, for example to replace the concept of step length in the normal walk equation.

During the design of the training stage, we argued and theorised about how users would adapt their WiP to a camera movement. Even within the small amount of data we obtained, we started to think that maybe neither of the two paths was completely correct. As camera speed changed users adapted their movement but they didn't just vary the frequency or feet movement, but a combination of frequency, feet speed and maximum step height. We started to theorize that if we obtained a bigger sample, and it included people with little to no experience with VR we would see different ways to adapt to the change in camera speed. We believed that most people would only increase step rate, but some would keep the frequency and instead increase step height and speed. With the purpose of understanding how Walking in Place changes as camera speed is increased, we designed and conducted an experiment.

6.2 Collecting people feet data

The goal of the experiment performed is to collect data of how each user Walks in Place when exposed to different camera speeds in order to understand how feet movement is adapted to virtual camera speed changes, and whether different users react in different ways or not. Analyzing the collected data will help us understand how the output virtual camera speed should be influenced by different user motion. As opossed to previous work, where biomechanical data is gathered from normal walk, the novelty of our study is that we gather biomechanical data directly from a walk in place movement. To be able to do so, we provide a fixed camera velocity to the Head Mounted Display and ask the user to perform a WiP that feels comfortable with the observed camera speed.

6.2.1 Experiment method

Design: After equipping the subject with an HMD and trackers on the feet as shown in figure 29, the virtual environment from [RP18] was shown to the user (see Figure 32) with the following virtual camera speeds: 0.5, 0.8, 1.1, 1.4, 1.7, 2.0 and 2.3 in ascending order. The user was told to perform a WiP motion until he felt that his motion matched the optical flow. The feet data was then recorded for 15 seconds for each speed.

Subjects: 13 physically unimpaired subjects performed the user study. Ages 20 to 48 (average = 26.6, median = 23). 9 male, 4 female. The user's heights ranged from 158cm to 183cm (average = median = 171cm). All participants had at least some experience with videogames and computers. 4 participants had extensive experience with VR, 1 had some experience, and for the other 8 this was their first contact with VR. No user had any experience with WiP systems. All subjects volunteered and were not paid to perform the experiment.

Procedure: The procedure is illustrated in figure 31. Before starting , participants received information about the purpose of the experiment and were explained the procedure. Subjects were told about space constraints in VEs, and introduced to the idea of Walking in Place interfaces to remove space constraints. Users were told that they would see different camera speeds, but not that the speeds would go from slowest to fastest. Participants who had never used any VR device before were left interacting with the Steam VR home simulation (Figure 30) for a few minutes until they felt comfortable performing normal walk on a virtual environment. A very general description of what the WiP motion is (moving one foot up and down, then the other foot) was given

Figure 29: User with the Vive HMD and trackers

to the users. The experimenter did at no point perform a WiP motion nor were the users shown any video or image depicting the movement. Some videos of the user's feet were recorded while they had the HMD on. Participants were informed that a video showing only their feet and legs may be recorded during the experiment, but were not told when that would happen.

In the experiment participants were located in the middle of a virtual train station. On the floor there was a yellow line showing the path of the virtual camera (screenshot in figure 32). Participants were not required to be on the line and its purpose was just to show the movement direction.

After the experiment, users filled one questionnaire to collect demographic data and another about simulation sickness.

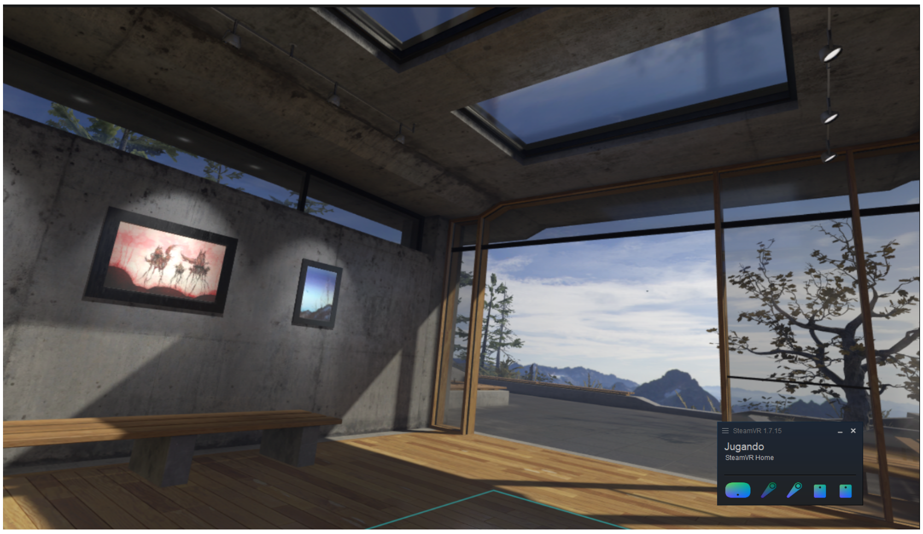


Figure 30: HTC Vive's home environment. The user moves through normal walk+teleport, can pick and throw some objects and can paint in 3D and use some other tools

