VRNav: A Framework for Navigation and Object Interaction in Virtual Reality

Sampath Shanmugam, Vaishak R Vellore, Sarasvathi V
Department of Computer Science Engineering
PESIT South Campus, Bangalore,
Karanataka, India
Email: sampath_shanmugam@outlook.com, vaishak.vellore@gmail.com, sarsvathiv@pes.edu

***Abstract*— Virtual Reality(VR) is currently a trending area of research and experimentation for both the industry and the academia. With the introduction of VR headsets like Google Cardboard, HTC Vive and Samsung Gear VR, anyone with a smartphone today can experience the magic of VR. Any good VR experience must allow a user to both navigate and interact with objects within the virtual world. However, neither of these two goals is easily achievable. Often, the use of external controllers and sensors reduces both immersion and freedom. The primary purpose of this paper is to build a framework that allows users to both navigate and interact with objects in virtual worlds, without having to use a controller or any other similar touch input based mechanism used by existing implementations, while keeping in mind two important points. First, the setup must be designed in such way that these goals can be achieved using only a smartphone and a VR head mounted display, with no other external hardware. Second, the setup must be kept as cost effective as possible. Thus, anyone, from anywhere, with a smartphone and a low-cost Google Cardboard headset should be able to experiment with VR, and deploy our framework in any virtual world that they may design and build.**

***Keywords- Virtual Reality, Virtual Environment, Walking in Place, Walk Detection, Head Mounted Display, Software Development Kit*.**

I. INTRODUCTION

Virtual Reality (VR) is defined as "an immersive and realistic simulation of a 360-degree three-dimensional environment, which has been created and built using interactive software and hardware, and controlled by the movement of the body", according to Wikipedia [4]. VR is typically experienced through a head mounted display, that provides a visual output of the virtual world, and may also provide an audio output for the user to experience sound effects.

VR applications are usually designed for a specific use-case and include virtual worlds are modeled based on the real world. As an example, consider a VR app that allows the user to explore the Taj Mahal. The user would thus be able to get a first-hand experience of exploring the monument. These virtual worlds could also contain objects that users can interact with. In the case of the Taj Mahal example, there could be a camera that could be used to take pictures.

II. MOTIVATION

A. Objectives

The primary objective of this paper is to build a framework that allows users to both navigate and interact with objects in virtual worlds, without having to use a controller or any other similar touch input based mechanism used by existing implementations, while keeping in mind two important points. First, the setup must be designed in such way that these goals can be achieved using only a smartphone and a VR head mounted display, with no other external hardware. Second, the setup must be kept as cost effective as possible. This paper incorporates the concept of walking-in-place to achieve navigation, wherein the user jogs on the exact same spot where they are standing, without moving forward, or in any other direction [1]. Walking in Place(WIP) is used to detect user activity, and effectively, the speed at which the user jogs in place is the speed at which the user moves in the virtual world. In addition to being able to navigate virtual worlds, the proposed framework implements a mechanism to allow users to interact with objects in the virtual world.

As a result, anyone, from anywhere in the world can virtually sightsee any other place, virtually play any sport or even virtually live another life, from their own room.

B. Related Work

Virtual Reality has been around for quite some time now and a lot of research and effort is being put into this field which is expanding at an alarming rate. If we survey existing VR systems, orientation tracking (support for 360 degree viewing of the virtual world) alone is supported [4]. Positional tracking is not commonly implemented. Even in those systems that support navigation, external sensors or controllers are used to track body movement [4]. Most existing implementations use such mechanisms for navigation primarily because when the user puts on their VR headset, they are no longer able to see the real world around them, thereby restricting the possibility for any movement [5]. But this approach has its limitations, because the setup as a whole becomes complex and expensive, in addition to reducing the immersion experience. Now, think of way in which a person can actually walk in and around the virtual world, experiencing all this by whilst being in a single place.

978-1-5386-2044-1/17/$31.00 ©2017 IEEE

C. *The need for VRNav*

The key goal of the paper is to build a system which allows navigation in VR worlds, based on a user's movements in the real world. As we have discussed in the previous section, existing systems lack this type of functionality. To achieve this goal, we use only a smartphone and a head mounted display, like Google Cardboard, keeping cost at a minimum. This effectively removes the need for any external sensors or even controllers for locomotion in the virtual world also making it easier to use. In addition to navigation, it is necessary to have some means to interact with objects in the VR world. We achieve this through head gestures. Thus, our framework essentially has four key components; step detection, virtual locomotion, object interaction and head gesture recognition. Potentially, this framework can be used in a variety of VR applications like learning, fitness, E-commerce, to name a few. The paper is described in detail in the following chapters which happens to put light onto some important points which will help in understanding the system as a whole.

