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. Saraswathi.V
Team 52

09th March 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.

Modules

- Step Detection: Uses the phone's accelerometer to detect steps.
- Virtual Locomotion: Computes velocity of walking in virtual world is based on WIP speed.
- Object Interaction: Objects are manipulated by looking at them and triggering an action.
- Virtual World: The environment within which the application is deployed.

Algorithm

Step Detection:

Step 1: Start.

Step 2: Capture the magnitude of current acceleration, $\mathbf{acc}_{\mathbf{cur}}$, computed as follows:

$$r_{magnitude} = \sqrt{x_{accel}^2 + y_{accel}^2 + z_{accel}^2}$$

Step 3: Update the value of average acceleration, $\mathbf{acc}_{\mathbf{avg}}$.

Step 4: Calculate **delta**, the difference between $\mathbf{acc}_{\mathbf{cur}}$ and $\mathbf{acc}_{\mathbf{avg}}$.

Step 5: If **delta** > $\mathbf{delta}_{\mathbf{threshold}}$, then increment value of **step**.

Step 6: Wait for a time $\mathbf{t}_{\mathbf{wait}}$, before repeating the above steps.

Step 7: Stop.

Algorithm

Virtual Locomotion:

Step 1: Start.

Step 2: Get $\mathbf{t_{step}}$, which is the time taken for the current step.

Step 3: If $\mathbf{t_{step}} > \mathbf{I_{max}}$, set **v** to $\mathbf{V_{min}}$.

Step 4: Else if $\mathbf{t_{step}} < \mathbf{I_{min}}$, set **v** to $\mathbf{V_{max}}$.

Step 5: Else if $\mathbf{I_{min}} \leq \mathbf{t_{step}} \leq \mathbf{I_{max}}$, we can linearly interpolate the value of **v** as follows:

$$v = \frac{t_{step} - I_{min}}{I_{max} - I_{min}} * (V_{max} - V_{min}) + V_{min}$$

Step 6: Stop.

Algorithm

- Based on Step Detection and Virtual Locomotion, WIP in the real world is translated to movement in the virtual world.
- Object Interaction is achieved by using a reticle that points in the direction of the user's gaze.
- Google Cardboard SDK has an in-built event management system, that detects taps/clicks.
- The virtual world will consist of paths for users to navigate, and objects for users to interact with.

Implementation

- Unity 5.4.2, a cross-platform simulation creation engine.
- Google Cardboard SDK for Unity, to implement VR features.
- 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.

PES Institute of Technology

Expected Output

- Low-latency, low-noise translatory motion tracking, via the Step Detection module.
- Varying velocities of user in virtual world, based on their WIP velocity in the real-world, via the Virtual Locomotion module.
- Effective interactions with objects in the virtual world, via the Object Interaction module.
- An interesting and practical application of our system, via the Virtual World module.

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.

PES Institute of Technology