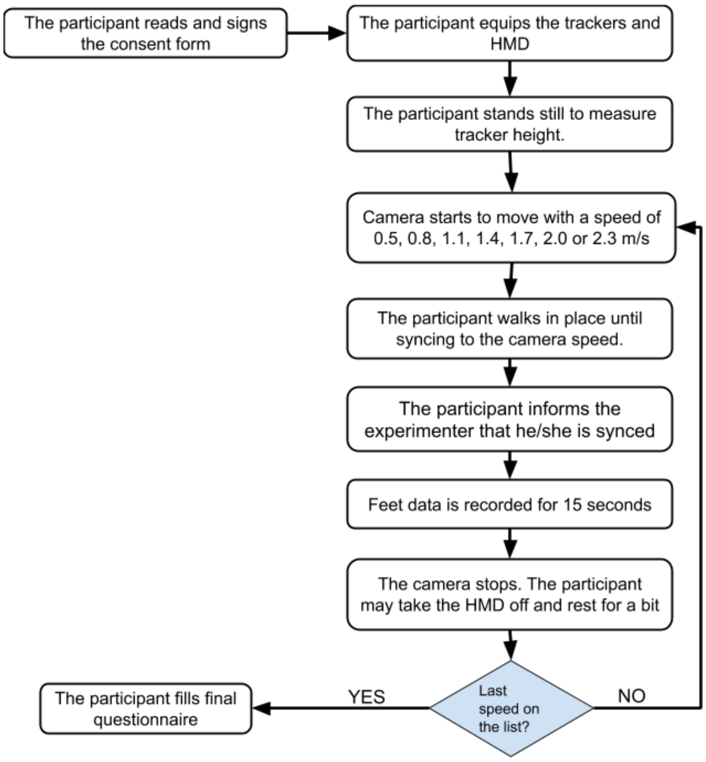


Figure 31: Experiments procedure flowchart

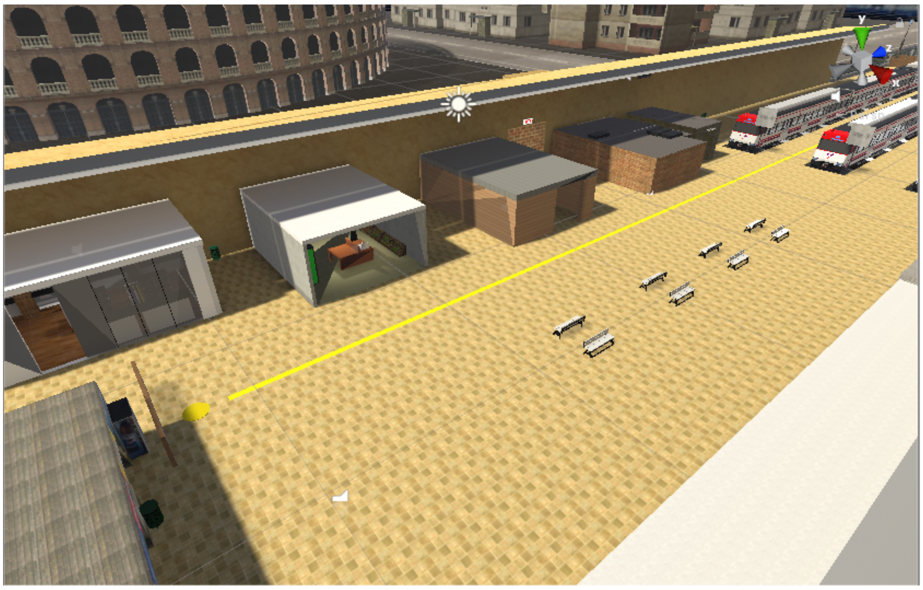


Figure 32: Experiment environment. The screenshot shows the yellow line that indicated the movement direction

6.2.2 Initial result discussion

While the data collected from the experiment still requires more study, our initial analysis of the data is already helping us to improve the system and to start to develop a new mapping function between feet movement and virtual camera speed.

Fixing the system sampling rate:

The recording of the WiP movement such as in figure 15, hinted that there was something that was causing changes in the speed to be sharper than they should be if we use the equations described in the base paper. Since the change was not too significant and it was hard to notice without looking at the chart from the recording, we put the issue aside and continued with the project. After applying the method on the data from the recording, we found that there was indeed something different in the code implementation of the equation.

The issue was found to be a sampling rate problem in the original project. In the code that was already written at the start of this thesis, the virtual camera speed was updated once every 5 frames, probably because doing so every frame resulted in some problems (please not that this is visible on the velocity graphs, but not perceptually noticeable for the user during the VR experience). Despite the recording system being changed to save values every frame at some point during this project, the 5 frame delay was kept. Changing the code to make the speed update every single frame resulted in a more smooth movement. A recording of normal WiP with the change is shown in figure 33. The recording is still not completely smooth, as it shows some small dents in the curves, but they are small enough to be near to impossible to notice.

Comparing the results to literature:

In the motivation to create a new WiP system, we discussed our belief that WiP literature was missing something to truly match normal walk to WiP. We have already started to compare the obtained results with equations from WiP literature. In [WWB10] it was stated that biomechanics studies support the following equation:

$$|v| = \left(\frac{f}{0.157} \times \frac{h}{1.72} \right)^2 \tag{7}$$

Where $|v|$ is walking speed, f is step frequency and h is subject height. The equation serves as the basis for them to only use step frequency to determine camera speed.

We used the data from 3 participants of our experiments with the same height (183cm) to eliminate the variable, and plotted the step frequency and feet speed at different camera speeds to see if only step frequency would change or if feet speed would also increase. The charts are shown in figures 34, 35 and 36. While indeed, there seems to be a relation between movement speed and step frequency, the feet speed (and thus step height) also seem to increase in a similar fashion. Another important result shown in the charts is that for each person, the importance given to step frequency and feet speed when adapting to the camera speed is different. To compare it we will use the coefficients from the trend lines, since the coefficients indicate the steepness of the slope. For user 10,

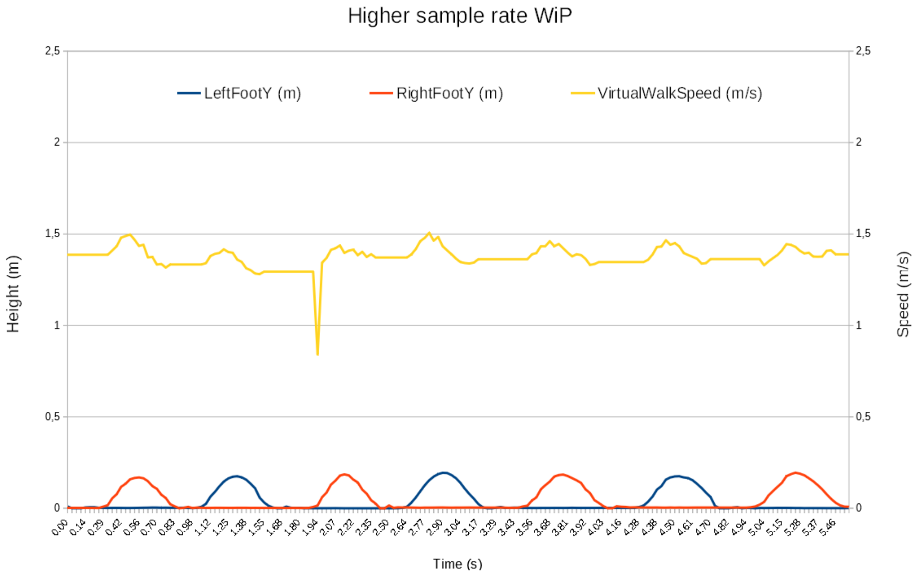


Figure 33: Normal WiP after the sampling rate adjustment. The mid-step speed changes are smoother. The sharp speed decrease is due to both feet being in the ground for too long and the system starting to slow down.

if we make the division *coefficient from feet speed trend line/coefficient from step frequency trend line* we obtain a value of $0,241$, for user 11 a value of $0,527$ and for user 12 a value of $0,155$. The values show that as camera speed increased, users increase the step frequency more than they increased feet speed, however they also show that user 12 focused more on increasing step frequency than user 11, who also gave a significant importance to increasing feet speed. Thus, as we suspected, it its likely that we will have to use both parameters to determine camera speed in our future WiP method if we really want the experience to be similar to natural walk.

