Positional Tracking for Mobile based Virtual Reality Systems

Sampath Shanmugam 1PE13CS131
Vaishak Vellore 1PE13CS174
Bharat Gogineni 1PE10CS021
Siddhant Priyadarshi 1PE12CS156

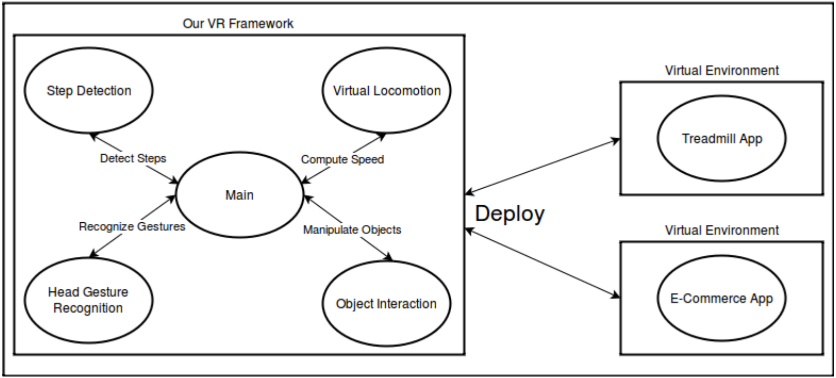
PESIT Bangalore South Campus
Guide: Dr. Sarasvathi.V
Associate Professor
Team 52

13th April 2017

Problem Description

- Objective: To develop a positional tracking system for smartphone based Virtual Reality(VR) Head Mounted Displays(HMDs), that tracks body movements for user interaction and navigation within the VR environment, with the user walking in place in the real world, to create a naturally immersive VR experience at a low-cost.

System Architecture



Step Detection

- We had discussed in the previous demo about our Step Detection module.
- In short, we capture accelerometer reading values along the X, Y, Z axes and compute their combined magnitude.
- Next, we subtract the average mean of acceleration values from the current input value, to remove any constant effects, like gravity.
- We then count the number of peaks. Each peak represents a step.

Virtual Locomotion

Step 1: Start.

Step 2: Get $\mathbf{t}_{\text{step}}$, which is the time taken for the current step. Add to $\mathbf{t}_{\text{total}}$.

Step 3: Compute the instantaneous speed, **v**, as:

$$v = \frac{1}{t_{step}}$$

Step 4: Compute average velocity, $\mathbf{v}_{\text{avg}}$, as:

$$v_{avg} = \frac{n_{step}}{t_{total}}$$

Step 5: Stop.

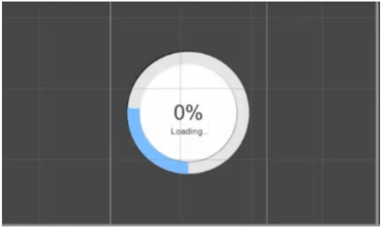
Head Gesture Recognition

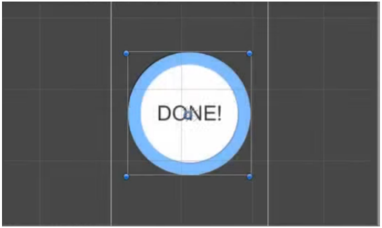
- Since Google VR SDK supports orientation tracking natively, we decided to experiment with head gestures.
- Our system detects rotation along the X and Y axes to be most effective.
- This allows us to work with two independent head rotation actions.
- Action 1: Move up and down/Rotate about X axis (Yes).
  Action 2: Move left and right/Rotate about Y axis (No).

Object Interaction

- Since the reticle can be used to look at objects, we can use it along with the time gazed progress bar.
- Upon looking at an object, we trigger a timed gaze progress bar, that moves it from 0 to 100 percent.
- It is easy to create responsive and stylish progress bars using the Unity UI.
- This setup allows users to interact effectively with the virtual world objects, discarding need for any touch input.

Screenshot





Screenshots of Radial Progress Bar with Timed Gaze on Unity.

