BBC

VR | BARRIERS RESEARCH

VR PERFORMANCE ANALYSIS & TOOLS

December 2019

Overview

The document provides a summery of our work so far exploring performance from a VR context and presents some future options we are looking to explore.

This Document:

1. What the targets for performance are and why we have chosen them.
2. Methods for validating performance we have explored and the results.
3. Future methods we are exploring and the positives and negatives of each approach.

About Us

The purpose of the BBC UX&D Accessibility team is to understand barriers present within BBC apps and services and to support teams to include the whole audience.

1. Targets

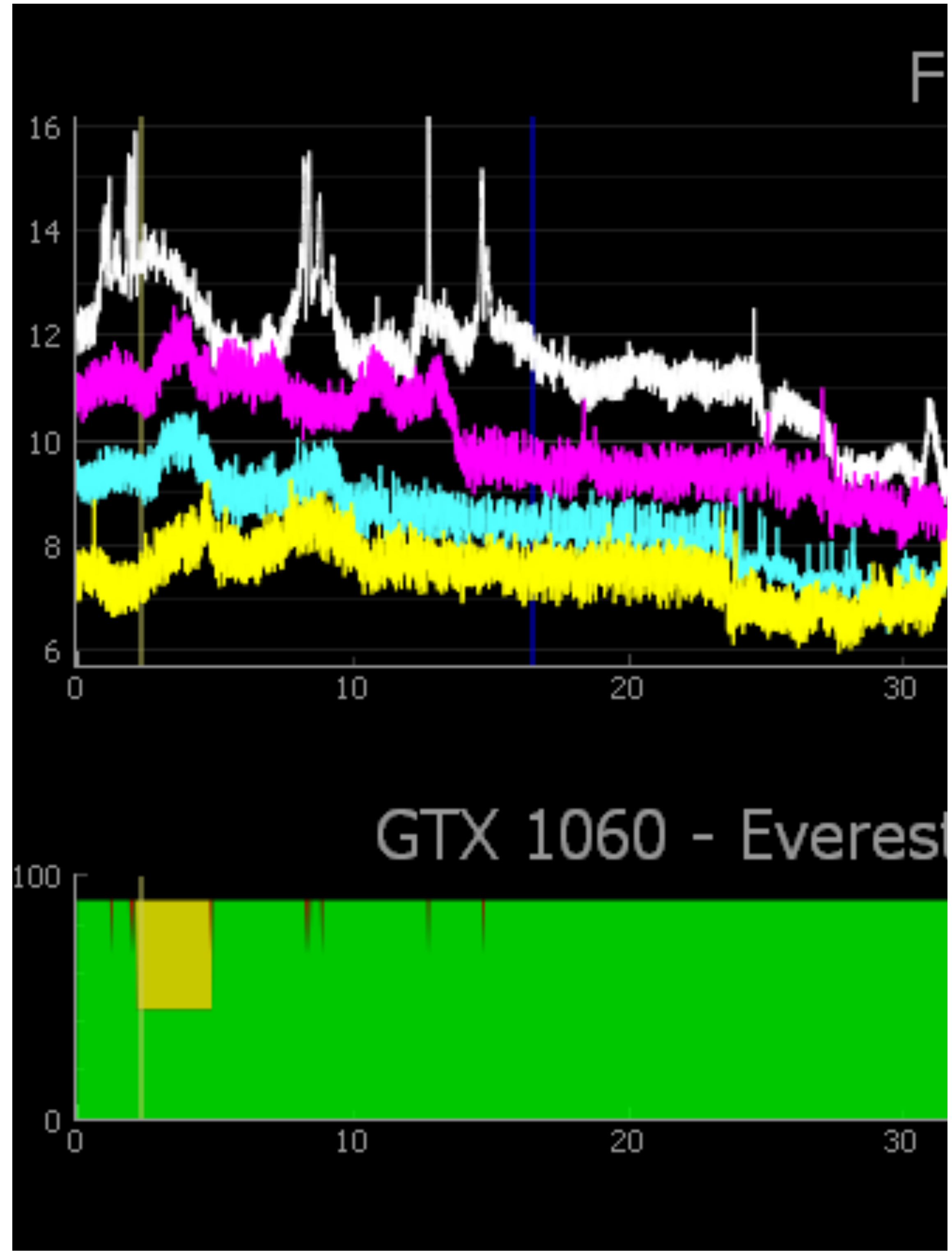
Background.