Idea for the new mapping function

As we have argued through this section, the new WiP function should combine feet speed and step frequency. While in normal walk, speed can be calculated as $|v| = f \times l$ in WiP there is no step length, but there is step height. Swapping the parameter, we obtain the equation $|v| = f \times 2h$ with h being the maximum height of the step, multiplied by 2 because in a step the foot goes up and then down. But the substitution is not entirely correct, since the vertical WiP height is not the same as the horizontal normal walk length, so we have to add a factor β, resulting in $|v| = f \times 2h \times \ \beta$. With the results of the experiments, we can calculate the β since we know the speed, step frequency and step height. Our idea is to calculate the β parameter for each user during the training stage in order to adjust the virtual camera speed to closely match the user's natural movement. In that equation, a full step is required to update the speed value, and it still doesn't use feet speed. Since feet speed changes during a step, we want to add another component in the equation that uses feet speed that is also different for each user and can be updated at mid step.

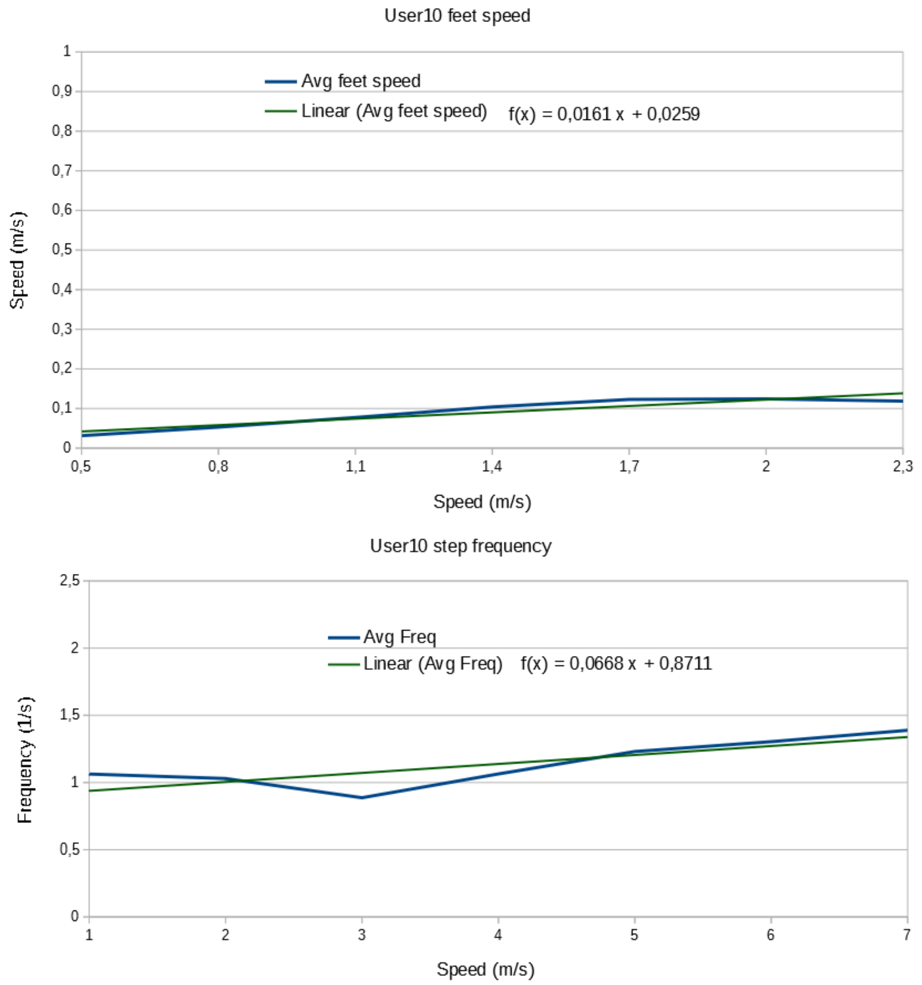


Figure 34: Data collected from user10 in the experiments. The charts shows the average speed and frequencies for each of the virtual camera speeds and a trend line see how the values change as speed increases

