

Wireless Virtual Reality for Entry-Level Smartphones

December 23, 2015 · Technology

Peter He

He P. Wireless Virtual Reality for Entry-Level Smartphones. Young Scientists Journal. 2015 Dec 23 [last modified: 2015 Dec 23]. Edition 1.

Abstract

In the recent decade, Virtual Reality (VR) has become more common through a range of systems including computer-powered head-mounted displays (HMDs) and smartphone-powered headsets. It has presented both designers and developers alike with a new field filled with unexplored potential. In this article, a new and inexpensive approach to VR is explored, allowing entry-level smartphones to run powerful computer-based VR experiences wirelessly through Bluetooth with relative high levels of performance. The project was recognised as a European and African Finalist in the Google Science Fair 2015.

Introduction

With the rise of virtual reality, we have witnessed nothing less than the birth of a new form of media. Virtual reality (VR) presents users with a virtual environment and mimics sensory stimuli (notably sight and sound). Its applications include military and medical simulations as well as gaming though this is not, by any means, an exhaustive list – a quick search for “Virtual Reality” returns uses in neuroscience, stroke recovery and the treatment of mental illnesses such as Post-Traumatic Stress Disorder (PTSD).¹

Over the past few years, VR has been made, to an extent, available to the masses through computer-powered peripherals such as helmets and controllers, as well as smartphone-powered headsets.² Current smartphone-based virtual reality, however, seems only to exist for a few high-end devices making use of on-board sensors, a cabled connection to a computer (which can limit movement) or expensive headsets housing embedded processors. With the shortcomings of current technology in mind, this project aims to develop an inexpensive build-it-yourself wireless VR headset prototype for entry-level smartphones targeted at “casual” gamers and schools.

For this project, a system was implemented using a computer and webcam to detect head movement and compute physics, artificial intelligence and other elements required to create a virtual world. The data from these calculations are sent to a smartphone over a Bluetooth connection and fed into its software. This software makes use of GPU accelerated motion tracking, depth detection and voice control, which allows users to interface with the program without ever leaving the experience. Finally, it is ported to the Windows Phone operating system – porting is the process of making software work in a different environment to the one for which it was designed (the Windows Phone operating system currently lacks any virtual reality applications). The final headset was cheap and easy to build, with materials costing less than £6.

Research and Literature Review

Current Technology

The concept of virtual reality is not new; the technology behind it has been developing for several decades. However, current domestic virtual reality technology tends to fall into one of three trends:

- Computer Peripherals
- Self-Contained Headsets
- Smartphone-Based Headsets

These all carry merits and shortcomings which can be summarised in the table below:

Image not found

https://www.youngscientistsjournal.com/wp-content/uploads/2015/10/Yays-and-Nays-of-VR1-300x126.png

Furthermore, many of these systems are currently targeted at developers and hard-core enthusiasts. It was therefore hypothesised that combining the portability and freedom of smartphone-driven VR with the performance of computer peripherals (using resources which students could easily get their hands on and some sort of wireless connection) could present a new, simpler, cost-effective VR headset for students and “casual” gamers left behind by current industry trends.

Audience and Constraints

Virtual reality has foiled designers and entrepreneurs for decades; it’s only recently that we seem to be leaving the CAVE (Cave Automatic Virtual Environment) era. Therefore, without much public research into developing such applications, research had to be carried out from the ground up.

The first problem that needed addressing was finding out what students have available to them. This was quite easily solved with a straw poll of 650 students across multiple schools and nations, returning a rather succinct set of results:

- The majority (85%) had access to basic craft equipment (card, scissors)
- The majority (90%) owned smartphones
- The majority (85%) had access to a webcam

All had access to a computer (of course, this was biased as computer access was needed in order to vote).

This meant that smartphones and computers could be utilised without losing much of the target audience. The crafts equipment figures were also quite promising, indicating that the majority of people wouldn’t have to spend much money at all (if the prototype design were to be successful).