Virtual Reality (VR) systems use a Head Mounted Display (HMD) to allow users to view a 3D rendered scene. By repositioning the content in sync with the users movement, it creates the illusion of being in a different place.

The consensus from HMD manufactures[1], academic studies[2] and our experience so far all indicate a frame rate of > 90FPS is required for an experience to be safe and comfortable.

To be confident that our test experience is safe we have been seeking to verify our test content performs at above 90FPS or frames are generated in less than 11ms.

1. https://developer.oculus.com/documentation/pcsdk/latest/concepts/dg-performance-guidelines/
2. *Temporal Resolution Multiplexing: Exploiting the limitations of spatio-temporal*



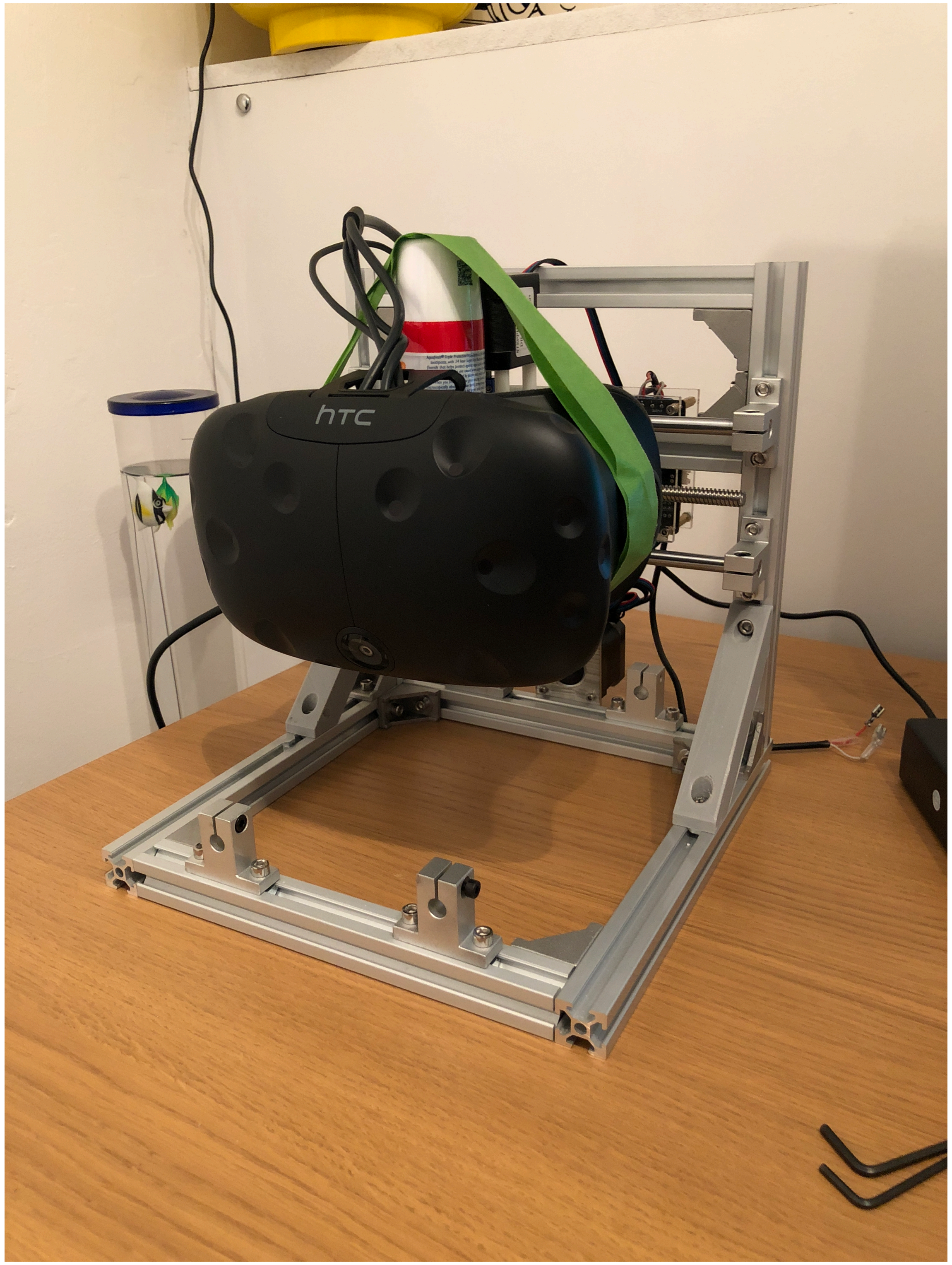
2. Methods we have explored

Introduction

So far in the VR barriers project we have explored performance across two dimensions.

First we have conducted long running system stress tests to find a stable performance threshold for the our VR testing system. These tests are not VR specific and test the underlying system.

Secondly we have looked at long running CNC controller tests moving the headset to replicate a more complete picture of VR environment performance.



METHODS WE HAVE EXPLORED

System Stress Testing.

During VR Barriers user testing sessions the VR system may be running for 6 or more hours. For our VR performance testing to be reliable we had to understand the performance characteristics of the underlying hardware.

To do this we ran synthetic stress test tools (Prime95 and Furmark) to provide a load to the CPU and GPU. This caused the hardware to overheat and slow down within 2 hours.

We then used the Intel® XTU to apply a small under-volt to the CPU. This decreased the thermal load and balanced the system.

We then repeated this process whenever a significant software or hardware change was made.

CNC Physical Test System

Once we have confidence in the underlying system, we moved on to testing the system end to end.

To provide a reliable and consistent input, we built a CNC testing rig to move the headset. We placed the headset on the robot and executed repeated test patterns. The movement was small (80mm in Z, 100mm in Y axis) but enough to prove that the system was meeting the 90 frames per second target during movement over a long period.

During testing we monitored frame rate using the NVIDIA FCAT and SteamVR toolsets.

This process was useful in developing a VR environment which was performance and reliable. Outside of the VR headset, we saw frame rates exceeding 240 frames per second.