Virtual Environment

- VR can be applied to many spaces like fitness, E-Commerce, learning, in addition to entertainment.
- Since we have built a VR framework in this project, we decided to deploy it in two virtual environments: Fitness and E-Commerce.
- Fitness: We built a Treadmill VR application that keeps track of your fitness.
- E-Commerce: We built a VR application that attempts to emulate the future of online shopping.

Treadmill App

- This goal of this 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, 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.

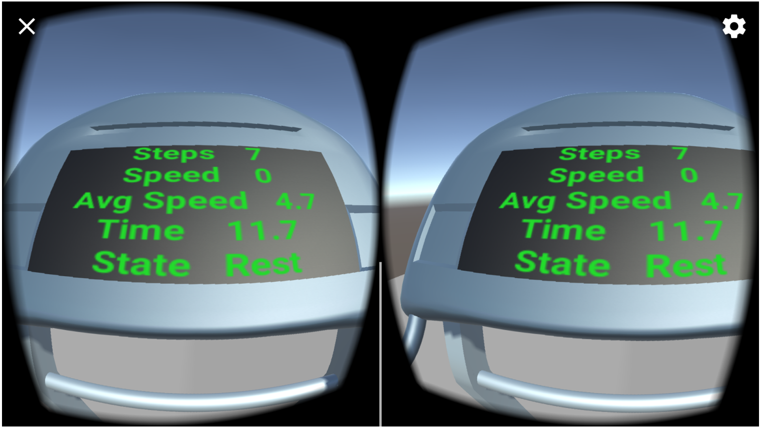
E-Commerce App

- The simulation features a shopping store and allows the user to explore the shop by letting him/her 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 Unity UI features.
- This would be a big step for E-commerce giants like Amazon, Flipkart as VR adds a new dimension in a way to interact with customers.

Implementation

- Unity 5.4.2, a cross-platform simulation creation engine.
- Google Cardboard SDK for Unity, to implement VR features.
- MATLAB, for analyzing the accelerometer input signal
- Smartphone, running Android 4.4 or above.
- Blender, a 3D content-creation program for building VR worlds.
- C#, the language of implementation, supported by Unity.

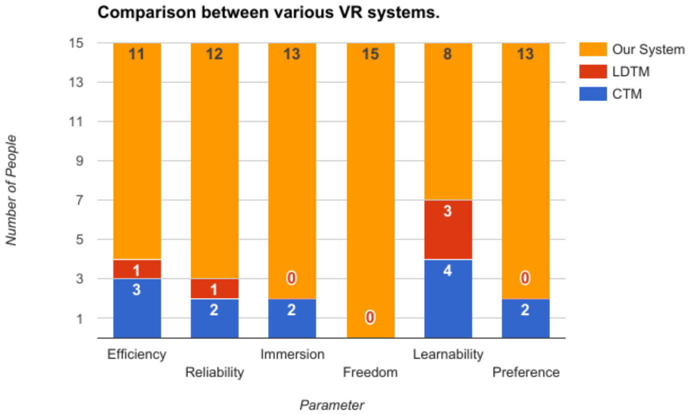
Results


Steps 7
Speed 0
Avg Speed 4.7
Time 11.7
State Rest
Steps 7
Speed 0
Avg Speed 4.7
Time 11.7
State Rest

Results

- 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.

Results


Comparison between various VR systems.
Our System
LDTM
CTM
Number of People
15
13
11
9
7
5
3
1
11
12
13
15
8
13
1
3
1
2
0
2
0
3
4
0
2
Efficiency
Reliability
Immersion
Freedom
Learnability
Preference
Parameter

Future Scope

- Our goal was to build a positional tracking system, that was both immersive and low-cost for a mobile VR system like Google Cardboard.
- Through this project we have achieved the same. As reflected in the survey, our system was the most preferred, compared to other existing systems.
- An appropriate system for hand tracking can be integrated with our framework, for a more immersive experience.
- Our framework can be deployed in a variety of virtual environments, with different use-cases.

Bilbliography

[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.