Tracking

Virtual reality has to be responsive; one of the current industry leaders (a computer peripheral) can achieve latency of about 30-40ms in perfectly optimized conditions (though it could be preferably lower)³⁴. Presented with two potential methods of tracking: a smartphone’s on-board motion sensors and a webcam, it was concluded that using a webcam would prove more effective for the following reasons:

- Not all smartphones have on-board sensors (my Nokia Lumia 520 was such the case)
- Even if they do, their responsiveness varies depending on the phone (and could cause latency to vary from phone to phone)
- Sending data from a phone to a computer only to have data sent back to the phone would produce unneeded latency

An existing implementation of object detection by colour for the CPU worked very well; however, it needed optimising and tweaks for its purpose. These were made with relative ease (and included replacing the colour detection algorithm, modifying the tracking algorithm and moving some processing to the GPU). This gave performance a huge boost, although it would only work on computers equipped with GPUs (fortunately, there are growing numbers of computers which do so).

Wireless Communication

Used in keyboards, mice and other wireless devices, Bluetooth has always been at the forefront of short-range device-to-device communication and is now included with most smartphones. It’s easier to set up and requires less energy than alternatives, but comes with a low data transfer-rate and therefore may present problems with latency. This would prove a challenge as the data sent over Bluetooth has to be as compressed as possible; however, if successful, could allow higher volumes of information to be transferred when ported to WiFi or another alternative.

Camera Model

Stereoscopy is the technique of placing two images (or cameras) side-by-side to produce an illusion of depth. There are 3 ways to do this:



<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/StereoCameraModels-300x147.png>

Of all these, off-axis is most effective for it provides each eye with an individual vanishing-point (not necessarily at the centre of the screen). However, any camera projection matrices in the virtual environment would need tweaking on a per-eye basis.⁵⁶

Fig. 1: 3D Object from an Off-Axis Camera Setup

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Eye-Model-0-300x137.png>

Fig. 1: 3D Object from an Off-Axis Camera Setup

Fig. 2: Near and Far Clipping Planes. From this we can derive a camera projection matrix.

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Eye-Model-300x203.png>

Fig. 2: Near and Far Clipping Planes. From this we can derive a camera projection matrix.

Method

Software

In terms of software, the project is divided into two sections: the core systems (tracking and networking) and the surface systems (APIs and wrappers). Here's a slightly truncated look at the main ones:

WebcamInput was achieved through DirectShow ⁷.

Tracking: The tracking algorithm scans in lines of pixels until it comes across a candidate pixel of the correct colour. When this happens, it attempts to flood-fill from the candidate pixel (so that any neighbouring candidates may be picked up) using an optimized 4-way fill algorithm. From this, markers are generated and an average centroid computed on the GPU. Changes in position are calculated and passed into the simulation.

Calculation of a Centroid with Bounding Boxes

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Bounding-Boxes-300x196.png>

Fig. 3: Calculation of a Centroid with Bounding Boxes

Depth Detection: With the area of a marker being processed on a per-pixel basis, it was quite easy to create approximate depth detection by comparing the current marker area to the initial marker area. A comparison of the distances between markers is also made. An interesting application of this was in hand-tracking in the streetfight/graffiti demos. This did, however, require the entirety of the marker to be visible at all times. This suggested that point-based calculations using smaller markers such as LEDs could be introduced (these were introduced in the third prototype and provided tracking resistant against changes in lighting).

Easy Integration into Unity

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Easy-GE-300x147.png>

Fig. 4: Easy Integration into Unity

The API was made for Unity, a cross-platform proprietary game engine which offers a quick, easy-to-use interface for game design. It includes an input class, modified camera system, basic pre-made controller "prefab" as well as other useful utilities. Using this, a series of demo applications were built (these can be seen in the 'Results' section).

Bluetooth Functionality was achieved through hosting a Bluetooth server-client interface on the computer with the smartphone.

Spatial Audio was implemented through head-related transform functions (HRTF) ⁸⁹, but was rendered obsolete by Unity 5 (game engine) which has built-in spatial audio.

The Smartphone App acts as a hub for all the VR apps. After a quick setup (fig. 6), users are brought to a menu (navigable by looking around) from which they can launch apps from within the experience via speech-recognition.

