COMP3617 – Virtual and Augmented Reality Coursework Report

frcg69

The headset in various tracking positions

A white object with a black handle

Description automatically generatedA white object with black text

Description automatically generatedA close-up of a frame

Description automatically generatedA white and black object

Description automatically generated

The headset from various eye locations

A close-up of a virtual reality headset

Description automatically generatedA computer mouses falling

Description automatically generated

# Tracking Correction

The aim of tracking correction is to account for drift from the true orientation of the headset due to the shortcomings of gyroscopic causing true up and true north to be shifted within the world away from the actual direction of said reference points. Accounting for drift from the up vector is done through the accelerometer, allowing acceleration due to gravity to show true up. Tilt correction uses this to reorient the headset so that the world and true up vectors match. By tuning the alpha value one can control how strong this reorientation is. At low values such as 0.01 the reorientation is subtle and smooth. At higher values such as 0.1 the reorientation can be very shaky and sudden especially when rotating the headset leading to disorientation if applied to an actual headset. At even higher values such as 0.5 the rotation of the headset becomes completely unaligned from the true values causing the headset to rotate wildly before returning to appropriate positioning when gyroscope readings return to stationary. I believe that this could be due to the headset moving in physical space causing the accelerometers to misunderstand acceleration in a non-vertical direction to be confused for the acceleration due to gravity. Using low values allows for some tilt correction to be performed while still allowing smooth gyroscopic rotations. Yaw drift is compensated for by using magnetometers which measure the direction of magnetic north the correction is formed quite similarly but ignoring the up direction. The alpha value being changed in this case is less impactful than with tilt correction. Much higher levels of yaw correction can be used without causing unwanted behavior in world rotation. Especially high values such as 0.5 still can cause jittering and should not be employed but with more moderate alpha values around 0.1 jittering is invisible, and the correction is smooth. Magnetometer correction is more usable than accelerometer correction due to its resilience to movement within space as the direction of magnetic north changes far less than direction of acceleration.

# Dead Reckoning vs Accelerometer Correction

A dead reckoning filter simply uses the gyroscopic movements to approximate the rotation of the headset in roll pitch and yaw directions. A dead reckoning filter works by compounding these gyroscopic readings over time. This can create quite simple and smooth rotations especially with high sampling rate. The simplicity also allows for faster calculations allowing for a lower latency experience. This also has fewer points of failure than using both the gyroscope and accelerometer. In that case, a faulty accelerometer reading can drastically impact the experience causing extreme rotation when none has occurred. If a gyroscopic sensor errs then the impact is far less, causing some drift but with overall, a much less noticeable change. A dead reckoning filter is susceptible to drift. This can be caused by a variety of factors, from slightly poorly calibrated sensors to rounding the rounding of figures, but is compounded through an extended period, causing disorientation. Drift can be mitigated by using accelerometer correction which uses acceleration due to gravity to fix this issue. This allows for increased accuracy especially over longer periods of use, preventing disorientation. Using accelerometers can be quite complex causing slower calculations resulting in higher latency. Implementing them is also more complex requiring the ability to account for lateral acceleration to prevent incorrect assessment of the direction of gravity.

For a dead reckoning filter, calculating orientation requires few calculations. First the values are converted into quaternions, then a single quaternion product is calculated to get the updated orientation quaternion. Incorporating accelerometer readings first requires all dead reckoning calculations. The accelerometer readings must then be converted to world space, this requires two quaternion products to be calculated. Next a tilt axis and an angle φ must be calculated from the resulting quaternion. These must me converted again to a quaternion using angle axis representation. This is finally combined with the dead reckoning filter orientation quaternion with another quaternion product. A dead reckoning filter requires one quaternion conversion and one quaternion product. Using acceleration requires two quaternion conversions three quaternion products and the calculation of tilt axis and φ.

# Level of Detail

The renderer automatically changes between three models depending on Euclidian distance from the eye. It uses a complete version of the headset object within fifteen units of the eye, beyond this it switches to one with half the polygons until 25 units from the camera where it then switches to a model with a quarter of the polygons. Further level of detail reduction could be done through extrapolating current known information into the future, reducing the number of physics updates required, this is especially beneficial for collision. Occlusion culling could be used to cull objects hidden behind other objects. This could be done by casting rays from the eye and culling a polygon if all vertices’ rays intersect with another polygon. For further objects fewer rays need to be cast as the objects appear much smaller and if one point has an intersecting ray, it is far more likely that others do. This would remove the need for all polygons within the frustrum facing the eye to be rendered and placed into the Z-buffer. A render distance could be applied to prevent far away objects from being rendered, this could be enhanced by adding a fog effect so that objects slowly fade away rather than disappearing sharply.

With the simple distance based LOD model swapping I have tested performance by spawning a number of headsets from 10 to 50 in the +Z direction with X and Y values appropriately scaled by Z. which are granted velocity towards 0,0,0 with some accounting for gravity. Headsets that fall below the view frustrum are then moved to above the view frustrum to allowing the same headsets to be re-rendered. Headsets out of view are culled to prevent unnecessary rendering. The FPS was measured as the average across the first 500 frames of a sequence. The desired number of headsets is created in addition to the static rotating headset with seeded randomness to ensure equal environments.

|  |  |  |
| --- | --- | --- |
| # of Headsets | FPS with LOD | FPS without LOD |
| 10 | 1.01 | 0.59 |
| 20 | 0.87 | 0.33 |
| 50 | 0.33 | 0.11 |

# Positional Tracking

A positional tracking methodology could be implemented with the use of the accelerometer in combination with the aid of gyroscopes and magnetometers. To use accelerometers to determine the location, initial calibrations must be made, positioning the headset in world space accurately. This could be done by placing the headset on the floor as a reference point or having the user input their height and stand up. From there positional tracking could be implemented in a similar way that gyroscopic tracking is, using acceleration to approximate movement by applying updates to the previous location within world space. This would require gravitational acceleration to be counteracted to prevent continuous downward movement. This could be done through trusting the gyroscopic approximation of where up is and applying acceleration equal to gravity in that direction. This would struggle to maintain accuracy over time due to drift from gyroscopic acceleration. This could be mitigated using the magnetometer. The direction of magnetic north could serve as an approximation of the true orientation of the headset. This would also be inaccurate if not properly parameterized and calibrated against local magnetic sources. This would require coordinate location as well as accurate altitude information to account for differences in the angle between flat ground and the magnetic field. The magnetometer up calculations would not be very trustworthy though due to the sensitivity of magnetometers and the changing of magnetic fields over time.