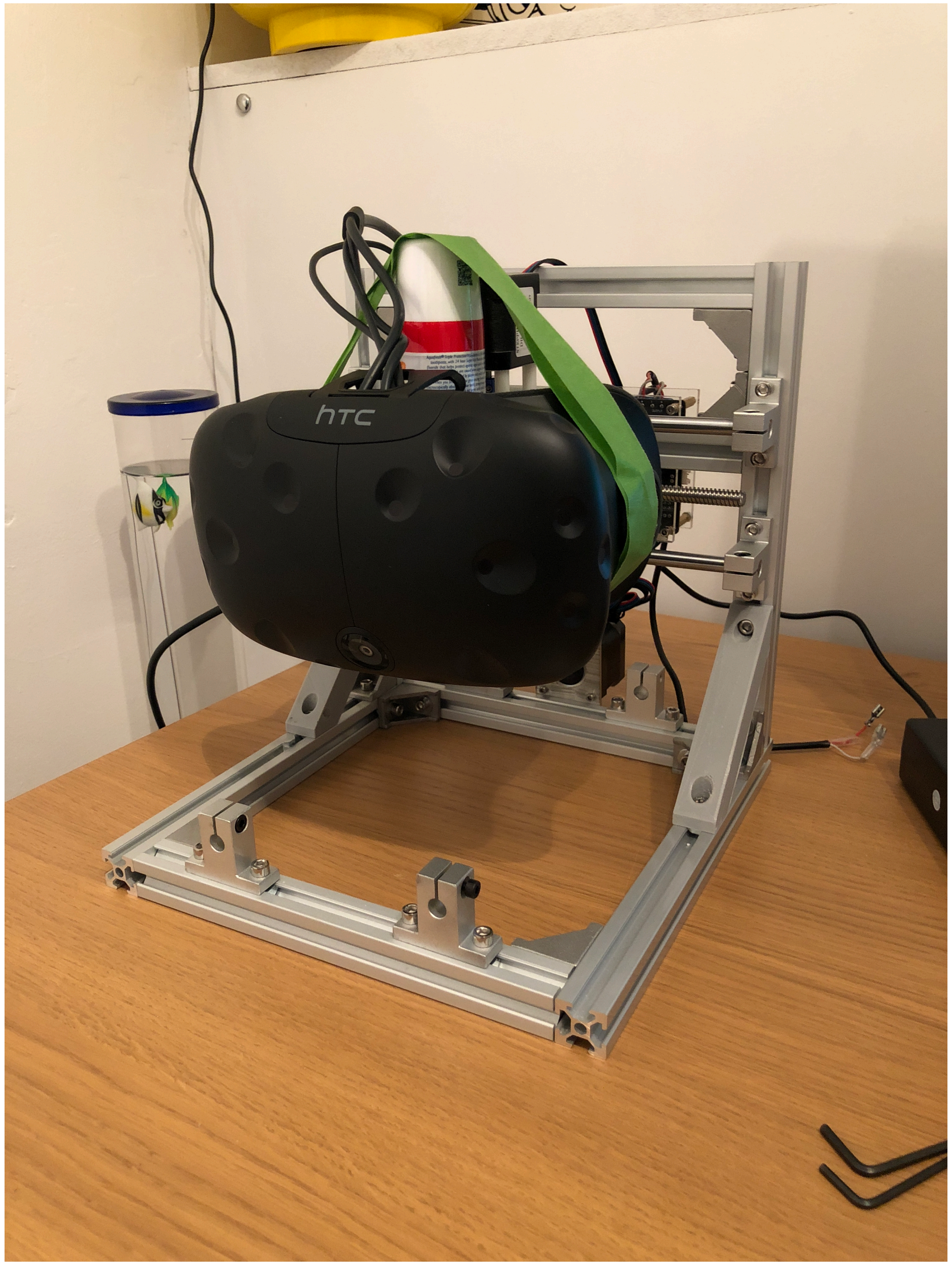
3. Future Methods

Introduction

Going forward we need to expand our performance testing and verification work to build confidence in a great variety of situations.

The CNC test approach we used was limited to 2 directions of movement, but could not test yaw, pitch or roll.

To test in more ways, we are exploring a selection of software and hardware approaches.



SOFTWARE METHODS

1. Unreal 4 Stereo Mode.

Within Unreal 4 there is a flag which enables headset emulation. This will switch the rendering pipeline to stereo mode and provide viewport rendering. This mode combined with scripted camera movement could emulate a user in a HMD.

Pros:

- Simple
- No hardware needed
- Agnostic of headset
- Robust to underlying platform changes

Cons:

- Doesn't test the full pipeline, only the game engine.
- Time consuming to programme an accurate user movement within the editor.

2. Custom OpenVR Device Driver

This method involves creating a virtual tracking device within the OpenVR driver. The virtual device can then relay positional information for a fake headset into the driver from a file.

Pros:

- End to end test
- No Hardware Needed
- Flexible to future needs
- Robust to underlying platform changes

Cons:

- Technically complex
- Brittle as drivers change / evolve
- Limited support for headsets and other technologies

HARDWARE METHODS

3. Bungee cord rig

This method involved building a small rig to physically move the headset and measure performance. Using a structure and some bungee to create high speed randomised movement much quicker than a human could produce.

The same energy is displaced with each run and a statistical mean is approach

Pros:

- Simple
- Agnostic of all hardware and software
- Robust to underlying platform changes

Cons:

- Hardware needed
- Not Portable or convenient
- May not map to real world usage well enough

4. Motion Controller Camera Dolly

This method involves using a motion controller camera dolly and fitting the headset to the dolly.

The dolly can then replicate movements within a space of a few meters.

Pros:

- Simple
- Agnostic of all hardware and software
- Robust to underlying platform changes
- Off the shelf hardware / rentals

Cons:

- Hardware needed
- Not Portable or convenient
- Expensive & requires external expertise