Fig. 5: System Overview. The basic transmission is in the following format: [Transformation Data] [Rendering and Lighting Helpers] [A

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/VR-System-300x218.png>

Fig. 5: System Overview. The basic transmission is in the following format: [Transformation Data] [Rendering and Lighting Helpers] [Audio Cues] [Extra Data]

Fig 7: Setting up a Bluetooth connection

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Bbsetup1-225x300.png>

Fig 6: Setting up a Bluetooth connection

Fig. 7: Setting up on the Computer-side

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/ExBawxHand-300x189.jpg>

Fig. 7: Setting up on the Computer-side

Hardware

Despite having set out with some pretty clear hardware design targets, a number of prototypes still had to be built. The headset design had to be:

- Wireless
- Capable of containing a phone
- Limited to using *basic* craft resources
- Easy to make
- Comfortable

Prototypes

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/ExBawxProto00as-300x104.png>

Having built a vertical slice of a software demo, hardware prototyping began. The first was built from a headband with magnifiers glued onto it. The phone was held in front of the user's face by a cardboard frame. This, however, was uncomfortable to wear, and let external light reflect off the screen and allowed the phone to move about during sharper turns.

Having learnt from the first prototype, another was assembled. This time the phone was confined in a dark box (this increased picture quality) and held to the face with elastic. However, it was difficult to insert the smartphone. A third prototype soon followed, featuring a more comfortable shape and a modular design, which allowed focus to be adjusted (therefore catering to people suffering from near/far-sightedness). LEDs were also incorporated into the markers to ensure consistent tracking regardless of light levels.

Results

The final prototype was first tested in terms of software: a few timers were put into the code and measured that the average latency from webcam input to signal output in optimal conditions was about **10ms** (Fig. 8).

The same was done with the smartphone app with a Windows Phone (512mb RAM); the average signal-received to data-processed time was **<6ms**. What was more difficult to measure was the speed of transmission; an attempt to do this was made using synchronised clocks. However, the results were erratic and therefore negligible. Internet searches suggested that latency from a conventional Bluetooth transmission could range from **3ms to 100ms** (which isn't much help) though Bluetooth 'Smart' is capable of consistently going as low as **3ms** ¹⁰. The headset was incredibly responsive – having wandered around a virtual environment (with a virtual gun built from a modified bluetooth mouse) for 10 minutes without feeling at all nauseated (which is

common to many VR headsets).

Fig. 4: Computer-side latency over 67 frames (approx. 3 seconds) of webcam-input. The sudden peak at the beginning was anomalous

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Latency-300x229.png>

Fig. 8: Computer-side latency over 67 frames (approx. 3 seconds) of webcam-input (higher webcam framer-rate would be ideal). The sudden peak at the beginning was anomalous and therefore excluded from average calculations.

Fig. 5: Headset Modules

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Modules-300x224.jpg>

Fig. 9: Headset Modules

Full thing

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Full-thing-300x204.png>

Fig. 10: Prototype 3

Costs

The resources required to make the final prototype can be seen in the table below, though there were some left-overs.

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Table-of-Costs-1-300x132.png>



Testing with others and going public

The headset was tested with a group of students, each provided with a vague how-to booklet. Fortunately, results were **overwhelmingly** functional; feedback was mostly positive. This demonstrated excellent user-friendliness. More surprisingly, in terms of aesthetics, no two headsets looked alike – some were multicoloured masses of card, others were minimal.

Fig. 7: Excerpt from "So You Want to Build an ExBawx*" (instructions guide) *ExBawx parodies Microsoft's X-Box

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/So-you-wanna-build-an-exbawx-262x300.png>

Fig. 11: Excerpt from "So You Want to Build an ExBawx*" (instructions guide) *ExBawx parodies Microsoft's X-Box

The project was then released onto various VR developer and enthusiast forums to see what a small section of the internet community thought of the project; feedback here was also hugely positive. Criticisms mainly revolved around aesthetics.

Demo Gallery

Fig. 12: Computer vs. Mobile. Normally, the computer does not render the image to screen (this takes time) - it has been for the sake

Image not found
<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/VR-Sync-Demo-300x286.png>

Fig. 12: Computer vs. Mobile. Normally, the computer does not render the image to screen (this takes time) – it has been for the sake of demonstration.