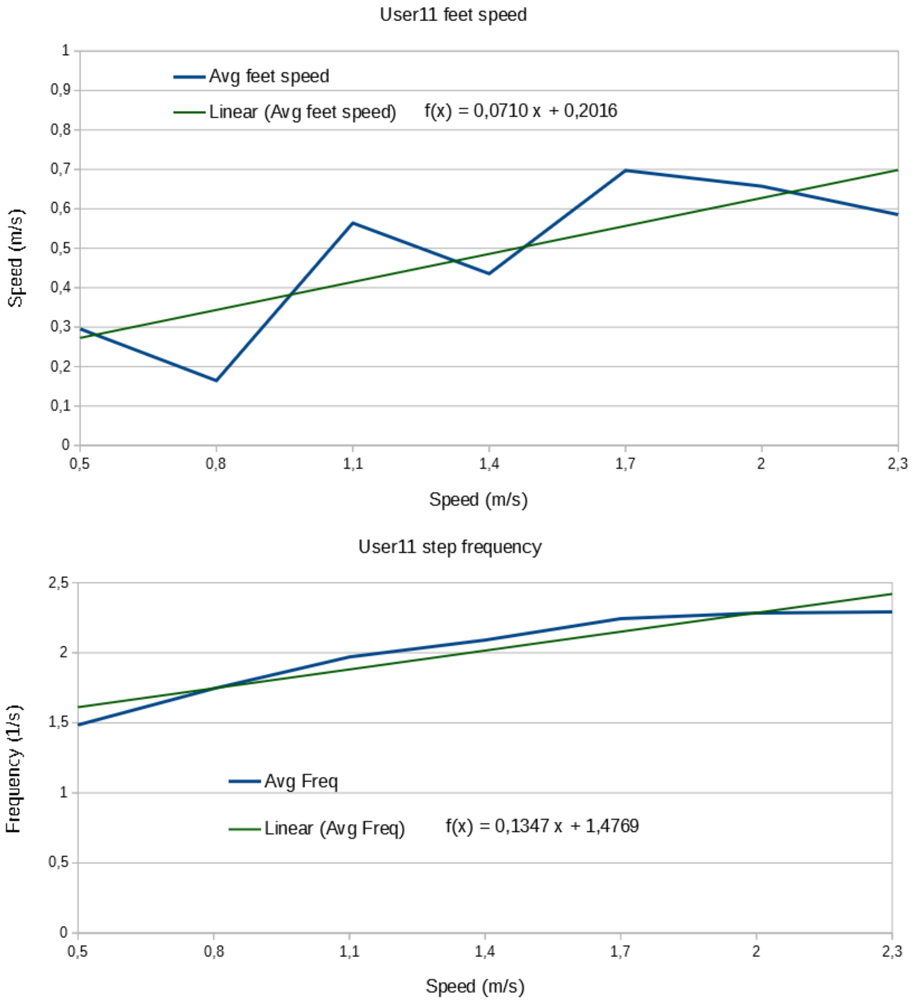


Figure 35: Data collected from user11 in the experiments.

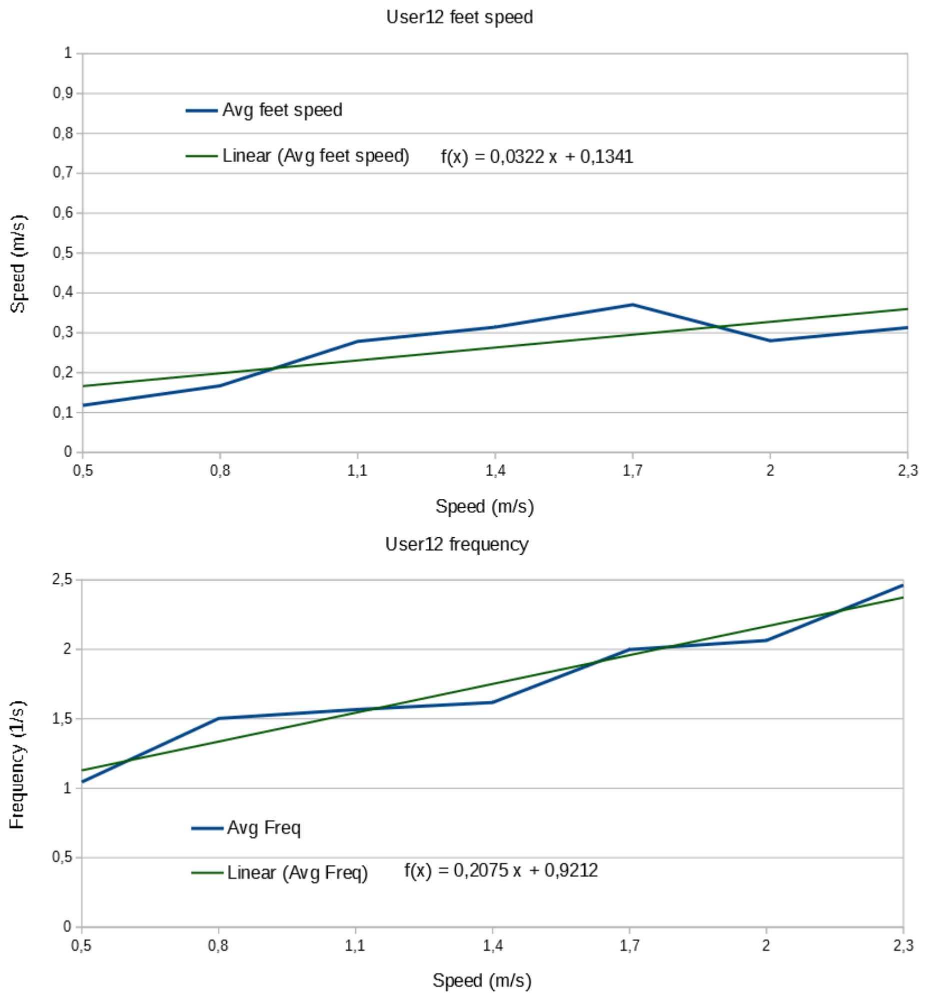


Figure 36: Data collected from user12 in the experiments.

7 Project management

The following sections show the original budget and planning created at the start of the project, we will then compare the original planning to the actual work done during the project.

7.1 Methodology

This project is mostly about investigating, trying different approaches to solve a problem, comparing them and trying to find the best solution. This means that many changes may be made throughout the project, new approaches may be tested, and some may be discarded. With this is mind, an Agile approach will be the best fit. In the agile methodology the requirements and solutions can evolve over time and it encourages rapid and flexible response to changes. It will allow us to create a solution, test it, and then trying another one or keep using that approach and investigate and develop it further.

The specific framework in which we will base our working methodology is called scrum. Scrum was designed to address complex adaptive problems. The scrum framework is very simple, it consists of what are called sprints, cycles of 1 to 4 weeks in which there is a defined goal to work towards with the tasks needed to complete it defined in the sprint backlog. At the end of the spring there is a meeting in which the stakeholders also participate to review the work done and defining the new goals. The team also meets daily in scrums to talk about their progress, adapting it towards the spring goal and to surface dependencies and impediments.

In our case, the sprints will be meetings with the project supervisors to show the progress, refine what has been done and talk about the new functionalities to be added in the next sprint. There will be no daily scrums, as the supervisors would not be able to meet me every day. Instead, I will regularly send them mails informing them of the progress and issues I'm having, and asking them for help if needed. Since they are the authors of the paper that this project seeks to further develop, the methodology used will allow me to work very closely with them so that they can help me and steer the project towards the ideas that have more potential and are more interesting to investigate.