III. PROPOSED FRAMEWORK

In the process of building our VR framework, there are a few essential goals that need to be achieved. Each of these goals is achieved through four components:

• **Step Detection:** Detect user's walking-in-place steps via the phone's accelerometer, and keep a count of the number of steps.
• **Virtual Locomotion:** Translate the user's WIP steps in the real-world to appropriate movements in the VR world, while computing instantaneous speed and average speed.
• **Object Interaction:** Allow the user to interact with objects in the VR world with the help of a mechanism to detect objects and trigger actions.
• **Head Gesture Recognition:** Recognize rotational movements of the head, like nodding of the head, to trigger events in the VR world.

In addition to the above primary components of the framework, a new component to is required within which the framework will be tested. This is called the Virtual Environment component.

A. *Step Detection:*

Steps are detected with the help of the phone's accelerometer as input, and some processing done at the application end. An accelerometer is an inertial device that measures acceleration along the three axes X, Y, Z. The acceleration value being measured is in terms of meters per second^2 (m/s^2). Values are sampled continuously by the accelerometer along the three axes, say acc_x, acc_y acc_z using which a combined magnitude is computed.
We are interested in fluctuations in this combined magnitude, which indicates some activity. So, we subtract the mean of acceleration values from the current input value, to remove any constant effects, like gravity. Following this, the number of peaks is computed, which then gives us the number of steps.

Algorithm:

1. Start.
2. Input accelerometer data along the X, Y, Z axes. These are acc_x, acc_y and acc_z.
3. Compute the magnitude of the current acceleration follows:
$$acc_{cur} = \sqrt{acc_x{}^2 + acc_y{}^2 + acc_z{}^2}$$
4. Update magnitude of average acceleration, acc_{avg}, with the current acceleration value, acc_{cur}.
5. In order to perform step detection, fluctuations are necessary in the accelerometer signal. To detect these, constant effects such as gravity can be eliminated as follows:
$$acc_{nog} = acc_{cur} - acc_{mag}$$
6. Check if acc_{nog} is a peak. Every peak is a step. The number of peaks gives the number of steps.
7. Stop.

B. *Virtual Locomotion:*

Now that we are able to detect steps, the next big challenge is to design an appropriate scheme to translate movements in the real-world to movements in the VR world. This includes two broad goals in the VR world:

1. Computing speed of movement.
2. Translation along the appropriate axis.

Two different types of speeds are computed in the system; instantaneous and average. In order to compute the instantaneous speed, we begin by first computing the time taken between two consecutive steps. We then compute the instantaneous speed by taking the reciprocal of this time computed. If the time taken between steps exceeds 1 second, then the speed is set to 0, because the user is at rest. In order to compute the average speed, we divide the total number of steps by the amount of time taken to cover those steps. Average speed is useful to analyze user behavior and usage of the system.
As for the next goal, now that we have the speed of the movement, it is a question of which direction to perform the movement in. For this, we keep track of the user's gaze direction, which is nothing but the direction in which the user is looking. We allow movement along both the X and Z axes, while keeping the Y axis position constant, to prevent unwanted actions like moving skywards or falling downwards. User's gaze direction is detected via rotational/orientation tracking about the Y axis, which provides for a full 360° rotation experience.

C. *Head Gesture Recognition:*

978-1-5386-2044-1/17/$31.00 ©2017 IEEE

While we were experimenting with other methods of providing input to the system, we came across the concept of head gestures. The head, because is attached to the neck, has six degrees of freedom; three rotational and three translational. Given that the VR system is finally deployed in a Head-Mounted Display, it makes sense to try recognizing gestures of the head. Translational tracking of the head could be difficult to isolate from translational tracking of the body, as is there is a limited scope for the translation of the head, thus it would be ambiguous. We opt to work with rotational tracking instead.

Based on experimentation, we discovered that rotation along the X and Y axes were the most effective and unambiguous. This allows us to work with two independent head rotation gestures.

1) *Gesture 1*: Move up and down/Rotate about X axis (Yes).
2) *Gesture 2*: Move left and right/Rotate about Y axis (No).