Fig. 13: Screenshots from various Demos. Left: from John and the Arbitrary Gem Quest by Steven Salmond (ported to VR with the API

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Demo-Screenshots-300x132.png>

Fig. 13: Screenshots from various Demos. Left: from John and the Arbitrary Gem Quest by Steven Salmond (ported to VR with the API) ¹¹ ; Right: First Person Shooter Demo

Fig. 12: Drawing a Smiley Face in a Graffiti Game with Hand Tracking

Image not found

<https://archive.archive.ysjournal.com/wp-content/uploads/2015/10/Drawing-Smiley-300x132.png>

Fig. 14: Drawing a Smiley Face in a Graffiti Game with Hand Tracking

Conclusion

This project has successfully overcome the limited processing power of smartphones to implement a wireless virtual reality system at low cost. When tested, the system as a whole ran with about **25ms** latency (adding a few milliseconds for the signal to travel over Bluetooth Smart), mitigating latency-induced simulator sickness (this was done on a Nokia Lumia 520 – a “low end” **12** smartphone which houses a mere 512mb of RAM). An implementation of hand and “auxiliary” object tracking has enabled users to interact with virtual environments in real-time and an easy-to-use multi-platform API was developed to allow quick integration into major game engines.

This project has validated the hypothesis and demonstrated that it is possible to design an inexpensive and wireless VR headset using only basic craft materials; it has also opened up new development opportunities, among which are:

- **Remote control of drones and robots** in dangerous situations through wireless VR (enabling users to interact with the real world in dangerous situations in relative safety).
- **Realistic simulations** – being able to move get up and walk around (which is possible through the depth detection system).
- **Education** – either building or using these headsets in the classroom (a computer can support multiple headsets simultaneously in one virtual environment).

Development opportunities more specific to the project include:

- **Further and faster** compression of data before transmission.
- **Moving even more processing to the GPU** (though this may be risky as many computers still aren't equipped with such technology).
- **Cross-platform deployment**
- **Using more powerful lenses** (though this will cost more as well as require correction for the distortion produced by the lenses with a fragment shader).
- **Recreating more senses** (opening up many more areas for research).

Bibliography

Pharr, Matt, and Randima Fernando. GPU Gems 2: Programming Techniques for High-performance Graphics and General-purpose Computation. Upper Saddle River, NJ: Addison-Wesley, 2005.

Nguyen, Hubert. GPU Gems 3. Upper Saddle River, NJ: Addison-Wesley, 2008.

Orland, Kyle. "How Fast Does "Virtual Reality" Have to Be to Look like "Actual Reality"?" (2013)

Polyakov, Alex, and Vitaly Brusentsev. Graphics Programming with GDI & DirectX. Wayne, PA: A-List, 2005.

Texturing & Modeling a Procedural Approach. San Francisco, Calif: Morgan Kaufmann, 2003.

Press, William H., and William T. Vetterling. Numerical Recipes in C: The Art of Scientific Computing. Cambridge: Cambridge U, 1992.

Tracy, Dan, and Sean Tracy. CryEngine 3 Cookbook: Over 90 Recipes Written by Crytek Developers for Creating Third-generation Real-time Games;. Birmingham Packt Publ.: n.p., 2011.

<http://extra-credits.net/articles/> (Analyses of Game Design)

Wann, John P., Rushton, Simon., Mon-Williams, Mark. Natural Problems for Stereoscopic Depth Perception in Virtual Environments

Felicia, Patrick. Proceedings of the 6th European Conference on Games Based Learning, 4-5 October 2012, , Ireland: Edited by Patrick Felicia. Reading: Academic International Limited, 2012

Gregory, Jason. Game Engine Architecture. Wellesley, MA: K Peters, 2009. Gregory, Jason. Game Engine Architecture. Wellesley, MA: K Peters, 2009.