To monitor the tasks we will use the online tool Trello. Trello allows you to define tasks, group them into lists. The tasks can have a deadline, subtasks, people in charge and other parameters. For our project, we will define the tasks to do in each sprint and we will track which ones were completed in the previous sprint. The new tasks that are needed during a sprint will also be written.

7.2 Initial Planning and budget

7.2.1 Task definition

Preliminary tasks

STP: Reading the base paper.
Dependencies: None.
Estimated duration: 3 hours.
Description: Reading the paper that this project will extend: **Smooth transitioning between two walking metaphors for Virtual Reality applications**. Understanding the key concepts and techniques that will be needed to start the project, such as Walk in Place, Virtual Reality and the senses provided by the vestibular system.

STU: Learning to use the Vive HMD and trackers, Unity and SteamVR
Dependencies: None
Estimated duration: 5 hours
Description: Becoming familiar with the main programs and devices used in this project. The task includes starting a new project in unity to understand its most important components and functionalities, setting up the HTC Vive and trackers inside Unity and learning how to configure the HMD through SteamVR.

STC: Learning C# and unity scripting
Dependencies: None
Estimated duration: 5 hours
Description: Learning the language required to program in unity, C#. Writing simple programs with the purpose of understating the differences between C# and C++ and Java. Learning the format of a unity script, some of the most important classes of the AND and how different components of the program can interact between them.

PRB: Understanding the code of the project and fixing bugs
Dependencies: ST1, ST2, ST3
Estimated duration: 20 hours
Human resources: None **Material resources:** HTC HMD, base and trackers. **Description:** Reading and understanding the code in the unity project to the implementation of the base paper. Cleaning the code with the purpose of making it more easy to read and understand. Fixing some bugs and changing parts of the code that do not match the concepts that apperar in the algorithm described in the paper.

Biomechanics method

BST: Study the biomechanics literature
Dependencies: ST1, ST2, ST3
Estimated duration: 40 hours
Description: Studying the most relevant biomechanics features of the walk cycle for normal walk and WiP. Deciding what parameters of the natural walk of a person should be computed in order to match each stage of normal walk to those that appear during WiP

BI: Implementation of the biomechanics method
Dependencies: BST
Estimated duration: 30 hours
Description: Writing the code of the biomechanics method. Programming the training stage. Studying and coding a function to map the feet movement of a person during WiP to the virtual camera movement.

Deep learning method

MST: Study of unity's learning methods
Dependencies: BST
Estimated duration: 20 hours
Description: Studying the options that Unity offers regarding deep learning and reinforcement learning and understanding them in dephth to determin how they could be applied to our problem.

MI: Implementation of the machine learning method
Dependencies: DST
Estimated duration: 20 hours
Description: Programming the code that extracts the parameters chosen from the trackers to train unity. Running the code to extract enough data from a person walking normally.

Testing

Tasks required to test the project's usability with other people, both in short cycles to test the new functionalities and in a final experiment to extract the results of the project

PiE: Pilot experiments performed with ViRVIG members
Dependencies: None
Estimated duration: 10 hours
Description: Usability tests performed regularly to test the changes. The test subjects will be the thesis supervisors and other ViRVIG members. There will not be a statistical study of the results, but the feedback will be taken into account to modify the code.

EP: Planning the final experiments to test the developed methods
Dependencies: None
Estimated duration: 15 hours
Description: Planning the experiments that will be performed. Including: the procedure, virtual environment that will be used,picking participants and creating the surveys that participants will have to fill. Defining what formulas and statistic methods will be used to extract the conclusions from the results of the tests.

EXP: Performing the experiments
Dependencies: IT, EP, DSI, BSI
Estimated duration: 10 hours
Description: Performing the experiments defined in EP and saving the results. The experiments will require to meet with different persons, introducing them to VR if necessary, performing the tests, and handing them surveys to know how they felt throughout the experiment.

Project management tasks

DEF: Defining the project and its exact goals,scope,planning,etc.
Dependencies: None
Estimated duration: 27 hours
Description: Initial documentation of the project. Defining the key aspects of the project such as the scope, goals, methodology, planning and budget.

Required to start any project.

FREP: Writing the thesis report
Dependencies: None
Estimated duration: 60 hours
Description: Writing the thesis report of the whole project.

MEET: Meetings
Dependencies: None
Estimated duration: 14 to 20h
Description: Meetings with the thesis supervisors every two weeks (Scrum methodology). Each meeting takes about 2h. This task has a variable length since more meetings may be required depending on the progress and problems of the project.

7.2.2 Task summary

Task	Description	Estimated duration(h)	Dependencies
DEF	Defining the project and its exact goals,scope,planning,etc.	27	-
MEET	Meetings	20	-
FREP	Writing the thesis report	60	-
PiE	Pilot experiments performed with ViRVIG members	10	-
STP	Reading the base paper	10	-
STU	Learning to use the Vive HMD and trackers, Unity and SteamVR	5	-
STC	Learning C# and unity scripting	3	-
PRB	Understanding the code of the project and fixing bugs	5	STP STU STC
BST	Study of the biomechanics literature	40	STP,STU,STC
BI	Implementation of the biomechanics method	30	BST
MST	Study of unity's learning methods	20	BST
MI	Implementation of the machine learning method	20	DST
EP	Planning the final experiments that test the developed methods	15	-
EXP	Performing the experiments	10	PiE EP MI BI

7.2.3 Budget Table and explanation

Activity	Cost€	Notes
Defining the project and its exact goals,scope,planning,etc.	661.76	
Meetings	2007.47	
Writing the thesis report	1470.59	
Pilot experiments performed with ViRVIG members	245.10	
Reading the base paper	73.53	
Learning to use the Vive HMD and trackers, Unity and SteamVR	122.55	
Learning C# and unity scripting	122.55	
Understanding the code of the project and fixing bugs	490.20	
Study of the biomechanics literature	980.39	
Implementation of the biomechanics method	735.29	
Study of unity's learning methods	490.20	
Implementation of the machine learning method	490.20	
Planning the final experiments	367.65	
Performing the experiments	245.10	
TOTAL CPA	8502.57	
HTC vive HMD	50	
2 Vive trackers	20.95	
PC	61.90	
Transport	125.4	
Electricity	40	
Space	40	
TOTAL GC	338.26	
TOTAL GC + CPA	8840.82	
CONTINGENCY	1326.12	15% of Total CPA + Total GC
Mappings for slow and fast WiP required	367.65	50% risk
Extra time biomechanics method	49.02	20% risk
Extra time machine learning method	49.02	20% risk
Extra WiP method developed	61.27	5% risk
Redirect method developed	220.59	30% chances
TOTAL INCIDENTIALS	747.55	
TOTAL BUDGET	10166.95	

Staffing costs Because I am doing research, we will assume that I am working under "Personal d'ajut a la recerca" contract. The contract cost 42.000€ annually for a workload of 37,5 weekly hours. The price includes paying taxes and social security. As for the time my supervisors will spend in meetings with me, we will use the amount it costs the UPC to have a teacher, around 65.000€ paying taxes and security.