Now that we have two gestures that can be recognized by the system, we can configure each of these to perform a trigger certain events based on the virtual environment of application. As an example, in the E-commerce application that we built as a proof of concept example, we configured Gesture 1 to work as a way to confirm a product purchase, while Gesture 2 was configured to work as a way to reject product purchase.

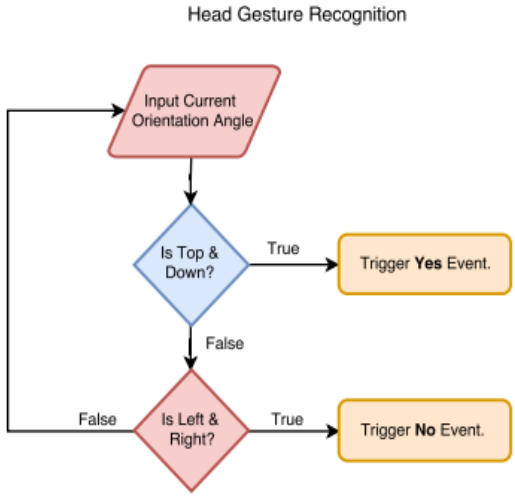

Figure 1. Head Gesture Recognition Flowchart

The Figure 1 shows a flowchart for Head Gesture Recognition. The system is able to detect rotation along the X/Y axes and the current orientation angle is noted down. If the gesture is top to down then an event which specifies a 'Yes' is triggered. If the gesture is left to right then an event which specifies a 'No' is triggered and the next gesture is recognized.

D. Object Interaction:

Movement in the VR world has been taken care of, thus the navigation part of the framework is taken care of. However, we realized the need to be able to interact with objects in the VR world, as an experience with just navigation seems incomplete. Given that we wanted to eliminate the use of any controllers, which may kill the immersion experience, we decided to experiment with a hands-free object interaction system.

We begin by keeping track of the user's gaze direction, as discussed previously. However, this time, we keep track of rotation about all three axes. In order to allow the user to specifically focus on an object, we implement the concept of a reticle, which is nothing but a small pointer, much like a cursor, that moves as you look around. The user can thus choose to interact with an object by first focusing on the object via the reticle.

However, just because a user looks at an object, it does not necessarily mean that they want to interact with the object. So to work around this issue, we introduce a timed radial progress bar that keeps track of the amount of time for which the user focuses on an object. If it exceeds five seconds, then we trigger an action associated with the object. Thus, an effective hands-free navigation system has been designed.

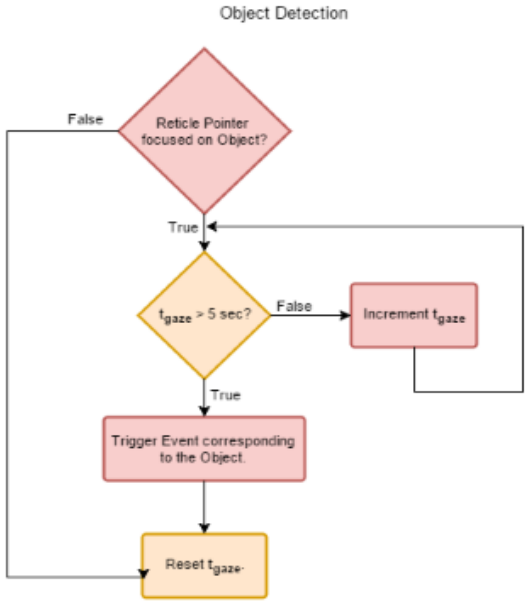

Figure 2. Object Interaction Flowchart

The Figure 2 shows a flowchart for Object Interaction. A reticle is able to detect the objects that are present in the virtual world. Using the time gazed progress bar along with the reticle pointer we are able to manipulate objects or trigger necessary actions. Upon gazing at the object for more than 5 sec, an event can be triggered and this object interaction takes place.

978-1-5386-2044-1/17/$31.00 ©2017 IEEE

E. Virtual Environment:

Although much of the VR content out there today is used solely for the purpose of Entertainment, VR can be applied to many spaces like Fitness, E-Commerce, Learning and Architecture. Since we have built a comprehensive VR framework in this paper, we decided to deploy it in two proof-of-concept virtual environments: Fitness and E-Commerce.

1) Fitness: We built a Treadmill VR application that keeps track of your fitness.