Aytekin, Murat, Elena Grassi, Manjit Sahota, and Cynthia F. Moss. "The Bat Head-related Transfer Function Reveals Binaural Cues for Sound Localization in Azimuth and Elevation." J. Acoust. Soc. Am. The Journal of the Acoustical Society of America 116.6 (2004): 3594.

http://www.umiacs.umd.edu/~ramani/cmsc828d_audio/HRTF_INTRO.pdf (Introduction to Head-Related Transfer Functions)

<http://graphicdesignjunction.com/2014/01/innovative-ui-design-concepts-to-boost-user-experience/> (GUI Inspiration)

<https://msdn.microsoft.com/en-us/library/windows/desktop/aa362927%28v=vs.85%29.aspx> (Bluetooth Documentation)

https://www.youtube.com/watch?v=iqcNrl_hkMs (Stereoscopic Rig Theory)

<https://strawpoll.me/> (Strawpolls ?)

http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/FUSIELLO4/tutorial.html (Elements of Geometric Computer Vision)

<https://www.wolframalpha.com/> (General Mathematical Reference)

<http://oculusrift-blog.com/john-carmacks-message-of-latency/682/><http://oculusrift-blog.com/john-carmacks-message-of-latency/682/> (Optimal VR Latency)

<http://www.codeproject.com/Articles/125478/Versatile-WebCam-C-library> (Webcam Library)

<https://touchless.codeplex.com/> (Existing Tracking Library)

<http://docs.unity3d.com/Manual/index.html> (Unity Manual)

<http://thrust.github.io/> (A parallel algorithms library)

<http://en.wikipedia.org/> (General Reference)

http://www.theverge.com/a/virtual-reality/oral_history (Oral History of VR)

<http://stackoverflow.com/> (General Reference)

https://ccrma.stanford.edu/courses/220c-spring-2009/final_reports_PDFs/Bejoy-Surround.pdf (Virtual Surround Sound Implementation with Decorrelation Filters and HRTF)

Acknowledgements

I would like to thank my friends for encouraging me through this even though, more often than not, they had no idea what I was doing. Kudos to Raghvi Arya for 'forcing' me to enter the Google Science Fair and Alice Birch for helping to test the very first pre-prototypes as well as providing encouragement. In addition, all those who responded to my strawpolls and helped to test the prototypes are to be acknowledged – they are totally awesome and are entitled to a high-five if and when we meet in real life (choice of forehand or backhand). I would also like to give the Virtual Reality and Indie Student Developers Facebook groups a shout-out for their support and feedback and ongoing passion for the cause – it was invaluable to me. Thanks goes to Steve Salmond for releasing the source to his entry to Ludum Dare 30 "John and the Arbitrary Gem Hunt" into the Public Domain. It was used in the Results section to demonstrate how easy it is to port games to Virtual Reality using my API. Special thanks go

to Microsoft's DreamSpark program which continues to provide free professional licenses and development tools to students worldwide.

References

1. Virtual Reality Exposure Therapy for Combat-Related Posttraumatic Stress Disorder
2. An Oral History of Virtual Reality
REFERENCE LINK
3. Oculus Blog - John Carmack's Delivers Some Home Truths About Latency
REFERENCE LINK
4. Orland, Kyle. "How Fast Does "Virtual Reality" Have to Be to Look like "Actual Reality"?" (2013)
5. Wann, John P., Rushton, Simon., Mon-Williams, Mark. Natural Problems for Stereoscopic Depth Perception in Virtual Environments
6. Fusiello, Andrea. Elements of Geometric Computer Vision.
REFERENCE LINK
7. DirectShow Official Website
REFERENCE LINK
8. Aytekin, Murat, Elena Grassi, Manjit Sahota, and Cynthia F. Moss. "The Bat Head-related Transfer Function Reveals Binaural Cues for Sound Localization in Azimuth and Elevation." J. Acoust. Soc. Am. The Journal of the Acoustical Society of America 116.6 (2004): 3594.
9. Bejoy, Jakes. Virtual Surround Sound Implementation Using Decorrelation Filters and HRTF
REFERENCE LINK
10. Bluetooth Smart - Official Bluetooth Website
REFERENCE LINK
11. Steve Salmond - John and the Arbitrary Gem Hunt
REFERENCE LINK
12. Nokia Lumia 520 - Nokia's low end Windows Phone 8
REFERENCE LINK