Material and transport The cost of buying a new HTC vive is 600€[HTC]

The estimated useful life of the headset is 4 years, keeping in mind that its life can end either for becoming outdated of due to a malfunction. The trackers cost 120€ . They are more resistant to breaking and less likely to become outdated. Therefore, the estimated useful life is 7 years.

The PC needs to be of at least a modern middle to high end in order to play virtual reality applications with the necessary resolution and screen refresh rate. We estimate the total cost of the PC to be around 1300€ taking into consideration the rest of peripherals, namely the screen, keyboard and mouse. We also assume it was bought as a whole and not in separate components. The estimated useful life of high end PC is around 7 years.

Regarding the transport, the price shown is the cost of me going to ViRVIG and Fib for around 4 months.

7.2.4 Budget control

Risks

The most important risk found in the project was the need to develop a new mapping function for walking in place slow and fast. Creating the functions would require 30h. This task is the core element of our work, so it cannot be removed in case of failure. We must find a solution even if the improvement compared to the current method are not very significant. We would like to innovate with learning techniques to find the best mapping function, and this is the step where we may not succeed and need to go back to a manual mapping.

So in the unlikely case that we require new functions for walking in place slow and fast, and 20h are not enough to create the new function with reinforcement learning techniques, will only implement a mapping that can adjust the value of the parameters to the biomechanical information gathered. in order to not further increase the budget.

The risks of the project would only have an effect on working hours. There would not be a need for more equipment or other extra costs. Thus, our main method to control the budget is tightly related to controlling the schedule of the project. Through the project, we will track the extra hours dedicated to each task, and the extra tasks needed. If in the first half of the project we have spent more than 30 extra hours, we will avoid doing any extra task even if they were planned. Instead, we will explain the need to reduce the scope in the final report. This way we assure that we do not increase the budget too much.

Extra costs In the case that the project advances smoothly and we finish our objectives before the scheduled finish time, we considered the option to increase the scope. The secondary objective would be to develop a method to redirect the user when it gets close to the edges of the physical room. The estimated total duration was 30h. While developing the method would increase the budget, it would not be by an excessive amount, 187.5€. If we are able to develop this extra, it would mean that we had little to no extra costs in the rest of the project. Thus, we would have money to spend on the extra feature, and it would be worth the price.

7.3 Changes in planning

During this project we followed the scrum framework described in the planning. There were regular meetings with my tutors, and internal tests were performed every time a new feature was added. Thanks to choosing an agile methodology we were able to add and remove goals, and invest more or less time in adding a feature or investigating as we saw necessary. However we never lost our original main goal, all our efforts were directed towards developing a biomechanics based WiP system. But still, looking back at the original tasks planned can show us whether the project advanced and concluded smoothly or had a lot of changes and problems.

Indeed, the biomechanics based WiP system was developed, however at the start we underestimated the scope and time needed to develop it. RiP and slow WiP at first were thought to be small tasks that would only require to adjust the original WiP system, however as we investigated more we found them to be interesting due to how our system handled the movements and how little research there was about RiP. In the end developing the new modes required a lot of experiments, programming and adjustments, but we believe that the time was well invested, and that the project would have been worse with RiP and slow WiP only half developed and left as future work.

As for the Deep learning method, as we investigated and started to see how we would develop it, we noticed how time consuming it would be. As time went on, and the WiP system became more complex, we realized that if we had tried to develop a similar system but using machine learning and we wanted get results the project would not finish in time. Thus, since we were obtaining results from the biomechanical method, we decided to use the time left to improve it even more instead of trying to build a new system.

Even if it not in the original planning, the design of the training stage also took more time than expected. As we started performing experiments and recording data we saw that there was a lot of potential and kept changing the parts of the training stage and the way the camera moved almost each week.

The time frame that we initially wanted to use on the user study was in the end spent on performing an experiment and collect feet data. While the results are not used to develop a new system in this project, will be much more useful in the future.

The project still obtained results and finished on time. The time spent on the project remained about the same as originally planned and the budget didn't increase, we just redistributed the time available to what we thought was best at any moment, which is expected when working under an agile methodology.

8 Sustainability report

8.1 Survey auto evaluation

The survey (goo.gl/kWLMLE) asks about my knowledge regarding sustainability and my ability to develop a project that takes into account the different dimensions of sustainability. I think the results of the survey were generally positive. There were a lot of questions about my knowledge regarding sustainability, economy and project management. I positively answered most of those questions, since during the bachelor I have been taught about those subjects in some courses. The survey had a few questions about my ability to analyze different aspects of the project, such as the impact in society, economy, ethics and environment. I believe I can think critically when I'm in a project or I'm trying to analyze it. I can identify the different aspects of the project that affect sustainability and evaluate the impact on the environment and society. However, I am a bit lacking when I have to apply the principles myself. I am about to end a bachelor and I have little experience in the working world. While I know how to evaluate a project, once I'm actually inside one I don't think that at this moment I would be able to make sure it's sustainable. I lack the experience to know what methodologies and what techniques we would have to use to develop a sustainable project. I could a bit help giving some ideas when talking about the sustainability aspects of the project, but I would not have the initiative to talk about these topics. If I had to start a real project on my own, I believe I would need a little help to make it sustainable.

8.2 Economic Dimension

8.2.1 Economic cost of the project

The cost of developing the project is shown in the budget table. While tasks performed in the project changed, the total development time remains aproximately the same, and therefore the budget of the project remains the same.