2) E-Commerce: We built a VR application that attempts to emulate the future of online shopping.

These two virtual environments give the user a chance to understand both the functionality, and the power of an immersive VR system. Our framework allows both navigation and interactions with objects in the VR world.

This goal of this Treadmill app is to simulate exercise on a treadmill, while users jog in place on any surface, anywhere. The treadmill app keeps track of the number of steps, current speed, and average speed, the exercise time and current state. According to My Fitness Pal, a 50 kg person jogging in place for 30 min loses up to 272 calories. This opens doors to using VR for fitness freaks, medical patients and just about anyone else.

The E-commerce app features a shopping store and allows the user to explore the shop by letting them walk inside. They can pick up objects, view information about it and choose whether or not to buy it. This object interaction is triggered using the time gazed reticle along with head gesture recognition. With the emergence of WebVR, this would be a potential app for E-commerce giants like Amazon, Flipkart to explore as VR adds a new dimension in a way to interact with customers.

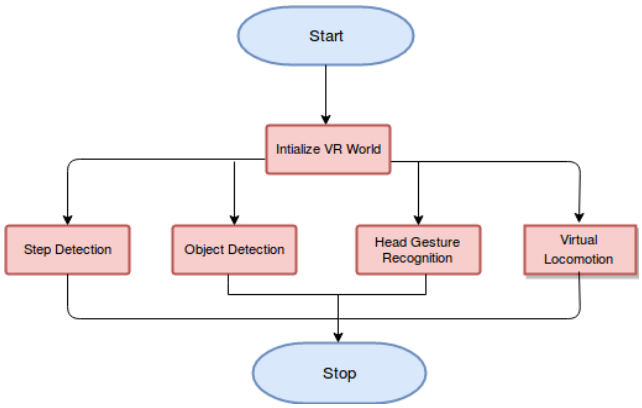

Figure 3. Final Flow of Events

The Figure 3 shows the flow of events that take place. Each of the components shown can work independently from one another but when used together the system reaches its goal of positional tracking and the core functionalities are met.

IV. IMPLEMENTATION & RESULT

For our smartphone mount, we used the Google Cardboard. The Cardboard features lenses with a 100 degree field of view. For user study we used an Android Nexus 5 smartphone with a Qualcomm Snapdragon 800 CPU (2.3 GHz quad-core) which features a InvenSense MPU-6516 six-axis (gyro + accelerometer) with a 50Hz sample rate. But any smartphone, running Android 4.4 or above could be used. Unity Daydream 5.4.2f2-GVR13 and Google VR SDK for Unity, v1.2 or above were used along with Blender, a 3D content creation platform, MATLAB a mathematical toolbox and Audacity, an audio processing tool for creating the framework. C# was effectively used to write various scripts for all components.

We asked 15 people, 8 male and 7 female, of varying age groups between 10-50 years to participate in a survey. The study compared three different navigation systems and asked users to rate the best system based on six different parameters. These parameters are: Efficiency, Reliability, Immersion, Freedom, Learnability and Preference. We selected these parameters based on the work of Sutcliffe et. al and Tregillus et. al.

The other two systems, Look Down To Move (LDTM) and Click To Move (CTM) are two of the most popular approaches to navigating in a virtual world. LDTM requires users to look down at the ground to start moving, and look down again to stop. Similarly, CTM requires the user to click on the screen or a button to start moving, and click the button again to stop moving. The biggest drawback with both these systems is that the user can move only at a constant speed, which reduces both freedom and immersion. The graph showing the comparison is shown in the Figure 4 as follows:

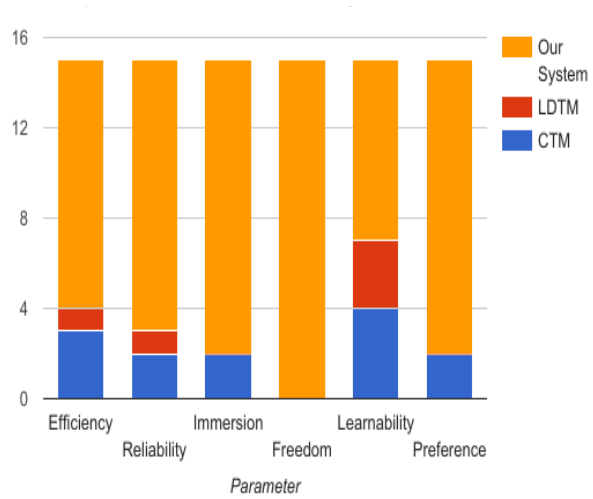

Figure 4. Comparison between Various Systems

As observed from above, our system outperforms the other systems in every parameter. According to this survey, users seem to experience maximum freedom while using our system, as expected. Similarly, our system seems to outperform the other systems in other parameters like efficiency, reliability and immersion. An observation was that users initially had some difficulty learning how to use our system. However after they were asked to try out a tutorial demo app, they felt much more comfortable. The overall consensus was that our system was the most preferred one, among the three. Being able to guarantee this level of freedom and immersion, while keeping in mind the cost factor as well as the constraint of not using any external hardware, is definitely both exciting and satisfactory.

V. CONCLUSION & SCOPE

Since the primary goal of this paper is to build a framework for VR systems, this paper has a broad scope. This framework can be used to build VR applications that can simulate a wide spectrum of use-cases, ranging from learning to fitness to sightseeing, or even gaming. The framework has been written in such a way that developers can build on top of it, to implement additional functionality to enhance the framework's capabilities, thereby allowing a wide-range of applications. Also, given that VR is currently trending and is an active area of research, there is a potential for usage of this framework for research purposes.

A lot of research is happening right now in VR, making it one of the hottest tech in the industry. VR has applications almost everywhere, from gaming and entertainment, all the way to sports and military. Much of today's VR systems use external sensors or controllers to track body movement or perform navigation in virtual worlds, and this has its limitations. Similarly, most of these support only orientation-tracking, and are yet to come up with exciting solutions for positional tracking. These systems are sophisticated and expensive.

The motive of the paper was to build a framework which can be used for positional tracking in mobile based virtual reality systems, using only hardware that exists on your phone and making sure that the setup is as cost effective as possible. We were able to successfully build a framework that in addition to allowing users to navigate in virtual world, also allows them to interact with objects present in these worlds through a variety of mechanisms, including head gestures. This framework thus can be tested in a variety of test environments. It will definitely be an exciting future with VR systems such as this one being created for everyone.

VI. LIMITATIONS & FUTURE ENHANCEMENTS

Since VR applications consume a lot of graphical computational power, smartphones with lower graphic capabilities suffer from lag and screen stutter. As a result, VR worlds built to experiment with the framework cannot be overly sophisticated in terms of graphics. In other words, it is preferable to use objects with a low poly count to create VR worlds. But this makes the application less realistic, visually.

Another limitation of the positional tracking system is that it can only detect steps, and no other types of actions like jumping, crouching, and punching, primarily because these require use of additional sensors external to the phone.

A major enhancement could be the addition of an appropriate hand tracking mechanism that will also capture hand gestures and simulate appropriate actions in the virtual world. We were unable to include such a setup because existing hand tracking devices require a PC to function. Given that our setup is hands-free, it is an ideal feature to incorporate. Upcoming device Leap Motion VR could be a potential solution. Another interesting challenge would be to integrate our framework with a proprietary application like Google Earth and Google Street View, which would potentially allow users to walk through streets and monuments anywhere in the world virtually, right from their own room.

REFERENCES

[1] Tregillus, S., & Folmer, E. (2016, May). Vr-step: Walking-in-place using inertial sensing for hands free navigation in mobile vr environments. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 1250-1255). ACM.

[2] Brajdic, A., & Harle, R. (2013, September). Walk detection and step counting on unconstrained smartphones. In Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing (pp. 225-234). ACM.

[3] Sutcliffe, A., & Gault, B. (2004). Heuristic evaluation of virtual reality applications. Interacting with computers, 16(4), 831-849.

[4] Virtual Reality, Wikipedia, Last Accessed on May 17, 2017 from https://en.wikipedia.org/wiki/Virtual_reality

[5] Unity Game Engine, Wikipedia, Last Accessed on May 17, 2017 from https://en.wikipedia.org/wiki/Unity_(game_engine)

[6] Blender Software, Wikipedia, Last Accessed on May 17, 2017 from https://en.wikipedia.org/wiki/Blender_(software)

[7] Positional Tracking, Wikipedia, Last Accessed on May 17, 2017 from https://en.wikipedia.org/wiki/Positional_tracking

Calorie Counter, My Fitness Pal, Last Accessed on May 17, 2017 from https://www.myfitnesspal.com/

978-1-5386-2044-1/17/$31.00 ©2017 IEEE