This project does not aim to generate any kind of revenue, instead its goal is to help other people that want to develop a WiP system. As a result, for this project to be economically profitable, the only condition is that the research performed and the results obtained are used on the developement of some aplication or in another person's research.

8.3 Comparison to other solutions

In our solution, we aim to find a middle ground between cost and performance. The setup is not cheap, but its neither so expensive that only a few people and research groups could use it. We believe the results it can produce are worth the economic cost.

There has been research on both ends, using cheaper and more expensive technology. On the side of cheaper technology, there are methods for navigating using a HMD consisting of a phone inside a headset or HMD that do not need a pc or any other component to work. Most common locomotion techniques use a hand held controller to move the virtual camera. Methods for WiP use the head bobbing to detect steps. However they are limited, because only the head orientation can be used to determine walking direction, you can not walk and look at different directions. Also, the lack of positional trackers makes it impossible to use for normal walk.

On the other side, some techniques use more complex setups. Some solutions use even more trackers to more accurately detect movement, usually positioned in the knees,legs, or chest. There are solutions that use a small room with projectors, such as a CAVE. And others that use a whole large room to create a bigger virtual room. Our project aims to achieve the same level of immersion and precision as those bigger, more expensive setups.

8.4 Environmental Dimension

8.4.1 Environmental cost of the project

In order to estimate the cost of developing the project we will use the following values:

Activity	Energy usage
PC	0,2 kWh
Screen	0.015 kWh
Vive setup	0.016 kWh [Red]

The whole project will make use of the PC and screen. Some tasks use the Vive for about 50% of the time, And the economic cost of the development of the project will be of about 62,59kWh.

The finished project itself will not have a direct environmental impact. The results wil allow other people to develop other projects or applications. Nevertheless, we hope that since the results will be used in other WiP systems, not needing to create a completely new RiP system or training stage will reduce the economical impact of other projects.

8.4.2 Comparison to other solutions

In the environmental dimension we find a similar case as in the economic one. Our solution does not attempt to improve other systems in this dimension, but to find a balance between the cost and the usability. Using a phone or a simple HMD with low energy usage will have a lower environmental impact. However using more devices or other, bigger setups will have a greater energy usage.

8.5 Social Dimension

8.5.1 Personal growth

Initial thoughts In the first phase of the project, while investigating, I realized the scope of the project and the investigation regarding VR and specifically WiP. Looking at different papers with various navigation techniques, I saw that there is still a lot of investigation needed in this field. For modern videogames and virtual environments that do not use VR there are some well established rules regarding movement. The joystick is always the best method to move the camera, and in some applications, there is automatic camera movement. In VR, full room setup seems to be the optimal way to move in a small area. However when we need to move in a large environment, there is no consensus regarding what technique to use. The added challenge of reducing motion sickness due to the use of a HMD makes the task much more difficult than when interacting with just a flat screen.

When this project concludes, I feel like I will have a greater understanding of the challenges of navigating a virtual environment and I hope will be able to continue investigating this field or that I will be a useful member of a team of developers of Virtual Reality Applications. While we've had modern HMD for a while, they are still at an early stage of development, and when in the future they become more widely used I hope that I will be ready to keep innovating and creating new games and applications.

Final thoughts

After finishing the project I think that I do have a much better understanding of navigation in virtual reality and the many problems it still has. After seeing many people react to different camera movements I was able to see how different they react, for some motion sickness is not a problem at all, while for others even small camera movements that they don't expect, or in our case that doesn't match the feedback they expect from their movements, cause motion sickness. In a time where videogames are starting to commit to being more accessible , VR is still quite behind and still doesn't even have a universally accepted good navigation method that doesn't cause motion sickness nor disorientation. I think that now I'm able to keep investigating to help improve it.

8.5.2 Impact of our project

Motion sickness is a problem of VR and a barrier for some people to start using it. With the techniques developed in this project we hope to reduce the likeliness of suffering from motion sickness compared to other solutions, thus improving the well-being of VR users. The techniques developed will be available for anyone to use. Developers will be able to use our research into their VR applications, and thus it will have an impact on the quality of the final product. The project could also be used by other people to create their own methods of movement, allowing for further research to be done. Because there is still not a standard way to move in large virtual environments, more research and techniques are required to some day find one method that can be universally used.

After finishing the project, we believe we have helped reduce the motion sickness problems of WiP and laid the groundwork for future investigation. The problem is still not solved and there are probably still a lot of improvements that can be made, but our results are step in the right direction to improving VR systems and allowing anyone to use them without feeling sick. Since VR has a lot of uses, our efforts can be used by a wide range of people, including fields outside of entertainment, where people who use VR are likely not familiar with graphical applications and are more likely to suffer from motion sickness.

9 Conclusions

During the development of this project we have improved current WiP systems by adding functionalities to allow users to interact with the environment in different ways while also reducing motion sickness. In the first stage we successfully improved the original project by adding small modifications that made the system more responsive but also smoother.

We created a RiP and slow WiP system, and got results that showed that Running in Place is significantly different to Walking in Place and require a different movement to virtual camera mapping function. The combined system increased immersion by allowing to move large distances with in-place motions and also allowed navigation in crowds or other environments where slow movements are required. We found a relation between slowing down during normal walk and during WiP or RiP that can help reduce motion sickness.

During the creation of a training stage we discovered a way to manipulate the user's WiP frequency. Some functions were created in order to adjust certain WiP and RiP parameters to each user's natural movement, thus improving their experience when using the system.

Finally we succesfully conducted an experiment to collect feet data that will be used to develop a new WiP method.

10 Future work

While a lot of experiments were performed, and a lot of aspects of WiP were studied, some work was left for a future projects due to time constraints.

Developing a WiP and RiP system but using some form of machine learning was the biggest goal of the project that we were not able to reach. The scope of developing the system was deemed to be much bigger than expected. We believe that using machine learning to let the computer either decide if we are performing WiP or RiP or even decide the output virtual camera speed is worth investigating, and has enough depth to require an entire thesis. It would require to find a way to generate training data for the machine, either by recording or by generating artificial data, creating a correct output, and making sure the computer is trained properly. During the development of this project, unity's ML-Agents [Unib] were looked into, and some tutorials and tests were performed. We saw how powerful the tool can be, but using it correctly requires an in depth understating of how the plugin and algorithms it uses work. The main problem we found was the amount of data it required for even the most simple artificial intelligence, which would be very hard to manually generate in our case. Thus it would be required to either extract large amounts of data from real people or to find a way to generate walking data that is very similar to real walking, which is probably a huge challenge by itself.

A side optional objective of this project was to find a natural way to redirect the user's physical displacement during WiP to avoid that they go out of the physical play area and are and hit some person or piece of furniture. After finishing the project, we believe that there is a very real need to redirect user's physical displacement during WiP. A lot of tables were hit during the internal tests, and during the experiment all users had to be told at some point to step back a few steps to avoid hitting the wall. The problem lies in precisely the strength of WiP: a physical movement generates a visual output. While this solves the mismatch of movement and visual flow, it also makes small physical displacements almost impossible to notice. If you were Walking in Place without a headset you would notice the movement, however with the headset and the WiP system, the physical displacement feedback is added onto the visual flow generated by the computer, and it becomes noise. Redirected WiP is not something new [Raz+02] but we believe more research and redirect methods are required in order to increase immersion.

At some points in the project we got some interesting results that were used to improve the system, but we were not able to further research. During the creation of the RiP mode we found a relation between virtual camera speed and stopping latency. The results seem very clear, since it was really easy to get strong motion sickness when latency was was too high. Further studies to find a more exact relation could help further reduce motion sickness. During the creation of the training stage, we think were able to manipulate the user's step frequency. We believe that it would be interesting to further studying the matter, testing how should the camera be to make the user synchronize, and whether or not it works for everyone, so it can be used to design new training stages. Further investigating it would allow to set a constant in the user's movement and focus on other biomechanical parameters.

References

[Bob60] AC Bobbert. “Energy expenditure in level and grade walking”. In: *Journal of Applied Physiology* 15.6 (1960), pp. 1015–1021.

[SSU95] Mel Slater, Anthony Steed, and Martin Usoh. “The virtual treadmill: A naturalistic metaphor for navigation in immersive virtual environments”. In: *Virtual Environments’ 95*. Springer, 1995, pp. 135–148.

[TDS99] James N Templeman, Patricia S Denbrook, and Linda E Sibert. “Virtual locomotion: Walking in place through virtual environments”. In: *Presence* 8.6 (1999), pp. 598–617.

[Raz+02] Sharif Razzaque et al. “Redirected walking in place”. In: *EGVE*. Vol. 2. 2002, pp. 123–130.

[DRT07] Frank H Durgin, Catherine Reed, and Cara Tigue. “Step frequency and perceived self-motion”. In: *ACM Transactions on Applied Perception (TAP)* 4.1 (2007), p. 5.

[FWW08] Jeff Feasel, Mary C Whitton, and Jeremy D Wendt. “LLCM-WIP: Low-latency, continuous-motion walking-in-place”. In: *2008 IEEE Symposium on 3D User Interfaces*. IEEE. 2008, pp. 97–104.

[Wen10] Jeremy D. Wendt. “Real-walking Models Improve Walking-in-place Systems”. AAI3418628. PhD thesis. Chapel Hill, NC, USA, 2010. ISBN: 978-1-124-17122-7.

[WWB10] Jeremy D Wendt, Mary C Whitton, and Frederick P Brooks. “Gud wip: Gait-understanding-driven walking-in-place”. In: *2010 IEEE Virtual Reality Conference (VR)*. IEEE. 2010, pp. 51–58.

[MPL11] Maud Marchal, Julien Pettré, and Anatole Lécuyer. “Joyman: A human-scale joystick for navigating in virtual worlds”. In: *2011 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE. 2011, pp. 19–26.

[NSN14] Niels Christian Nilsson, Stefania Serafin, and Rolf Nordahl. “Establishing the range of perceptually natural visual walking speeds for virtual walking-in-place locomotion”. In: *IEEE transactions on visualization and computer graphics* 20.4 (2014), pp. 569–578.

[RO16] Lisa Rebenitsch and Charles Owen. “Review on cybersickness in applications and visual displays”. In: *Virtual Reality* 20.2 (2016), pp. 101–125.

[Jul+18] Arthur Juliani et al. *Unity: A General Platform for Intelligent Agents*. 2018. arXiv: `1809.02627 [cs.LG]`. URL: `https://github.com/Unity-Technologies/ml-agents`.

[Liu+18] James Liu et al. “Increasing walking in VR using redirected teleportation”. In: *The 31st Annual ACM Symposium on User Interface Software and Technology*. ACM. 2018, pp. 521–529.

[RP18] Alejandro Ríos and Nuria Pelechano. “Follower Behavior in a Virtual Environment”. In: *Virtual Humans and Crowds in Immersive Environments (VHCIE)*. Reutlingen (Germany), Mar. 2018.

[Han+19] Sara Hanson et al. "Improving Walking in Place Methods with Individualization and Deep Networks". In: (2019), pp. 367–376.

[Jan+19] Omar Janeh et al. "Infinity Walk in VR: Effects of Cognitive Load on Velocity during Continuous Long-Distance Walking". In: *ACM Symposium on Applied Perception 2019*. ACM. 2019, p. 9.

[SRP19] Isadora Salvetti, Alejandro Rios, and N Pelechano. "Smooth transitioning between two walking metaphors for Virtual Reality applications". In: (2019).

[Ber] Jimmy van den Berg. *DigitalFilters*. URL: `https://github.com/overlord1123/LowPassFilter` (visited on 12/30/2019).

[HTC] HTCCorporation. *Vive Product Description*. URL: `https://www.vive.com/us/product/vive-virtual-reality-system/` (visited on 12/29/2019).

[Red] RedditUser:TealcLOL. *Vive Power Draw Test Results*. URL: `https://www.reddit.com/r/Vive/comments/51ir1h/vive_power_draw_test_results/` (visited on 12/31/2019).

[Unia] UnityTechnologies. *Unity engine webpage*. URL: `https://unity.com` (visited on 12/29/2019).

[Unib] UnityTechnologies. *Unity ML-Agents plugin*. URL: `https://github.com/Unity-Technologies/ml-agents` (visited on 12/31/2019).

[Val] ValveCorporation. *SteamVR page in the Steam storefront*. URL: `https://unity.com` (visited on 12/29/2019).

[ViR] ViRVIG. *ViRVIG webpage*. URL: `https://www.virvig.eu` (visited on 12/29/2019).